Author: Witsenburg, Tijn
Title: Hybrid similarities : a method to insert relational information into existing data mining tools
Date: 2012-12-20

Chapter 3

Hybrid Similarity Measures

There is a need for generative methods to create hybrid data mining techniques. We propose a method that will allow the user to include relational information in any data mining tool that uses a content-based similarity or dissimilarity measure. We also show what the effect is of using this method on a distance or metric. The method is explained and formalized in two ways: one from a graph perspective and one from a matrix perspective.

3.1 Introduction

Data mining tries to find interesting patterns in large data sets. What patterns are interesting depends on the goals of the user. What these patterns look like depends on the data. Early data mining algorithms focus on content-based data. In a content-based dataset, all the data tells something about a single element and not how such an element relates to others. Despite the fact that it is not always written in that form, it could be seen as single table data: every row is an element, and every column is an attribute for those elements.

There are also datasets that do not fit in this form. Here, the information does not tell something about an element, but it tells something about how an element relates to its context. This is considered contextual data. Data mining algorithms designed for single table data are useless for contextual data. Therefore, this type of data requires the creation of a variety of data mining algorithms. Many come from the field of ILP or graph analysis.

There is also data that consists of both content-based and contextual data. A data mining algorithm that uses only one of the two types of information, cannot explore the complete data space. Therefore, there is a need for methods that work on such datasets and can explore the data space fully. Recently, Neville et al. [72] and Zhou et al. [102] created methods that insert content-based information in already existing contextual data mining techniques.

We propose a method that inserts contextual data in already existing data mining techniques that work on content-based data. It does so by adapting the similarity measure that the original data mining algorithm uses. In this chapter, the exact working of this hybrid similarity is explained in detail. The chapter is organised as follows. Section 3.2 explains the basic principle without going into technical details. Section 3.3 gives the formal definitions that describe the data for which this method is created. Section 3.4 defines the proposed hybrid similarities from a graph-based point of view, and describes how they relate to the original, content-based similarity. Section 3.5 describes the same hybrid similarities, but now from a matrix-based point of view. Section 3.6 explains how these hybrid similarities are plugged into the existing data mining algorithm. Finally, Section 3.7 gives some advice on the situation in which it is useful to use any of the hybrid similarities.

3.2 The Basic Principle

A data set comprises of a set of “objects of interest,” which are the objects about which the data set is (e.g. scientific papers, people, purchases, et cetera). These objects will be referred to as “elements.” Besides these elements, a data set normally contains information that tells us more about these elements. In this case, two different types of information are distinguished: information that tells the user more about a certain element, and information that tells the user more about how a certain element relates to other elements. The first type will be known as the *content-based* information and the second type will be known as the *contextual* information.

To clarify this a bit more, consider a database of scientific papers with their texts and citations. The objects of interest, or elements, are the scientific papers. A data mining tool therefore should be saying something about a paper, or a group of papers. Examples can be: “this paper is about that subject,” or “these papers are very similar and thus probably about the same subject.” The content-based information in this case is the text of the paper and the relational information is the information about the citations.

As stated in Chapter 2, it is not so straightforward to combine these two types of information, and therefore data mining tools typically use only one of the two. The method used in this thesis combines the two types of information indirectly. To do so, it assumes there is a similarity or dissimilarity measure that can tell how much any two elements look like each other. With a similarity measure, a high value between two elements means they are very similar and thus look a lot like each other. A dissimilarity measure works in the exact opposite way: the lower its value, the more two elements are alike. Despite this difference in meaning between high and low values for similarity and dissimilarity measures, they act in a similar way. So from now on, we consider similarity measures only, but the same story could be told for dissimilarity measures.

To summarize, there is a data set with elements that have content-based information that can be quantified by a similarity measure expressing how similar two elements are, and relational information between these elements. Now, a hybrid similarity measure can be constructed by regarding the similarities from one element to the "neighborhood" of another. This neighborhood is defined by the contextual information in the dataset. The similarity between one element and the neighborhood of the other is defined by the content-based information in the dataset. Therefore, it is clear that the hybrid similarity measure uses both types of information.

This is visualised in Figure 3.1 where the hybrid similarity between two elements v and w can be calculated. Two types of information are drawn in this figure. The contextual information is drawn by a solid line indicating that there is a relation between these two elements. Elements that are related are known also as "neighbors." The neighborhood of an element is the set of all its neighbors. Thus, in this case, the neighborhood of v is $\{a, b\}$ and the neighborhood of w is $\{c, d, e, f\}$. The content-based information is illustrated in this figure by either the similarity between v and w (dashed line) or the similarity between v and the neighbors of w (dotted lines).

Indirectly, the two types of information are used. The relational information is used to determine which are the neighbors of an element. The content-based information then is used to calculate the similarity between an element and the neighbors of the other element.

3.3 The Data Format: Annotated Graphs

Consider the dataset that needs to be clustered to be an annotated graph. This data set D can be defined as $D = (V, E, \alpha, \lambda)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of n vertices or elements, $E \subseteq V \times V$ is the set of edges, $\alpha : V \rightarrow \mathcal{A}$ a function that assigns to any vertex v of V an "annotation", and $\lambda : V \rightarrow \mathcal{L}$ a function that assigns to any vertex v of V a "label". The annotation $\alpha(v)$ is considered to be the *content* of vertex v and the label $\lambda(v)$ is considered to be the class of vertex v.

The graph is undirected and an edge cannot loop back to the same vertex, so with two vertices $v, w \in V$ this means that $(v, w) \in E \Leftrightarrow (w, v) \in E$ and $(v, v) \notin E$. Furthermore, for every vertex $v \in V$, the neighborhood of that vertex is defined as $\mathcal{N}(v)$, where $\mathcal{N}(v)$ is the set of vertices that are connected to v with an edge. So with two vertices $v, w \in V$, $w \in \mathcal{N}(v) \Leftrightarrow (v, w) \in E$. The amount of neighbors of v can be seen as the size of $\mathcal{N}(v)$ which is denoted as $|\mathcal{N}(v)|$, and since there are no loops ($(v, v) \notin E$), the amount of neighbors of v is the same as the degree of v, denoted as $\textsc{deg}(v)$. Furthermore, every element has at least one neighbor, so $\mathcal{N}(v) \neq \emptyset$, and thus $|\mathcal{N}(v)| > 0$

The space of possible annotations is left open; it can be a set of symbols from, or strings over, a finite alphabet; the set of reals; an n-dimensional Eu-

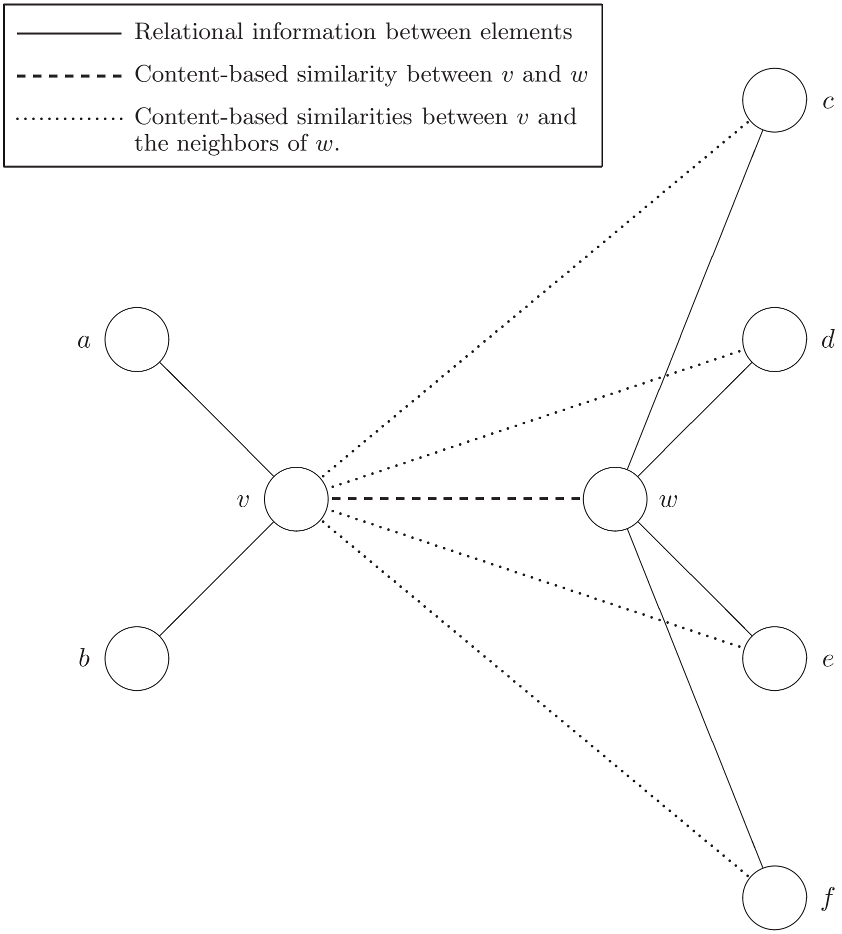

——— Relational information between elements
- - - Content-based similarity between v and w
⋯ Content-based similarities between v and the neighbors of w.
a
b
v
w
c
d
e
f

Figure 3.1: Visualisation of the basic principle behind the hybrid similarity. In this picture, the solid lines are the relations as described by the contextual information in the database. Thus, a and b are the neighbors of v, and c, d, e, and f are the neighbors of w. When calculating the similarity between v and w, a data mining tool designed for data in the single table form would only consider the content-based similarity between v and w (dashed line). A graph mining tool, on the other hand, only would consider the relational information (solid lines). The hybrid method regards the content-based similarities from one node to the neighbors of the other (dotted lines). In this picture, for clarity reasons, the content-based similarities between w and the neighbors of v (a and b) are left out.

clidean space; a powerset of one of the sets just mentioned; etc. The only constraint on $\mathcal{A}$ is that it must be possible to define a similarity measure $\mathcal{S} : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ as a function that assigns a value to any pair of annotations expressing the similarity between these annotations. Since this similarity is entirely based on the content of the vertices, it will be called the *content-based similarity*, or $\mathcal{S}_{content}$. Normally the value of this similarity is in the range $[0, 1]$ where 0 stands for no similarity at all and 1 means that the elements are considered to be identical.

The space of possible labels is defined more strictly. It is a finite, discrete set, so $\mathcal{L} = \{L_1, L_2, \ldots, L_l\}$. The label of a vertex can be seen as its class. With supervised learning, this label is given and it is the task of the DM tool to learn a relation between V, E and α on one side and λ on the other. With semi-supervised learning, part of λ is given and the rest is unknown or hidden. The DM tool then needs to predict this rest, given V, E, α and part of λ. With unsupervised learning, λ is completely unknown or hidden and the DM tool needs to predict it, given V, E and α. When information about λ is hidden, it can be used to check the quality of the found solution after the DM tool has finished.

3.4 Similarity Measures

This section defines the main similarity measures as used in this thesis.

3.4.1 Content-based Similarity

As said, we assume a similarity measure $\mathcal{S} : \mathcal{A} \times \mathcal{A} \rightarrow \mathbb{R}$ is defined over the space of annotations $\mathcal{A}$. From this, we can immediately derive a first similarity measure on the nodes of the graph, which we call the content-based similarity:

$$\mathcal{S}_{content} : V \times V \rightarrow \mathbb{R} : \mathcal{S}_{content}(v, w) = \mathcal{S}(\alpha(v), \alpha(w)) \tag{3.1}$$

In other words, the content-based similarity, $\mathcal{S}_{content}$, is a similarity that is based purely on the content, or annotations, of the vertices. In general, this is the similarity measure that originally is used in data mining tools as described in Section 2.2. In Figure 3.1, it is illustrated by the dashed line between v and w. There, it is also clear, that no relational information is used to compute this similarity.

3.4.2 Contextual Similarity

To define the contextual similarity, it is necessary to first define the *neighborhood similarity* $\mathcal{S}_{neighbor} : V \times V \rightarrow \mathbb{R}$ between two vertices v and w from V, as the average content-based similarity between v and all neighbors of w:

$$\mathcal{S}_{neighbor}(v, w) = \frac{\sum_{u \in \mathcal{N}(w)} \mathcal{S}_{content}(v, u)}{|\mathcal{N}(w)|} \tag{3.2}$$

In Figure 3.1 this similarity is computed as the average of the dotted similarities. It is not symmetric, but it can easily be symmetrized. This leads to the *contextual similarity* $\mathcal{S}_{context} : V \times V \to \mathbb{R}$:

$$\mathcal{S}_{context}(v, w) = \frac{\mathcal{S}_{neighbor}(v, w) + \mathcal{S}_{neighbor}(w, v)}{2} \tag{3.3}$$

The motivation behind this similarity measure is that, if similar nodes tend to be linked together, then the neighbors of w in general are similar to w. Hence, if similarity is transitive, a high similarity between v and many neighbors of w increases the reasons to believe that v is similar to w, even if there is little evidence of such similarity when comparing v and w directly (for instance, due to noise or missing information in the annotation of w).

3.4.3 Combined Similarity

The contextual similarity measure from (3.3) is complementary to the content-based similarity from (3.1); it does not use the content-based similarity between v and w at all. This can be seen easily in Figure 3.1. The content-based similarity is represented by the dashed line and the contextual similarity by the dotted lines, of which it is the average. The dotted lines are derived from the relational information drawn by the solid lines, but the dashed line is never used. In practical settings, it may be good not to ignore the content-based similarity entirely, so Witsenburg and Blockeel [99] also introduced the weighted average of the content-based similarity and the contextual similarity as the *combined similarity* $\mathcal{S}_{combined} : V \times V \to \mathbb{R}$:

$$\mathcal{S}_{combined}(v, w) = c \cdot \mathcal{S}_{content}(v, w) + (1 - c) \cdot \mathcal{S}_{context}(v, w) \tag{3.4}$$

with $0 \leq c \leq 1$. The constant c determines the weight of the content-based similarity in the combined similarity. As Witsenburg and Blockeel [99] found no strong effect of using different values for c, from now on we consider only the combined similarity with $c = \frac{1}{2}$.

Both the contextual and the combined similarity measures are hybrid similarity measures, since they use both content-based and graph information. Whereas the contextual similarity measure between two nodes v and w does take the contents of v and w into account, it just does not use the direct similarity between these contents.

Note that any standard clustering method that can cluster nodes based on (only) their content similarity can also cluster nodes based on the contextual or combined similarity, and in the latter case it implicitly takes the graph structure into account; the method itself does not need to be altered to be able to process graph data.

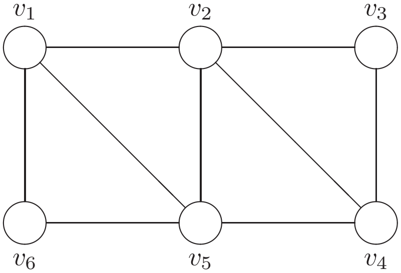
3.5 Matrix-based Computations

In this section, the contextual similarity will be explained in the light of matrix multiplication, to create a broader perspective on the hybrid similarities.

Considering the dataset $D = (V, E, \alpha, \lambda)$ as described in Section 3.3, it can be seen that there are two types of information in it: content-based information and relational information. The content-based information is defined by the function α that assigns an annotation to each element. The relational information is defined by the set of edges E, which defines whether there is a relation between two elements or not.

From these two types of information, two different $n \times n$ matrices can be constructed (where n is the amount of vertices in V, or the amount of elements in the dataset D). The first matrix would be the similarity matrix S, where every element s_{ij} is the content-based similarity between v_i and v_j: $s_{ij} = \mathcal{S}_{content}(v_i, v_j)$. The second matrix is the adjacency matrix A where each element a_{ij} is defined by:

$$a_{ij} = \begin{cases} 0 & \text{if } (v_i, v_j) \notin E \\ 1 & \text{if } (v_i, v_j) \in E \end{cases} \tag{3.5}$$



$$\begin{array}{ccccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \\ v_1 & 0 & 1 & 0 & 0 & 1 & 1 \\ v_2 & 1 & 0 & 1 & 1 & 1 & 0 \\ v_3 & 0 & 1 & 0 & 1 & 0 & 0 \\ v_4 & 0 & 1 & 1 & 0 & 1 & 0 \\ v_5 & 1 & 1 & 0 & 1 & 0 & 1 \\ v_6 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$$

Figure 3.2: Example of a graph and its adjacency matrix

Figure 3.2 shows an example of an undirected and unweighted graph, and its adjacency matrix. The adjacency matrix has one feature that is aspecially interesting for calculating the hybrid similarity measure. When the adjacency matrix is multiplied with itself, $A^2 = A \times A$, then every element a'_{ij} in A^2 with $i \neq j$ gives the amount of paths from v_i to v_j with length 2. An element a'_{ii} in A^2 gives the amount of neighbors of v_i.

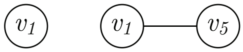
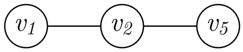
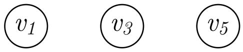
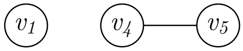
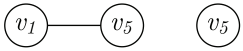
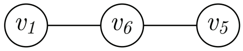
k	a_{ik}	a_{kj}	$a_{ik} \cdot a_{ik}$	*graph representation*
1	0	1	0	
2	1	1	1	
3	0	0	0	
4	0	1	0	
5	1	0	0	
6	1	1	1	

Table 3.1: Visualisation of how an element a'_{ij} from A^2 denotes the amount of paths of length 2 in the graph represented by A. In this example $i = 1$ and $j = 5$. The first column shows k which in the sum goes from 1 to $n = 6$. The second and third column show the corresponding a_{ik} and a_{kj} respectively. The fourth column shows the product of a_{ik} and a_{kj}. The fifth column shows the edges from the graph that match the a_{ik} and a_{kj} from the second and third column. It clearly shows that the fourth column gives all the terms of the sum from (3.6) and that each of those terms is one whenever there is a path through the corresponding v_k.

To illustrate this, consider that each element a'_{ij} in A^2 is calculated with

$$a'_{ij} = \sum_{k=1}^{n} a_{ik} \cdot a_{kj} \tag{3.6}$$

In the case of the graph from Figure 3.2, we have $n = 6$. In the same figure it can be seen that an element in A is either 0 (no edge) or 1 (edge), so each term in the sum from (3.6) is 1 when both a_{ik} and a_{kj} are 1, and 0 otherwise. In the graph, this means that such term is 1 for every k where the edges (v_i, v_k) and (v_k, v_j) both exist, and thus, the sum counts the amount of vertices that are connected to both v_i and v_j. There exists a path of length 2 from v_i to v_j if, and only if, there is a vertex that is connected to both v_i and v_j. So counting the number of vertices that are connected to both v_i and v_j is the same as counting the number of paths of length 2 from v_i to v_j, and thus every element a'_{ij} in A^2 gives the amount of paths of length 2 between v_i and v_j. This principle is also illustrated in Table 3.1.

Regarding how (3.6) was converted to the drawing in the fifth column of Table 3.1, it is easy to see how on the main diagonal of A^2 the degrees of the vertices arise. First, take a closer look at this drawing. Figure 3.3 shows it again,

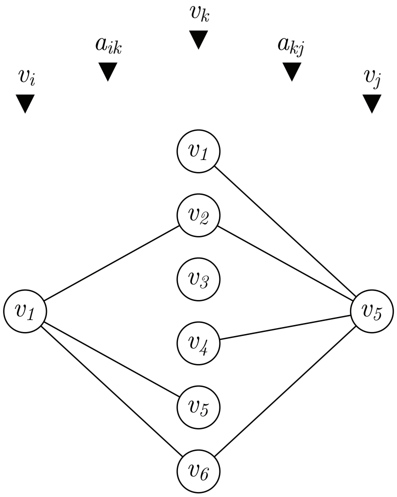


Figure 3.3: This drawing is created from the fifth column of Table 3.1 by merging all begin and end vertices (v_1 and v_5 respectively). Also the corresponding labels for the calculation of a_{15} from A^2 are added. On the far left and far right, v_i and v_j respectively, which in this case are v_1 and v_5. In the middle, the column of vertices comes from k going from 1 to n. Vertices v_i and v_j are connected to the vertices in the center column when they are connected to them in the graph.

but this time with labels indicating what part of the matrix multiplication is represented by which element in the drawing. Thus, it is easy to see what the effect is of substituting an element in (3.6).

For instance, the effect of substituting v_j with v_i and adjusting the connecting edges accordingly, is shown in Figure 3.4. This is the same effect as substituting v_j with v_i in (3.6). So calculating an element a'_{ii} in A^2 results in Figure 3.4 (right). Here it is easy to see that on the main diagonal in A^2 the degrees of the vertices arise. The result is that the amount of paths of length 2 in Figure 3.4 (right) is therefore the same as the amount of vertices connected to v_i.

It is also possible to substitute one of the matrices from the multiplication. In that case the edges connecting either one or the other vertex come from this new matrix. To ensure that the result is useful, this new matrix needs to have the same “shape” as A (i.e. every row or column must refer to the vertex in the graph as that row or column would do in A). The similarity matrix S is therefore a suitable candidate. Both the first and the last ‘A’ in ‘$A \times A$’ can be substituted, resulting in either ‘$A \times S$’, or ‘$S \times A$’. Figure 3.5 shows the effect of these substitutions. A comparison between these pictures and Figure 3.1 shows the resemblance between the two.

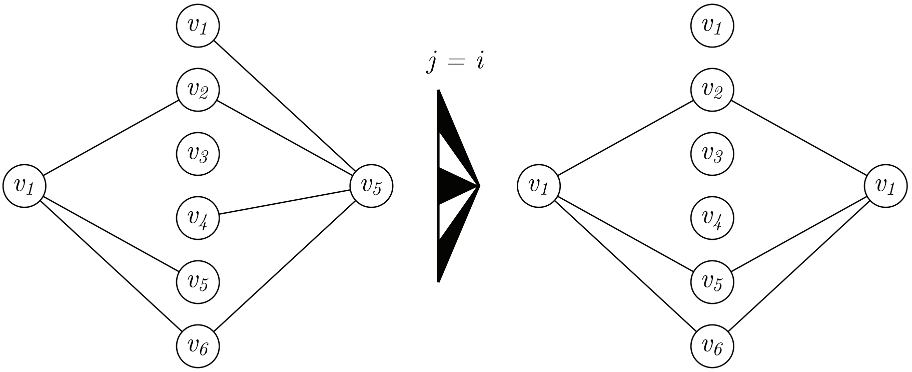


Figure 3.4: Taking Figure 3.3 (on the left), substituting v_j with v_i, and adjusting the edges connecting the second v_i accordingly, results in the picture on the right. Here, it is easy to see how on the main diagonal of A^2 the degrees of the vertices arise. This is simply counting the paths of length 2 that have occurred, which is the same as the amount of vertices connected to v_i.

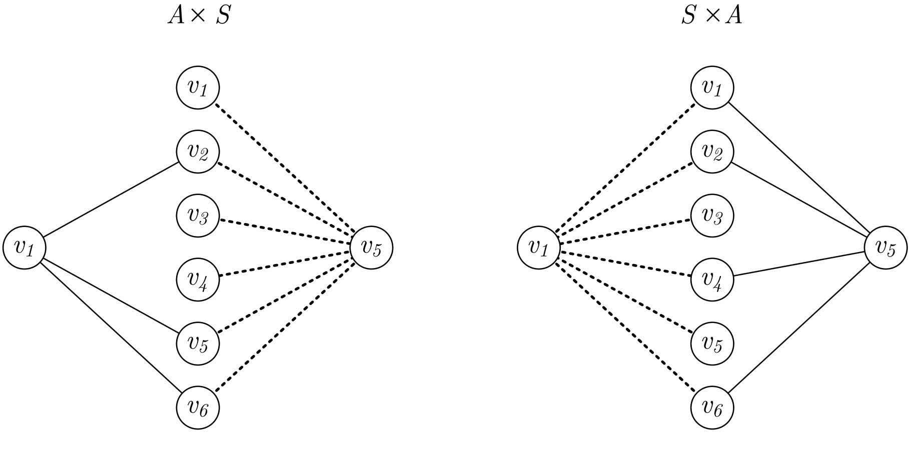


Figure 3.5: The result of substituting a_{kj} with s_{kj} (left) and a_{ik} with s_{ik} (right). Both substitutions come from Figure 3.3.

As the edges have been replaced, again, the amount of paths of length 2 needs to be counted. But now, any such path is a combination of an edge from the graph and a similarity from S. To be more precise, any element x_{ij} from $A \times S$ is the sum of the similarities between v_j and all neighbors of v_i, and any element x_{ij} from $S \times A$ is the sum of the similarities between v_i and all neighbors of v_j. This is very close to the contextual similarity from (3.3), but that is the average of all similarities instead of the sum.

So, to formulate the similarities from Section 3.4 with matrices, only a little adjustement from combining $A \times S$ with $S \times A$ is needed. In order to come to the average similarity instead of the sum, one needs to divide all values in the resulting matrix by the degree of the vertex whose neighborhood is used. This is the vertex that corresponds to the row or column in the adjacency matrix that is used for calculating the element in the resulting matrix. Thus, every element in the i-th row of the result of $A \times S$ needs to be divided by the degree of v_i, and every element in the j-th column of the result of $S \times A$ needs to be divided by the degree of v_j.

Dividing by a number is the same as multiplying with its inverse. Therefore, consider a neighbor matrix N, which is a diagonal matrix where each element n_{ii} is the *neighbor factor* and is one divided by the number of neighbors of v_i. Now, the contextual similarity from the previous section can also be computed by:

$$\frac{N \times (A \times S) + (S \times A) \times N}{2} \tag{3.7}$$

The hybrid similarity measure is therefore a method that multiplies the adjacency matrix with the similarity matrix and vice versa, and it uses these as similarity measures for the data mining tool. The only thing that needs to be done is a small adjustment in the form of taking average values (to make sure all values stay in the same range).

3.6 Plug and Play

The method proposed in this thesis is not a new data mining tool. Instead of that, it can be seen as a method to add relational information to an already existing data mining tool that only uses content-based information. To illustrate this idea, first consider Figure 3.6. This figure shows a data mining tool that can be considered as a black box; it is not important to know how it works precisely. The only features that are important are that it only uses the content-based information from the data set as input and some similarity measure that is based only upon that. With that, the data mining tool comes to its results.

Many data mining tools work regardless of which exact distance or similarity measure is used. For instance, when the content-based information in the data set consists of vectors, a data mining tool that uses the cosine between two vectors as the similarity, also could use the Jaccard index or Manhattan distance

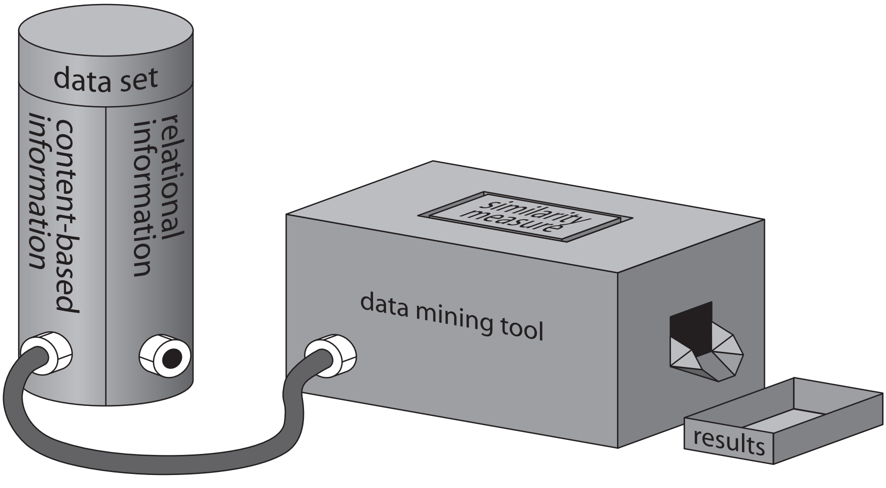


Figure 3.6: A data mining tool can be considered as a black box. The content-based information from the data set is used as an input leading to results as an output. The data mining tool uses some similarity measure based only on the content-based information from the data set. The relational information from the data set is not used.

between them. This principle is illustrated in Figure 3.7. The similarity measure can be seen as a plugin for the data mining tool. For the data mining tool to work, one needs to plugin a similarity measure. This can be anything as long as it fits the data. This similarity measure can therefore be regarded as independent of the data mining tool.

Instead of plugging the preferred content-based similarity directly into the data mining tool, one can also choose to plug this content-based similarity into the hybrid similarity measure from this thesis. At this point, one can use this result as similarity measure for the data mining tool. This principle is illustrated in Figure 3.8. Any similarity measure that can be plugged into the data mining tool can be plugged also into the hybrid similarity measure. This combines the content-based similarity measure with the relational information from the data set to come to a new similarity measure. Then, this new similarity measure can be used in the data mining tool in the same way that the content-based similarity would have been used. The data mining tool does not need to be adjusted for this.

One benefit of the hybrid similarity measure is the fact that the user does not need any extra domain specific knowledge to use it. This cannot always be said of content-based measures. For instance, Monge and Elkan use an alphanumeric edit distance to identify duplicate alphanumeric records [68]. Also,

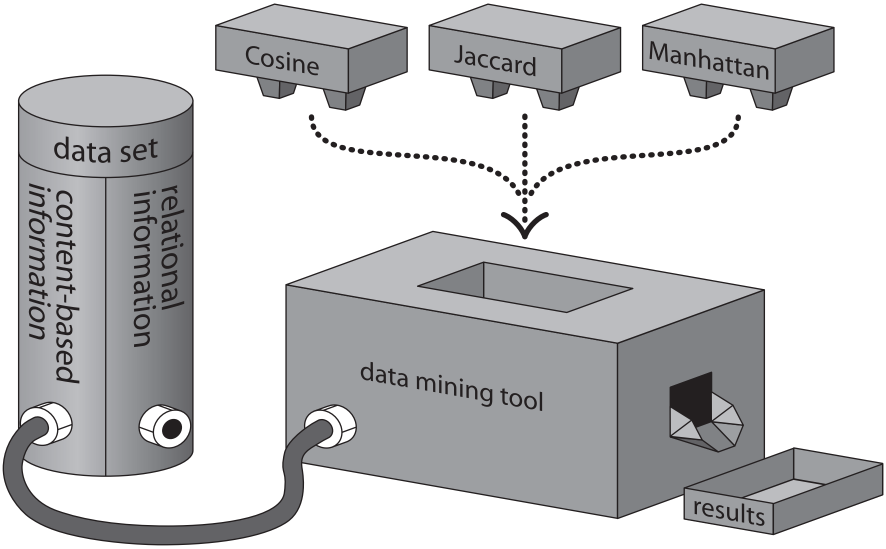


Figure 3.7: A data mining tool can use different similarity measures. For instance, a data mining tool that uses the cosine similarity, would work according to the same principles when using the Manhattan distance or the Jaccard index.

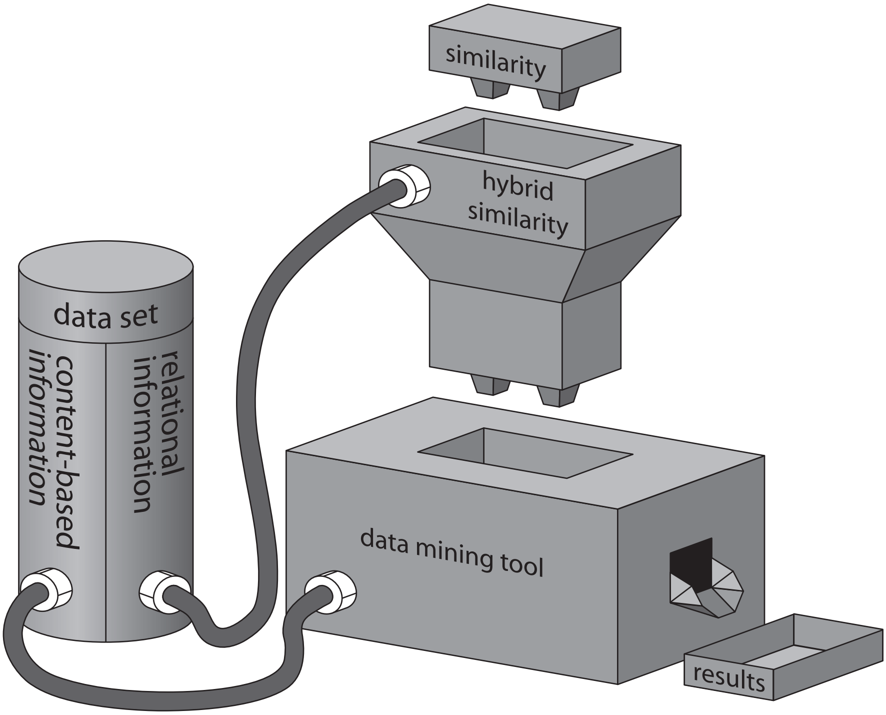


Figure 3.8: The hybrid similarity measure proposed in this thesis can be seen as a plugin for a data mining tool. Instead of plugging a content-based similarity measure directly into the data mining tool, this measure can be plugged into the hybrid similarity measure. The hybrid similarity measure combines the content-based similarity measure with the relational information from the data set. In this way, it comes to a new similarity measure that can be used by the data mining tool in a similar way as it would have used the content-based similarity.

data mining on protein structures is often done using a distance function that rotates and translates structures to superimpose them. If the user understands the content-based similarity, then this is enough to use the hybrid similarity.

3.7 Using the Right Content-based Similarity

The previous section showed that the hybrid similarity measure can replace the content-based similarity. This can always be done. However, to increase the chance that the extra effort and computing time will indeed lead to better results, one needs to make sure that the original similarity measure is not based upon the relational information. As an illustration, consider a data mining tool that uses a distance measure that takes the length of the shortest path between two elements in an unweighted and undirected graph as the distance between them. In this case, using the hybrid similarity does not add much to the original similarity. This can easily be seen.

Assume the distance between two elements v and w, as defined by the length of the path between them, is p. To calculate the contextual distance between v and w, the distances between v and all neighbors of w and between w and all neighbors of v need to be aggregated. These distances are limited in comparison to the distance p. Since a neighbor of v or w is, by definition, respectively connected to v or w, the maximum length for any of these distances is $p+1$. In a similar way, it can be concluded that the minimum length of these distances is $p-1$. Thus, all distances between v and the neighbors of w, or between w and the neighbors of v are in the range $\{p-1, p, p+1\}$.

This means that the contextual distance between v and w is highly dependent on the original distance between v and w. Therefore, one can safely conclude that it is to be expected that the results with the contextual distance will not differ a lot from using only the original distance. Hence, the added value of the hybrid similarity is extremely small. So, if we want the hybrid similarity to actually add something to the original distance, we should take care that the original distance is not based on relational information.