Chapter 4

Motivation

We give motivation for the potential improvement of the hybrid similarities in comparison to a similarity measure that is only based on the content. For many different reasons, it is difficult for a similarity measure to get the semantic similarity between elements precisely. When calculating the similarity between elements, the hybrid similarities explore a greater part of the data space. This creates a more confident value for a similarity between two elements.

4.1 Introduction

In general, data mining algorithms are designed for a specific type of data: content-based data or contextual data. Only recently, methods are proposed that can work on hybrid data (i.e. data that consists of both types). In this thesis we propose a new method for this task. This method inserts contextual information in a data mining algorithm that is designed for content-based information.

As proposed in this thesis, the contextual similarity between two elements is the average, content-based similarity of these elements and the neighbors of the other. This thesis also proposes that the combined similarity is the average of the contextual similarity and the original, content-based similarity between the two elements. From a matrix perspective, these new similarities are created by multiplying the adjacency matrix of the graph with the similarity matrix of the elements, and vice versa, and then adjusting the values for the size of the neighborhood.

This chapter gives a theoretical motivation for why we expect the hybrid similarities could outperform the content-based similarity. We also try to formulate characteristics for datasets where it could be beneficial to use the hybrid similarities. The rest of this chapter is organized as follows: Section 4.2 describes three characteristics of datasets. Section 4.3 describes the principles about distance and similarity measures. Section 4.4 describes the difference between semantic and measured similarities. Section 4.5 combines all these characteristics and principles in an explanation on why the hybrid similarity will, most likely, outperform the original similarity. Finally, Section 4.6 gives a summary of this chapter.

4.2 Important Characteristics for Networks

To understand the working of the hybrid similarity measure, it is necessary to first take a closer look at three possible characteristics of network based databases that play a big role in this. These are homophily, transitivity of similarity, and content variability.

Homophily

In a network or linked dataset, *homophily* is the characteristic that can be described as the tendency of elements that are similar to be connected. One example is a dataset that consists of scientific papers. In this dataset is, amongst others, information about which paper cites which other paper. The latter can be seen as relational information, for it links two papers when one cites the other. In general, most citations in a scientific paper are to papers that are (roughly) about the same subject. Thus, a set of papers about the same subject will be highly connected.

Another example is a social network. Here, the relations are defined by the ability of users to indicate which other users are their friends. Often, friends of friends have a good chance of becoming your friends as well, thus creating highly connected groups of friends. Also, friends tend to like the same things (e.g. movies, books, activities). This goes in two directions: friends of someone share their interests with him and then he gets excited by it himself, or someone goes to an event for people with a certain interest and meets people there who become his friends.

Overall, there are many situations where similar items tend to be highly connected. Also, the neighborhood of an element tends to consist of elements that are fairly similar. Of course, it can always happen that highly dissimilar elements are connected or that highly similar elements are not connected.

Transitivity of Similarity

Given a domain and a similarity measure, *transitivity of similarity* can be described as the fact that when two elements are similar, a third element that is similar to one of the two, is also fairly similar to the other. If the chosen similarity measure does not follow this rule, very undesirable things can happen. For instance, consider three elements, u, v and w, where v is very similar to both u and w, but u and w are very dissimilar to each other. A clustering algorithm could place u and w in the same cluster as v since they are very similar. However, this means that u and w are in the same cluster, which is highly unwanted since they are very dissimilar.

When a distance measure is used instead of a similarity measure, the principle is shown in the triangular inequality. It states that for any three elements u, v and w, it should hold that $d(u, w) \leq d(u, v) + d(v, w)$, where $d(v, w)$ denotes the distance between v and w.

Let the reader be advised that this is not exactly the same as 'transitivity of equality', which is precisely formulated as '$a = b \wedge b = c \Rightarrow a = c$.' Transitivity of similarity, on the other hand, can be formalized as '$a \sim b \wedge b \sim c \Rightarrow a \approx c$'.

Content Variability

The contents of nodes from the same class may differ, and in most real life data where elements are characterized by their content, they do so. Consider, for instance, a database with texts about various subjects. Amongst it might be some texts about football players. All of these texts about football players will be different, but some words that are related to football (e.g. cup, goal, and league) will, on average, appear more often in those texts than in other texts. To illustrate this, Table 4.1 shows how often certain words that are highly related to football appear in five texts about football players.

The amount of times a particular word about football appears in such a text is obviously not the same for all texts. However, regarding all texts about football players, there is some sort of average amount of times that certain words appear in them.

It is obvious that even for words that are related to football, the amount of times they appear in a text about a football player will differ. However, in general, those words appear more often. From this, as one takes all texts about football players, it is possible to create a mean vector that gives for each word the most typical amount of times it appears in a text about a football player. The actual amount of times it appears in a specific text about a football player can of course highly differ from this average. But, in general, the keyword vector of a text about a football player will more or less point in the same direction as this mean keyword vector. *Content variability* refers to how much the content of an individual element may differ from the mean content of all elements in the same class.

4.3 Distances, Similarities, and Dissimilarities

The hybrid similarity works for data mining tools that use some kind of measurement to compare two elements with each other. This measurement should state how much these element look alike. This can of course be done in many

Keyword	Mean	Best	Daei	Di Stéfano	Kuijt	Sócrates
ball	1.8	8	0	0	1	0
career	9.4	11	12	8	9	7
club	19.0	16	22	21	28	8
coach	4.4	1	17	2	1	1
competition	3.6	3	7	2	6	0
cup	39.4	16	98	37	31	15
draw	1.2	1	1	0	3	1
stadium	7.8	2	0	24	13	0
fans	2.0	1	5	1	2	1
fifa	10.4	5	32	10	2	3
final	5.8	3	4	6	15	1
football	11.4	16	22	7	3	9
footballer	5.4	9	3	6	2	7
game	11.2	8	13	3	28	4
goal	34.4	25	45	9	84	9
international	12.4	3	29	20	9	1
league	21.0	14	35	10	32	14
manager	5.8	7	13	6	3	0
match	16.2	18	32	11	18	2
opponent	1.2	2	1	1	2	0
pass	1.2	1	1	0	2	2
penalty	3.0	1	0	0	14	0
pitch	2.2	4	2	1	1	3
play	17.0	22	16	17	19	11
player	10.6	11	16	14	7	5
professional	2.0	3	1	0	3	3
qualification	17.4	9	47	7	23	1
result	2.4	4	4	1	3	0
scorer	4.2	2	9	4	5	1
score	20.4	18	20	4	58	2
season	14.2	15	12	5	38	1
squad	2.4	2	3	0	6	1
striker	2.6	2	1	0	9	1
supporter	1.0	0	1	0	1	3
team	14.2	9	34	8	15	5
tournament	1.6	0	4	1	3	0
uefa	3.2	5	5	3	3	0
victory	2.2	1	0	3	7	0
win	14.2	11	9	14	34	3

Table 4.1: Example of the amount of keywords in different texts about football players. This table is created using the wikipedia pages of five football players: George Best, Ali Daei, Alfredo Di Stéfano, Dirk Kuijt and Sócrates.

different ways, but three categories can be distinguished: distances, similarities, and dissimilarities.

A *distance measure* calculates a numeric value that indicates how close or how far two elements are from each other. The smaller the distance between two elements, the closer they are, and thus the more they are considered to look alike. Distance measures in general have no maximum. This means there is no strict upper boundary for a distance measure. In practice, however, the distance is most of the time limited by the domain of the data set.

More formally, a distance measure is defined as a metric. A *metric* is defined as a function $d : V \times V \to \mathbb{R}$ that for all $u, v, w \in V$ meets four properties:

1. *Non-negativity* property: all distances are positive, so $d(v, w) \geq 0$.

2. *Identity* property: when two elements have a distance of 0 then, and only then, are they considered the same. Therefore, $d(v, w) = 0 \Leftrightarrow v = w$.

3. *Symmetry* property: the distance from one element to another is the same as the distance in the opposite direction, so $d(v, w) = d(w, v)$.

4. *Triangle inequality* property: when going from v to w directly, the distance travelled is always shorter than, or equal to the distance travelled when going from v to w through u, so $d(v, w) \leq d(v, u) + d(u, w)$.

The triangle inequality is an important property for a distance measure. This principle has been used by Elkan to seriously speed-up k-means [21], and by Kryszkiewicz and Lasek to improve the efficiency of DBSCAN [52]. Both use the fact that when $d(v, u)$ and $d(u, w)$ are known, an upper boundary for $d(v, w)$ is defined by $d(v, u) + d(u, w)$ without need to calculate $d(v, w)$. Also, Baraty et al. [3] showed that improper use of data mining tools in an environment where there is no triangular inequality could result in highly unwanted outcomes.

Another way to approach the comparison of two elements is not by looking at the distance between them, but rather by comparing how similar they are. This can be done with a *similarity measure*, which is a function $\mathcal{S} : V \times V \to \mathbb{R}$ that for all pairs of elements in V states how similar they are. A high value for a similarity means that they are very similar, which would correspond to a low value for a distance and vice versa. Nevertheless, it is not possible to state that a similarity is the exact opposite of a distance. The four conditions previously described for a metric do not necessarily hold for the inverse of a similarity measure.

The only condition that could be fulfilled by a similarity measure is that its value is always greater than or equal to 0. Any similarity measure used in this thesis is also symmetric, but this is not a constraint in general. The benefit of these constraints is that clever use can speed up many data mining algorithms. The benefit of not fulfilling these constraints is that this leaves a much wider range of possible similarity measures.

	decreasing value means elements are more alike	*increasing value means elements are more alike*
does not meet the properties of a metric	dissimilarity	similarity
meets the properties of a metric	distance	

Table 4.2: Overview of the important differences between distances, similarities and dissimilarities.

Thus, a similarity measure differs from a distance measure in two important ways. First, it does not meet the properties of a metric. Second, an increasing value for a similarity measure means that the elements are more similar, while for a distance measure a decreasing value means that the elements are closer. Sometimes an algorithm needs that an increasing value of the measurement means that the elements are more distant, but the domain of the dataset does not allow to define a metric that meets all four properties. When it is not necessary that the algorithm has these properties, then the user can choose to use a dissimilarity measure (which is basically the opposite of a similarity measure). A *dissimilarity measure* is a function $\mathcal{D} : V \times V \to \mathbb{R}$ where a low value between one pair of elements means that they are closer than a pair of elements with a high value for $\mathcal{D}$. It does not necessarily meet the properties that would make it a metric.

To summarize, a distance measure differs from both a similarity measure and a dissimilarity measure in that it meets the four properties of a metric. A similarity measure differs from both a distance measure and a dissimilarity measure in that an increasing value means that the two elements are more alike. The differences between the three concepts are schematically described in Table 4.2.

It is easy to see that it is not possible to create a measurement where an increasing value means that elements are more alike, and that meets the properties for a metric. According to these properties, all values must be greater than or equal to 0, and if, and only if, an element is compared to itself, this value must be 0. This means that when two different elements are compared, the value for this measurement should be greater than 0. However, this would imply that the value between two different elements is higher than the value when an element is compared to itself. This, of course, is in conflict with the idea that an increasing value will mean that two elements are more alike.

To conclude, there are three types of measurements to compare elements in a data set: distances, similarities and dissimilarities. The hybrid similarity can be used on all three kinds. Since it is calculated as the average of the original measurements, the value of the hybrid similarity will remain in the same range as the original measurement used by the data mining tool.

The downside of the hybrid similarity measure is that when it is used as a plugin for a data mining tool that uses a distance, the triangle inequality that held for the original distance does not necessarily hold for the hybrid similarity measure. This can easily be seen. Consider a data set $D = (V, E)$, with $V = \{u, v, w\}$ and $E = \{(u, v), (v, u), (u, w), (w, u)\}$. There also is a distance measure that gives 0 when the two elements are the same and 1 otherwise. Calculating the *contextual distances*, d_c, using (3.3) from Section 3.4.2, gives: $d_c(u, v) = 1/4$, $d_c(u, w) = 1/4$, and $d_c(v, w) = 1$. In the triangle inequality, $d_c(v, u) + d_c(u, w) \geq d_c(v, w)$, this will lead to $1/4 + 1/4 \geq 1$, which of course is not true.

This means that data mining tools that use the triangle inequality cannot use the hybrid similarity as such. This does not necessarily need to be a problem. Data mining tools that use a distance measure can be divided into three categories with regard to the triangle inequality:

1. The triangle inequality is not used in the data mining tool.

2. The triangle inequality is only used to speed up the data mining tool.

3. The triangle inequality plays an important role in the data mining tool.

It is obvious that for the first category, there is no problem in using the hybrid similarity measure. For the second category, the speed-up cannot be used, but the original algorithm should have no problem in using the hybrid similarity measure. Only for the third category is it impossible to use the hybrid similarity measure. Research should try to find out to what extent this actually is a problem.

4.4 Semantic Similarity v. Measured Similarity

A similarity measure is a tool that quantifies the degree to which two elements are similar, using the information about these elements in the dataset. The question is: to what extent does this calculated similarity relate to the actual, or semantic similarity between the two elements? Many different aspects can cause a difference between the semantic similarity and the calculated similarity. To get a good understanding of this, it is useful to first take a closer look at what this “semantic similarity” actually is.

The semantic similarity between two elements depends on the goals of the user. Consider a dataset with animals and their characteristics. If the user wants to divide these animals in classes like 'mammals,' 'birds,' 'fishes,' only some rough characteristics of the animals are needed. For instance, when two animals both have a spine, breathe air, have hair on their body, and breast-feed their young offspring, they can be considered similar, namely mammals. Another dataset with characteristics of animals may be used by park rangers to keep track of the movement of the more special animals (e.g. lions, rhinos, and elephants). In this case, to conclude whether two animals from different sightings are the same, it is necessary to look at more detailed characteristics such as individual markings or scars.

The semantic similarity between two elements also depends on the context: the other elements in the data set. Consider a dataset with texts where the user wants to group the texts that are about the same subject. When there are two texts about sport, whether or not they should be in the same group depends on the subjects of the other texts in this dataset. If the dataset consists of texts from different categories such as 'sport,' 'science,' 'romance,' or 'politics,' two texts about sport should be considered similar (i.e. about the same subject) and grouped together. On the other hand, if all the texts in the dataset are about sport, then categories can be 'football,' 'rowing,' or 'cycling,' or maybe those categories are 'professional,' 'recreational,' 'tactics,' or 'training,'.

In the previous cases, the semantic similarity was either TRUE or FALSE. Two animals are the same, or they are not the same; two texts are about the same subject, or not. In other occassions, the semantic similarity can be on a more gradual scale. Consider once again the dataset with texts about 'sport,' 'science,' 'romance,' 'politics,' and so on. When a text is about the love between two athletes, its subject is somewhere between 'romance' and 'sport,' making it difficult to determine exactly in which category this text belongs.

To conclude, the semantic similarity can be defined as "the actual similarity, the goal for the similarity measure, depending on the task and the context." Still, this definition is not very precise. In actuality, it is impossible to give a precise definition, because the semantical similarity is a very abstract concept. As a result, it is not straightforward how a similarity measure that calculates a value describing this semantical similarity needs to be defined. Besides that, there are other difficulties that arise when constructing a similarity measure, and which can cause a difference between the calculated similarity and the semantic similarity.

First of all, as described in Section 4.2, there is the principle of content variability. This states that the contents of elements from the same class vary around a center. A similarity measure uses this content to calculate how similar two elements are. So, when the contents of all elements in the same class vary and these contents are used to calculate the similarity, the similarities from one element to all elements in the same class also will vary.

Another problem is that it is sometimes difficult for a computer to get the semantic meaning of an element. Consider again a dataset with texts. For a human it is quite easy to understand what the subject of a text is, but for a computer this is much more difficult. One method to calculate the similarity between two texts is to calculate the cosine of the angle between their 'keyword vectors.' A keyword vector is a vector where each element of the vector represents a word that appears in any of the texts. The value for such an element is the amount of appearances of that specific word in that specific text. Two texts about the same subject have relatively more words in common, and thus the cosine of the angle between their keyword vectors will be higher.

The problem that arises is that the meaning of a word depends on the context. Consider, for instance the next two sentences: "A pitcher ducks when a bat flies to him." and "Ducks, flies and a bat drank from the pitcher." These sentence have relatively many words in common, but their meanings are very different. The opposite can also be true. Consider the sentences "I made an appointment with her by telephone" and "I called her so we could meet." These sentences have hardly any words in common, but mean roughly the same.

All in all, there is something like a "semantic similarity," which is the true similarity, and the goal of the measured similarity is to approach this semantic similarity as well as possible. This is difficult because the semantic similarity also depends on the goal of the user and the other elements in the dataset. Furthermore, grasping the semantic similarity in a numerical value could be difficult anyway. This difficulty arises because the contents of elements from the same class may vary around a mean, resulting in difficulties for a computer to get the semantic meaning of an element.

4.5 Hybrid Similarity v. Original Similarity

Following the definitions in Section 3.3, the dataset $D = (V, E, \alpha, \lambda)$ consists of V, a set of elements; E, a set of edges connecting these elements; α, a function that assigns an annotation to each element; and λ, a function that assigns a label to each element. The space where the semantic, or real, objects live will be called $\mathcal{R}$. An element $v \in \mathcal{R}$ is represented using an annotation $a \in \mathcal{A}$. We distinguish between the semantic similarity $\mathcal{S}_{\mathcal{R}} : \mathcal{R} \times \mathcal{R} \to \mathbb{R}$, and the similarity measure used by the DM tool $\mathcal{S}_{\mathcal{A}} : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$.

There is a difference between the real similarity $\mathcal{S}_{\mathcal{R}}$ and the measured similarity $\mathcal{S}_{\mathcal{A}}$. This difference can be considered 'noise.' Noise can arise in a dataset because it is difficult for a computer program to exactly get the semantic meaning of an element. This has been discussed previously in this chapter. Another possibility for noise to appear in a dataset is by errors that occur while processing the data.

Users of DM tools are interested in the real similarities, but since these similarities cannot be formalised, the measured similarity needs to be used.

Thus, it is important that the difference between $\mathcal{S}_{\mathcal{R}}$ and $\mathcal{S}_{\mathcal{A}}$ is as small as possible. This could be done by altering $\mathcal{S}_{\mathcal{A}}$ to make it more befitting $\mathcal{S}_{\mathcal{R}}$, but when this is not possible, other methods should be used to reduce the gap between these similarities. The hybrid similarity measures as described in Chapter 3 can do such a thing. To show this, we compare the content-based similarity with the neighborhood similarity as described in (3.2). To emphasize that they are both similarities that are calculated using the annotations, from now on they will be called $\mathcal{S}_{\mathcal{A}1}$ and $\mathcal{S}_{\mathcal{A}2}$, respectively.

The content-based similarity $\mathcal{S}_{content}$ is an estimator of the real similarity. Let ϵ denote the difference between this estimator and what it estimates. We then have

$$\mathcal{S}_{\mathcal{A}1}(v,w) = \mathcal{S}_{content}(v,w) = \mathcal{S}_{\mathcal{R}}(v,w) + \epsilon(v,w). \tag{4.1}$$

Any distribution over pairs of annotated nodes gives rise to distributions of $\mathcal{S}_{\mathcal{R}}$, $\mathcal{S}_{\mathcal{A}}$ and ϵ. If the estimator is unbiased, ϵ has a distribution centered around 0, so the expected outcome will be 0: $\mathbb{E}[\epsilon(v,w)] = 0$. Note that any estimator can be made unbiased by subtracting its bias from it. Therefore, in the following, we simply will assume that the estimator is unbiased. Hence $\mathbb{E}[\epsilon(v,w)] = 0$. Finally, we denote the variance of this distribution by σ_ϵ^2.

Now consider, as a second estimator, the neighbor similarity:

$$\begin{aligned} \mathcal{S}_{\mathcal{A}2}(v,w) = S_{neighbor}(v,w) &= \frac{\sum_{u\in\mathcal{N}(w)} \mathcal{S}_{content}(v,u)}{|\mathcal{N}(w)|} \\ &= \frac{\sum_{u\in\mathcal{N}(w)} \mathcal{S}_{\mathcal{R}}(v,u)}{|\mathcal{N}(w)|} + \frac{\sum_{u\in\mathcal{N}(w)} \epsilon(v,u)}{|\mathcal{N}(w)|} \end{aligned} \tag{4.2}$$

From this, consider the conditional distribution of $\mathcal{S}_{\mathcal{R}}(v,u)$, with $u \in \mathcal{N}(w)$ given $\mathcal{S}_{\mathcal{R}}(v,w)$. The semantic similarity $\mathcal{S}_{\mathcal{R}}(v,u)$ is likely dependent on $\mathcal{S}_{\mathcal{R}}(v,w)$ since w and u are connected. More specifically, due to homophily and transitivity, we expect them to be positively correlated. Let us write for $u \in \mathcal{N}(w)$:

$$\mathcal{S}_{\mathcal{R}}(v,u) = \mathcal{S}_{\mathcal{R}}(v,w) + \xi(w,u) \tag{4.3}$$

where ξ denotes the difference between the real similarity between v and w on the one hand, and the real similarity beween v and u on the other hand. This formula is generally valid (by definition of ξ), but in the presence of homophily and transitivity, we additionally expect ξ to be small.

Using (4.3) to substitute $\mathcal{S}_{\mathcal{R}}(v,u)$ in (4.2) gives

$$\mathcal{S}_{\mathcal{A}2}(v,w) = \mathcal{S}_{\mathcal{R}}(v,w) + \frac{\sum_{u\in\mathcal{N}(w)} \xi(w,u)}{|\mathcal{N}(w)|} + \frac{\sum_{u\in\mathcal{N}(w)} \epsilon(v,u)}{|\mathcal{N}(w)|}. \tag{4.4}$$

Observe that, among v, w and all $u \in \mathcal{N}(w)$, there is no relation except for the fact that w is connected to each element from $\mathcal{N}(w)$. This entails the following. For symmetry reasons, the distribution of $\epsilon(v,u)$ must be equal to that of $\epsilon(v,w)$ (from v's point of view, u is just a random vertex, just like w). For each u, $\epsilon(v,u)$ therefore has an expected value of 0 and a variance of σ_ϵ^2.

Again because of symmetry, there is no reason to believe that, on average, $\mathcal{S}_\mathcal{R}(v,w)$ should be greater or smaller than $\mathcal{S}_\mathcal{R}(v,u)$. This means that $\mathbb{E}[\xi(w,u)] = 0$. Since all $u \in \mathcal{N}(w)$ are also interchangeable among themselves, all $\xi(w,u)$ have the same distribution. We denote the variances of these as σ_ξ^2.

Now consider the case where all $\epsilon(v,u)$ are independent, all $\xi(w,u)$ are independent, and $\epsilon(v,u)$ is independent from $\xi(w,u)$ for all $u \in \mathcal{N}(w)$. For this special case, we obtain the following squared errors for the two estimators:

$$SE(\mathcal{S}_{\mathcal{A}1}(v,w)) = \mathbb{E}[(\mathcal{S}_{\mathcal{A}1}(v,w) - \mathcal{S}_\mathcal{R}(v,w))^2] = \sigma_\epsilon^2 \tag{4.5}$$

$$SE(\mathcal{S}_{\mathcal{A}2}(v,w)) = \mathbb{E}[(\mathcal{S}_{\mathcal{A}2}(v,w) - \mathcal{S}_\mathcal{R}(v,w))^2] = \frac{\sigma_\epsilon^2 + \sigma_\xi^2}{|\mathcal{N}(w)|} \tag{4.6}$$

It is now obvious that the first estimator has an expected squared error of σ_ϵ^2, and the second estimator has an expected squared error of $(\sigma_\epsilon^2 + \sigma_\xi^2)/|\mathcal{N}(w)|$. This second term can be larger or smaller than the first. However, it tends to become smaller as $|\mathcal{N}(w)|$, the number of neighbors, increases. When (and whether) it becomes smaller than σ_ϵ^2 depends on the relative size of σ_ξ^2.

Note that, intuitively, σ_ϵ^2 is related to content variability (it reflects the extent to which the observed content similarity between two nodes approximates their real similarity), whereas σ_ξ^2 is related to the amount of homophily and transitivity in the network (it expresses to what extent nodes that are linked together have comparable similarities to other nodes). In networks with strong homophily and high content variability, the second estimator can be expected to be more accurate than the first.

The case where ϵ and ξ are not independent is mathematically more complex. We already have σ_ϵ^2 and σ_ξ^2 for the variance of $\epsilon(v,u)$ and $\xi(w,u)$. Now for every $u_i, u_j \in \mathcal{N}(w), u_i \neq u_j$, we denote the covariance between $\epsilon(v,u_i)$ and $\epsilon(v,u_j)$ as $\textsc{Cov}_\epsilon$, the covariance between $\xi(w,u_i)$ and $\xi(w,u_j)$ as $\textsc{Cov}_\xi$, and the covariance between $\epsilon(v,u_j)$ and $\xi(w,u_i)$ as $\textsc{Cov}_{\epsilon\xi}$. Then, using the rule that $\textsc{Var}(\sum_i X_i) = \sum_i \textsc{Var}(X_i) + \sum_{i,j\neq i} \textsc{Cov}(X_i, X_j)$, and exploiting the linearity of $\textsc{Var}$ and $\textsc{Cov}$, one quickly arrives at

$$SE(\mathcal{S}_{\mathcal{A}2}(v,w)) = \\ \frac{\sigma_\epsilon^2}{|\mathcal{N}(w)|} + \frac{\sigma_\xi^2}{|\mathcal{N}(w)|} + \frac{|\mathcal{N}(w)|-1}{|\mathcal{N}(w)|}\textsc{Cov}_\epsilon + \frac{|\mathcal{N}(w)|-1}{|\mathcal{N}(w)|}\textsc{Cov}_\xi + \textsc{Cov}_{\epsilon\xi} \tag{4.7}$$

which shows that strong positive autocorrelations of ξ and ϵ are likely to spoil the advantage of using $\mathcal{S}_{\mathcal{A}2}$. Such correlations are not impossible. For instance, $\epsilon(v,u_1)$ may be high because $\alpha(v)$ deviates strongly from its expected value (due to content variability), which makes $\epsilon(v,u_2)$ more likely to be high too. It is difficult to estimate how large this effect can be. On the other hand, if

$\epsilon(v, u_1)$ is high because $\alpha(v)$ deviates strongly from its expected value, it means that the content-based similarity as an estimator is very bad and there is much room for improvement. Also, this means that there is a good chance that, due to homophily and transitivity, the average similarity of w with all elements in the neighborhood of v is a much better estimator for the semantic similarity. Since this average is defined as $\mathcal{S}_{neighbor}(w, v)$, which is also incorporated in $\mathcal{S}_{contextual}$, it can be expected that the contextual similarity will be a better estimator for the semantic similarity.

Finally, note that $S_{context}(v, w)$ is the average value of the two similarities $S_{neighbor}(v, w)$ and $S_{neighbor}(w, v)$. Hence, it is likely to be slightly more accurate than $\mathcal{S}_{\mathcal{A}2}$ (its standard error is $\sqrt{2}$ times smaller, in case of independence of the error terms for v and w). This is again due to the error-reducing effect of averaging. The similarity $S_{combined}$, being the average of the similarities $S_{content}$ and $S_{context}$, can be more accurate than either or them, or have an accuracy somewhere in between.

4.6 Summary

Many data mining tools use a measurement to compare different elements in a data set. Three main categories for these measurements can be distinguished: distances, similarities and dissimilarities. It is difficult for a computer program to correctly calculate the real, semantic similarity between two elements. This will result in a difference between the semantic similarity and the measured similarity. There are several different causes for this:

- The semantic similarity can depend on the goals of the user.
- The semantic similarity can depend on the context in the data set.
- The semantic similarity can be on a gradual scale between two categories.
- Elements in the same class have a variety of content.
- The meaning of a single element can be difficult for a computer already.

All these problems make it difficult for a similarity measure to calculate the similarity between elements correctly. The hybrid similarity measure proposed in this thesis tries to improve the quality of the previously existing similarity measures by exploring the environment of an element. Due to homophily, it can be expected that the environment of an element tends to consist of elements of the same class. By taking the average similarity of the environment, problems that occur during the calculation of the similarity of a single element can average out, resulting in a more accurate similarity.