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

Chapter 8

Reducing the Impact of Content Variability

The hybrid similarities give a better approximation of the semantic similarity between two elements. Theoretical analysis showed that the hybrid similarities use homophily to compensate for the content variability in the dataset. Experiments on a synthetic dataset, where these two concepts can be regulated, can provide an experimental validation for this. Experimental research will show that the hybrid similarities are less susceptible to content-based noise.

8.1 Introduction

Previous chapters showed that the hybrid similarities can be used to improve existing data mining techniques. Chapter 4 gave a theoretical motivation for why it could be that the hybrid similarities can get better results than the content-based similarity. It stated that, in a dataset where there is much homophily, the neighborhood of an element can also be a good estimator for that element. This is especially the case when there is much content variability in the dataset, so the content of a single element could deviate a lot from the expected form.

In this chapter we present an experimental approach to investigate the influence of the amount of homophily and content variability in a dataset on the performance of the hybrid similarities with respect to the results of a content-based similarity. The amount of homophily can be adjusted by changing relational information, and the amount of content variability can be adjusted by changing content-based information. When this is done in real-life data, it is difficult to predict whether a particular change in the content/relations will lead to an increase or a decrease of the content variability/homophily. Therefore, in order to regulate the amount of homophily and content variability in a controlled fashion, a synthetic dataset was created.

The rest of this chapter is organized as follows. Section 8.2 explains the problem that we are trying to investigate in more detail. Section 8.3 gives the experimental setup, where Section 8.3.1 describes the used synthetic dataset and Section 8.3.2 briefly describes the used clustering methods. Section 8.4 shows the results from the experiments and to finalize, Section 8.5 draws the conclusions from these results.

8.2 Problem Setting

It can be difficult for a computer program to get the real, semantic meaning of an element precisely. There can be many reasons for that, which are described in Section 4.4. When the computer has the wrong 'understanding' of the semantics of a particular element, it will most probably calculate the similarities between that element and others incorrectly. As a result, data mining techniques that use these similarities will make wrong decisions regarding this element.

Since there is a difference between observed similarity and semantic similarity, and since we are really interested in semantic similarity, the question arises: can we find a better approximation to the semantic similarity than this observed similarity? This would of course be possible by changing the annotation space $\mathcal{A}$, making it richer. However, we assume that $\mathcal{A}$ and the data elements in it are given and cannot be changed. (The question of how to change $\mathcal{A}$ is essentially the feature construction problem, which is orthogonal to the method proposed in this thesis.) Fortunately, when the elements are also linked together, we may be able to obtain a better approximation of the semantic similarity by taking the link structure into account.

It could be that the annotation of a particular element is a bad approximation of the actual, real-life content of that element, and thus any measured similarity between that element and any other element is a bad approximation of the semantic similarity. In many natural datasets, linked elements tend to be more similar, and similar elements tend to be linked. This principle is called homophily and is described more elaborately in Section 4.2. This means that the neighborhood of an element usually consists of elements that are similar, and thus from the same class. So, in the case of an element where its annotation is a bad approximation of its actual content, the annotations of its neighbors might be better approximations than its own annotation. Therefore, in this case, it could be good to also take the similarities to the neighbors of this element into account.

The hybrid similarities can create a better approximation of the semantic similarity. This has been described more elaborately in Chapter 4. However, in this chapter, we aim to give an experimental validation for this. We try to do so by measuring what the influence is of these bad annotations on the hybrid similarities, compared to their influence on a content-based similarity.

8.3 Experimental Setup

8.3.1 The Synthetic Dataset

In order to test and characterize the relative performance of $S_{content}$, $S_{context}$, and $S_{combined}$, the similarity measures described in Section 3.4, and to determine under which circumstances which measure works best, we have created a synthetic data set in which we can vary the amount of content variability and homophily in the network. Roughly, one could think of the dataset as a set of documents; each document is represented as a Boolean vector that shows which words occur in the document, and documents may be linked to each other. The documents are organized into classes.

We start with a dataset with zero content variability and perfect homophily. Zero content variability means that each document of a particular class is the same (has all the words characteristic for the class, and no words characteristic for other classes). Perfect homophily means that within a class all documents are linked to each other, and there are no links between classes. Afterwards, we increase content variability by removing words from documents and introducing words from other classes into them. Similarly, we decrease homophily by introducing links between documents of different classes and removing links between documents from the same class. Our goal is to investigate the effect of these changes on the accuracy of the similarity measures.

More specifically, our dataset $D = (A, G)$ consists of 1000 elements. It is described by two matrices: A, which denotes the content-based information, and G which denotes the relational information. The content of an element is described by a binary vector with 1000 components, each of which represents one particular word; each class has 100 words that are characteristic for it, and each word is characteristic for only one class. All these vectors are combined in A which is a 1000×1000 matrix where each element a_{ij} is a Boolean that states whether the j-th word appears in the content of i-th element. The relational information is described in the 1000×1000 adjacency matrix G where each element g_{ij} is a Boolean that states whether the i-th and the j-th element are connected.

The elements are divided into 10 classes with 100 elements each. The elements are arranged in an orderly manner, so the class of an element v_i is defined by $class(v_i) = \lfloor \frac{i-1}{100} \rfloor$. The same holds for the words that are characteristic to a class, so every word a_i is characterizing class for a word is defined by $class(a_i) = \frac{i-1}{100} \rfloor$. In the original dataset, there is perfect homophily, and no content variability, so A and G are both block matrices. This means that the elements a_{ij} from A are 1 when $class(v_i) = class(a_j)$, and 0 otherwise.

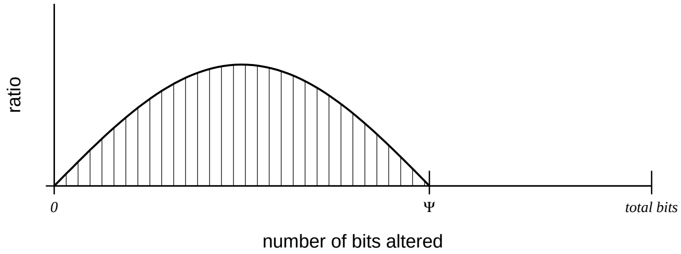

Figure 8.1: Distribution of the amount of bits that will be flipped.

Similarly, the elements g_{ij} from G are 1 when $class(v_i) = class(v_j)$, and 0 otherwise.

Now, increasing content variability means flipping bits in A, and decreasing homophily means flipping bits in G. The amount of bits that is flipped is determined by a parameter p (where relevant, we write p_A and p_G for the corresponding parameter for A and G), with $0 \leq p \leq 1$. Originally, we flipped bits randomly, giving each single bit a probability p of being flipped. Unfortunately, this led to artifacts in the results. These artifacts were probably caused by the fact that each document has approximately the same amount of flipped bits (a narrow binomial distribution around 1000p). Therefore, instead, we have chosen p to be the maximum fraction of bits flipped, but to let the actual number of flipped bits vary between 0 and $\Psi = 1000p$, according to a sinus-shaped distribution:

$$f(x) = \begin{cases} \frac{\pi}{2\Psi} \sin\left(\frac{\pi x}{\Psi}\right) & \text{if } x \in [0, \Psi] \\ 0 & \text{otherwise} \end{cases} \tag{8.1}$$

This distribution (visualized in Figure 8.1) was chosen ad hoc; the actual distribution does not matter much, the important thing is to have an easily computable distribution that is not too narrow and gradually approaches zero near the borders.

We will alter p_A and p_G separately. We expect that increasing p_A (content variability) renders the content-based similarity less accurate, while the other similarities suffer less from this. Conversely, increasing p_G is expected to have no influence on content based similarity, but cause the other similarities to deteriorate.

8.3.2 Clustering Methods

We can evaluate the practical usefulness of the similarity measures by using them to perform clustering; a similarity measure is considered more relevant if it yields clusters that are closer to the actual (hidden) classes. We have used three different clustering algorithms: agglomerative hierarchical clustering using average linkage (hereafter referred to as agglomerative clustering), an approximate version of k-means (k-means-NAMA (See Chapter 6), hereafter referred to as k-means), and k-medoids.

We express the quality of a clustering using the *F-measure*, as suggested by Larsen and Aone [56]. This is the harmonic mean between the precision and the recall of a cluster/label-combination. It is described in detail in Section 5.4. Since there are 10 classes in the data set, we choose $k = 10$ for k-means and k-medoids, while the hierarchical clustering is simply cut off at 10 clusters.

8.4 Results

8.4.1 Results on the Synthetic Dataset

We ran experiments for many combinations of p_A and p_G. In the Figures 8.2 through 8.4 the overall results are presented. For any particular combination of p_A and p_G, they also show which similarity had the highest F-score. Of course, changing the value for p_G should have no effect on the performance of the content-based similarity since it does not use the relational information. Any small differences in results are due to random choices that the algorithm needs to make. In the case of k-means and k-medoids, this is the choice for the initial k prototypes. In the case of agglomerative hierarchical clustering, this is the choice which two clusters to merge when there is more than one pair of elements that is the most similar.

For agglomerative hierarchical clustering, Figure 8.2 shows that there is not much difference between the results of the content-based similarity and the combined similarity, but the contextual similarity behaves very differently. When there is little content-based noise ($p_A \leq 0.3$), the content-based and combined similarity reach the perfect score, but the contextual similarity has difficulties as the relational noise increases. On the other hand, when there is only a small amount of relational noise, the contextual similarity starts to outperform the two others when p_A increases.

Figure 8.3 shows the results for k-means. To some extent, they are comparable to the results for agglomerative hierarchical clustering, but the resulting behaviour of the contextual similarity is even more profound here. Once again, the content-based similarity and the combined similarity do not differ much. They are both hardly influenced by the relational noise. Also, they perform very well for low values of p_A, and start to drop when p_A increases. The difference here is that the results of the combined similarity do not drop so fast as those of the contextual similarity (especially when p_G is low).

The contextual similarity behaves very differently. It is hardly influenced by noise on the content, except when the relational noise is high. Furthermore, it performs very well for low values of the relational noise and starts decreasing

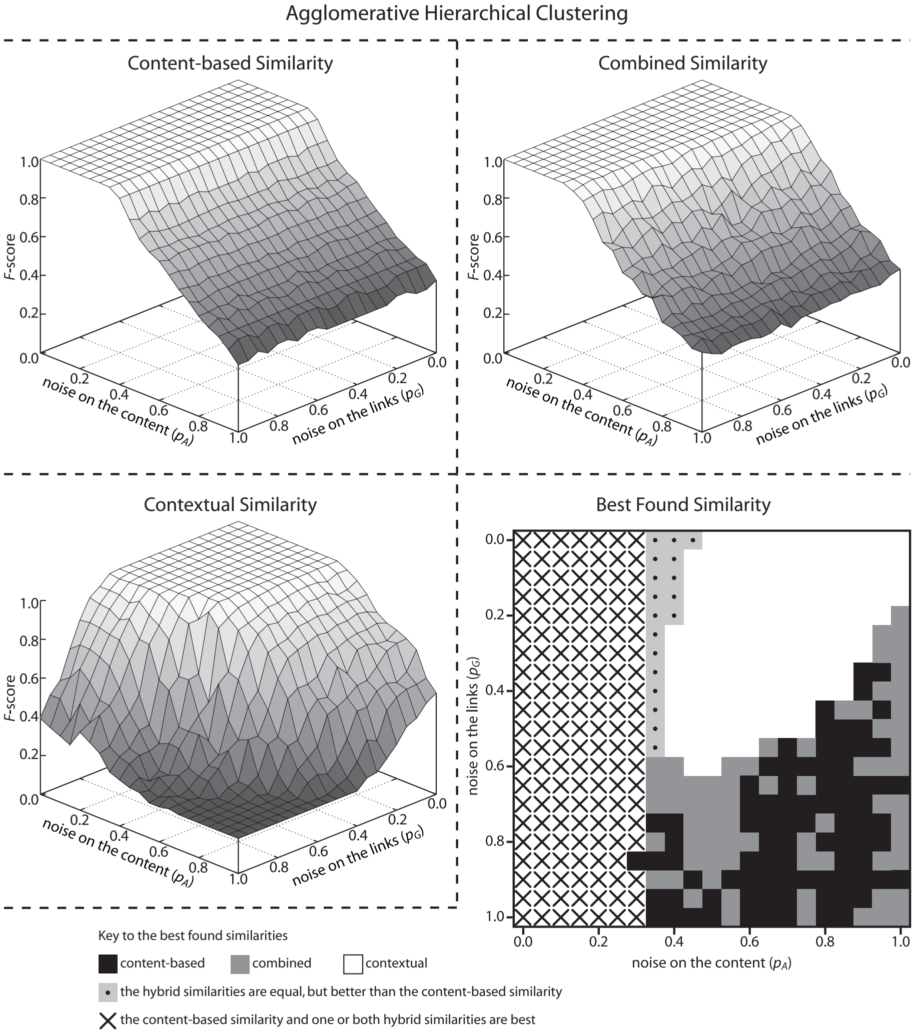

Figure 8.2: Results for agglomerative hierarchical clustering. The three-dimensional graphs give the found F-scores for that combination of p_A and p_G. The two-dimensional graph tells which similarity had the highest F-score for each combination of p_A and p_G. Special symbols are used when more than one similarity is the best. The vertical axis in the bottom-right panel is reversed to keep the same orientation as the three-dimensional graphs.

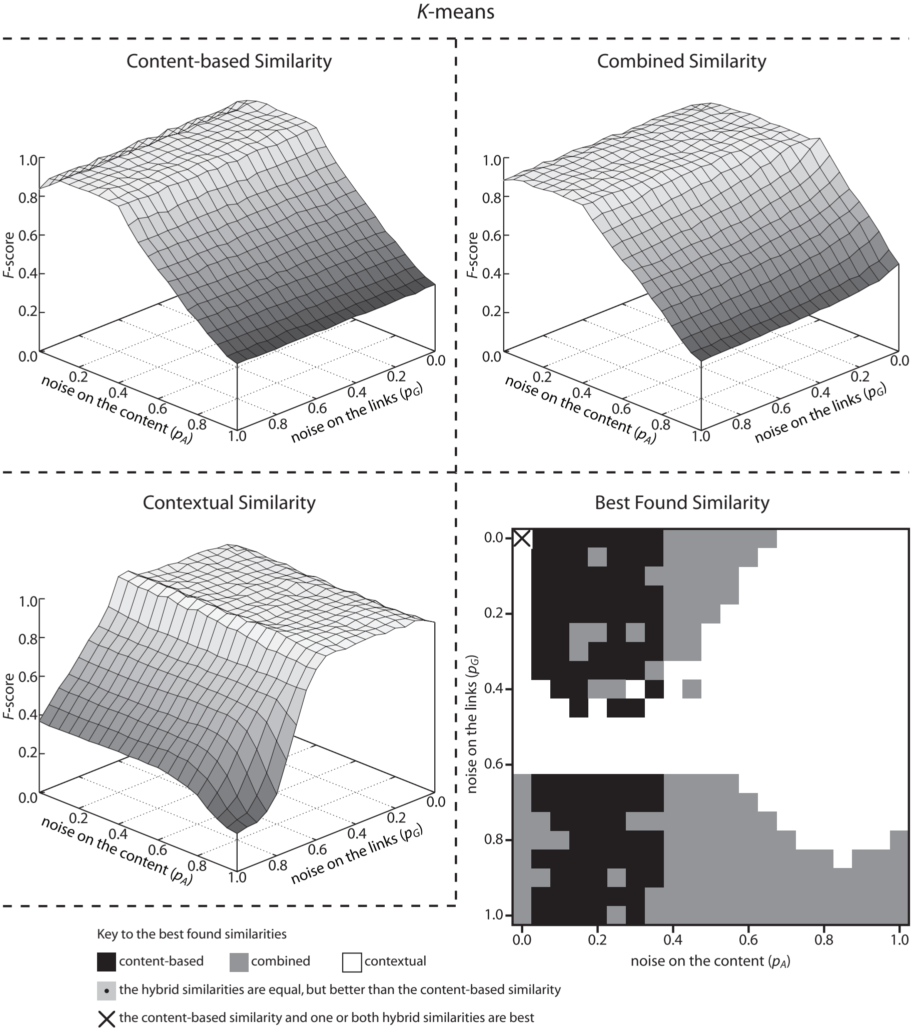

Figure 8.3: Results for k-means. The three-dimensional graphs give the found F-scores for that combination of p_A and p_G. The two-dimensional graph tells which similarity had the highest F-score for each combination of p_A and p_G. Special symbols are used when more than one similarity is the best. The vertical axis in the bottom-right panel is reversed to keep the same orientation as the three-dimensional graphs.

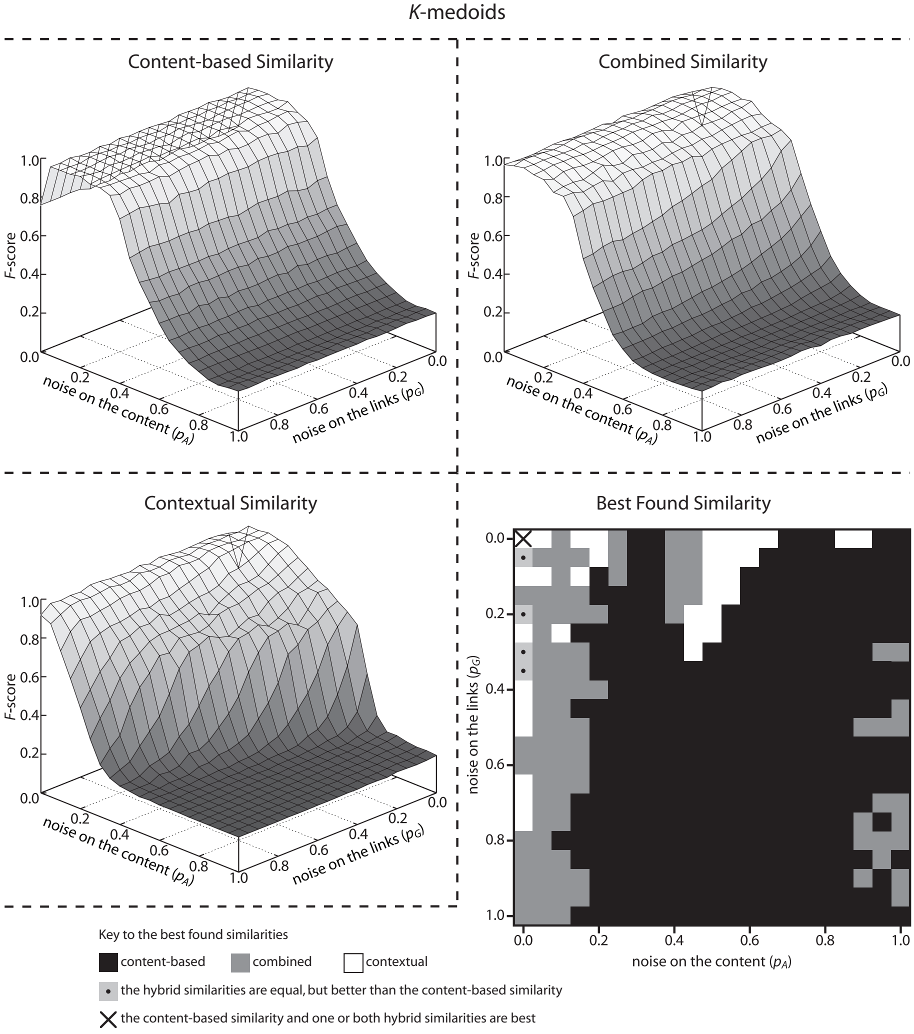

Figure 8.4: Results for k-medoids. The three-dimensional graphs give the found F-scores for that combination of p_A and p_G. The two-dimensional graph tells which similarity had the highest F-score for each combination of p_A and p_G. Special symbols are used when more than one similarity is the best. The vertical axis in the bottom-right panel is reversed to keep the same orientation as the three-dimensional graphs.

when this noise is high ($p_G > 0.6$). There is a peak in performance in a small range ($0.5 \leq p_G \leq 0.6$) where the contextual similarity outperforms the other similarities, even for low values of p_A. It is uncertain what causes this peak.

Figure 8.4 shows the results for k-medoids. These figures show similar characteristics. Again, the content-based similarity and the combined similarity do not differ much from each other; they both hardly are influenced by the amount of relational noise, both perform very well with low content-based noise, and both their performances start to drop when p_A rises. The difference this time is that here, the content-based similarity slightly outperforms the combined similarity most of the time.

On the other hand, the behaviour of the contextual similarity is very different from what we have seen before. It performs poorly, and only in a few cases does it manage to get the best F-score. This is in the area where relational noise is low, and there is a fair amount of content-based noise.

8.4.2 Results on Real Data

The above analysis used synthetic data, where homophily and content variability were controlled. It is clear that, depending on the amount of homophily and content variability, one method will be better than the other. The question remains in which areas certain real-life datasets lie. Earlier work with the Cora dataset (See Chapters 5 through 7) has shown that the area where contextual and combined similarity perform better is certainly not empty. This, in turn, implies that some networks indeed contain enough homophily to make the use of hybrid similarities rewarding.

8.5 Conclusion

When estimating the similarity between two annotated nodes in a graph, one can simply rely on the nodes' annotations. However, we have argued that it may be useful to take also the network context into account. Using a synthetic dataset, we have observed experimentally what the effects of homophily and content variability on the different similarities are. The precise effects differ from one data mining technique to another.

Several general conclusions can be drawn. First, the content-based similarity and the combined similarity do not differ much in behaviour. Second, the contextual similarity outperforms the others when the amount of relational noise is small, and the amount of content-based noise is high. This shows that this similarity measure indeed behaves as expected by our theoretical analysis. This confirms and explains why similarity measures that look not only at content but also at graph context are of practical importance.

The question that rises is: to what extent does the synthetic dataset behave as a real-life dataset? The experiments on subsets of the Cora dataset, as described in previous chapters, show some important differences. First of all, on Cora, the two hybrid similarities behaved more or less the same. This is opposed to the synthetic dataset, where it is the content-based and the combined similarity that behave similarly. Secondly, during the experiments on the subsets of Cora, the hybrid similarities outperformed the content-based similarity. However, with the synthetic dataset, this was not the case.

This could mean that the Cora dataset probably has the ratio of homophily and content variability where the hybrid similarities outperform the content-based similarity. The region where this is consistent for the three data mining techniques is where there is much homophily (p_G is low) and much content variability (p_A is high). In the Cora dataset, the content is defined by the words that are in the abstracts. Since abstracts in general are relatively small pieces of text, it is likely that there is indeed much content variability in the Cora dataset. The relational information comes from the citations. Usually, most citations from or to a paper are of papers that are about the same subject. Therefore it is also likely that there is much homophily in the Cora dataset. All in all, it seems that Cora is in the right range for the hybrid similarities to perform well.