Heterogeneous network decomposition and weighting with text mining heuristics

Jan Kralj[1,2], Marko Robnik-Šikonja[3] and Nada Lavrač[1,2,4]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia
[3] Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia
[4] University of Nova Gorica, Vipavska 13, 5000 Nova Gorica, Slovenia
jan.kralj@ijs.si, marko.robnik@fri.uni-lj.si, nada.lavrac@ijs.si

Abstract. The paper presents an approach to mining heterogeneous information networks applied to a task of categorizing customers linked in a heterogeneous network of products, categories and customers. We propose a two step methodology to classify the customers. In the first step, the heterogeneous network is decomposed into several homogeneous networks using different connecting nodes. Similarly to the construction of bag-of-words vectors in text mining, we assign larger weights to more important nodes. In the second step, the resulting homogeneous networks are used to classify data either by network propositionalization or label propagation. Because the data set is highly imbalanced we adapt the label propagation algorithm to handle imbalanced data. We perform a series of experiments and compare different heuristics used in the first step of the methodology, as well as different classifiers which can be used in the second step of the methodology.

Keywords: · Network analysis · Heterogeneous information networks · Network decomposition · PageRank · Text mining heuristics · Centroid classifier · SVM

1 Introduction

The field of *network analysis* is well established and exists as an independent research discipline since the late seventies [24] and early eighties [1]. In recent years, analysis of *heterogeneous information networks* [20] has gained popularity. In contrast to standard (homogeneous) information networks, heterogeneous networks describe heterogeneous types of entities and different types of relations.

This paper addresses the task of mining *heterogeneous information networks* [20]. In particular, for a user-defined node type, we use the method of classifying network nodes through network decomposition; this results in homogeneous networks whose links are derived from the original network. Following [6], the method constructs homogeneous networks whose links are weighed with the number of intermediary nodes, connecting two nodes. After the individual homogeneous networks are constructed, we consider two approaches for classification of network nodes. We classify the nodes either through label propagation [25], or using a propositionalization approach [6]. The latter allows the use of standard classifiers such as the centroid and SVM classifier on the derived feature vector representation. The propositionalization approach was already applied to a large heterogeneous network of scientific papers from the field of psychology in our previous work [13]. In this work, we propose two improvements to the presented methodology: 1) a new variant of label propagation is proposed, resulting in improved classification performance on imbalanced data sets; 2) new heuristics for homogeneous network construction are proposed, which are inspired by word weighting heuristics used in text mining and information retrieval.

The paper is structured as follows. Section 2 presents the related work. Section 3 presents the two-stage methodology for classification in heterogeneous networks. We present a method for constructing homogeneous networks from heterogeneous networks and two methods for classification of nodes in a network: a network propositionalization technique and a label propagation algorithm. Section 4 presents how these two methods are further improved. We first introduce a variant of the label propagation algorithm, which improves performance on imbalanced data sets. We then show how the homogeneous network construction can be improved by using different weighting heuristics. Section 5 presents the application of the methodology on a challenge data set of customers linked to products they purchased, followed by the three stage analysis of the results. First, we examine the effect of using different classifiers in the final step of classification via propositionalization. Second, we test different heuristics for homogeneous network construction. Finally, we analyze the improved label propagation method for imbalanced data sets. Section 6 concludes the paper and presents the plans for further work.

2 Related work

In network data analysis, instances are connected in a network of connections. In ranking methods like Hubs and Authorities (HITS) [11], PageRank [18], SimRank [9] and diffusion kernels [12], authority is propagated via network edges to discover high ranking nodes in the network. Sun and Han [20] introduced the concept of *authority* ranking for heterogeneous networks with two node types (bipartite networks) to simultaneously rank nodes of both types. Sun et al. [21] address authority ranking of all nodes types in heterogeneous networks with a star network schema, while Grčar et al. [6] apply the PageRank algorithm to find PageRank values of only particular type of nodes.

In network classification, a typical task is to find class labels for some of the nodes in the network using known class labels of the remaining network nodes. A common approach is propagation of labels in the network, a concept used in [25] and [23]. An alternative approach is classification of network nodes through propositionalization, described in [6]. There, a heterogeneous network is

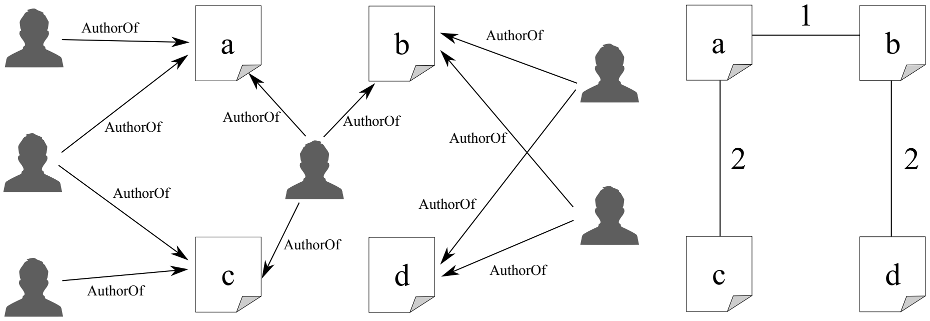

Fig. 1. An example of a heterogeneous network, decomposed into a homogeneous network where papers are connected if they share a common author. Weights of the edges are equal to the number of authors that contributed to both papers

decomposed into several homogeneous networks which are used to create feature vectors corresponding to nodes in the network. The feature vectors are classified by SVM [16, 14, 4], kNN [22] or centroid classifier [7] to predict class values of these nodes. The network propositionalization approach was also used in [13].

Our work is also related to text mining, specifically to bag-of-words vector construction. Here it is important to correctly set weights of terms in documents. Simple methods like term frequency are rarely used, as the term-frequency inverse-document-frequency (tf-idf) weighting introduced in [10] is more efficient. A number of weighting heuristics which also take into account labels of documents have been proposed, such as the χ^2, information gain [3], Δ-idf [17], and relevance frequency [15].

3 Methodology

This section addresses the problem of mining heterogeneous information networks, as defined by Sun and Han [20], in which a certain type of nodes (called the target type) is labeled. A two step methodology to mine class labeled heterogeneous information networks is presented. In the first step of the methodology, the heterogeneous network is decomposed into a set of homogeneous networks. In the second step, the homogeneous networks are used to predict the labels of target nodes.

3.1 Network decomposition

The original heterogeneous information network is first decomposed into a set of homogeneous networks, containing only the target nodes of the original network. In each homogeneous network two nodes are connected if they share a particular direct or indirect link in the original heterogeneous network. Take as an example a network containing two types of nodes, Papers and Authors, and two edge types, *Cites* (linking papers to papers) and *Written_by* (linking papers to authors). From it, we can construct two homogeneous networks of papers: the first, in which two papers are connected if one paper cites another, and the second, in which they are connected if they share a common author (shown in Figure 1). The choice of links used in the network decomposition step requires expert who takes the meaning of links into account and chooses only the decompositions relevant for a given task.

3.2 Classification

In the second step of the methodology, the homogeneous networks are used to classify the nodes. We compare two approaches to this task: the label propagation algorithm [25] and the network propositionalization approach [6].

Label propagation. The label propagation algorithm starts with a network adjacency matrix $M \in \mathbb{R}^{n,n}$ and a class matrix $Y \in \mathbb{R}^{n,|C|}$, where $C = \{c_1, \ldots, c_m\}$ is the set of classes, with which the network nodes are labeled. The j-th column of Y represents the j-th label of C, meaning that Y_{ij} is equal to 1 if the i-th node belongs to the j-th class and 0 otherwise. The algorithm constructs the matrix $S = D^{-\frac{1}{2}} M D^{-\frac{1}{2}}$, where D is a diagonal matrix and the value of each diagonal element is the sum of the corresponding row of M. The algorithm iteratively computes $F(t) = \alpha S F(t-1) + (1-\alpha)Y$ until there are no changes in the matrix $F(t)$. The resulting matrix F is used to predict the class labels of all unlabeled nodes in the network. Zhou et al. [25] show that the iteration converges to the same value regardless of the starting point $F(0)$. They also show that the value F^* that $F(t)$ converges to can also be calculated by solving a system of linear equations, as

$$F^* = (I - \alpha S)^{-1} Y. \tag{1}$$

To classify a heterogeneous network, decomposed into k homogeneous networks, we propose classification of nodes using *all* available connections from all k homogeneous network. We construct a new network with the same set of nodes as in the original homogeneous networks. The weight of a link between two nodes is calculated as the sum of link weights in all homogeneous networks. In effect, if the original networks are represented by adjacency matrices $M_1, M_2, \ldots, M_k$, the new network's adjacency matrix equals $M_1 + M_2 + \cdots + M_k$.

Classification by propositionalization. An alternative method for classifying the target nodes in the original heterogeneous network (called network propositionalization) calculates feature vectors for each target node in the network. The vectors are calculated using the personalized PageRank (P-PR) algorithm [18]. The personalized PageRank of node v (P-PR$_v$) in a network is defined as the stationary distribution of the position of a random walker who starts the walk in node v and then at each node either selects one of the outgoing connections or travels to the starting location. The probability (denoted p) of continuing the walk is a parameter of the personalized PageRank algorithm and is usually set to 0.85. Once calculated, the resulting PageRank vectors are normalized according to the Euclidean norm. The vectors are used to classify the nodes from which they were calculated.

For a single homogeneous network, the propositionalization results in one feature vector per node. For classifying a heterogeneous network, decomposed into k homogeneous networks, Grčar et al. [6] propose to concatenate and assign weights to the k vectors, obtained from the k homogeneous networks. The weights are optimized using a computationally expensive differential evolution [19]. A simpler alternative is to use equal weights and postpone weighting to the learning phase; due to the size of feature vectors in our experiments, we decided to follow this approach. Many classifiers, for example SVM classifier [16, 14, 4], kNN classifier [22] or a centroid classifier [7] can be used.

4 Methodology improvement

We present two improvements to the methodology described in Section 3. First, we describe handling of imbalanced data sets, then we present a novel edge weighting approach used in the construction of homogeneous networks from the original network.

4.1 Imbalanced data sets and label propagation

The label propagation algorithm, as defined in [25], works by simply propagating class labels from each member, belonging to a certain class. By doing so, it seems possible that the algorithm may have a tendency to over-estimate the importance of larger classes (those with more instances) in the case when data is imbalanced. For example, Figure 2 shows an example in which the label propagation algorithm will classify the central node as belonging to the first (larger) class, simply because it has three neighbors of class 1 and only two neighbors of class 2. This may, in some cases, not be the ideal outcome. It could be argued that the node we wish to classify is actually adjacent to *all* elements of class 1, but only *some* elements of class 2. Therefore, in this relative sense, class 1 nodes cast a stronger vote in favor of class 1 than nodes of class 2.

Using the reasoning described above, we see that there is a reason to believe that the label propagation approach may not perform well if the data is highly imbalanced, i.e., if the frequency of class labels are not approximately equal. We propose an adjustment of the label propagation algorithm i.e., to change the initial label matrix Y so that larger classes have less effect in the iterative process. The value of the label matrix Y in this case is no longer binary, but it is set to $\frac{1}{|c_j|}$ if node i belongs to class j and 0 otherwise.

If the data set is balanced (all class values are equally represented), then the matrix Y is equivalent to the original binary matrix multiplied by the inverse of the class value size. This, along with (1), means that the resulting prediction matrix only changes by a constant and the final predictions remain unchanged.

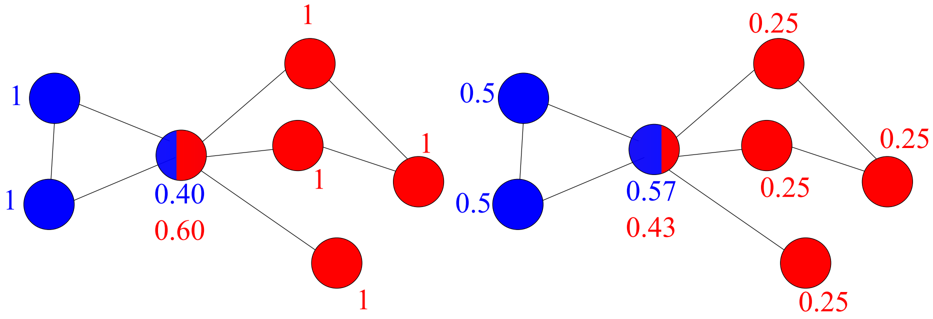

Fig. 2. Results of a label propagation algorithm on an imbalanced data set. If we run the label propagation algorithm as originally defined, each labeled node begin their iteration with a weight of 1 for the class they belong to. In each step of the iteration, every node collects the votes from its neighboring nodes and adds a portion (defined by α which was set to 0.6 in this example) of its original weight. We calculate that in this case, the central node receives a proportional vote of 0.40 from class 1 and a vote of 0.60 from class 2. However, using our modified weights, the labeled nodes start their iteration with a weight of $\frac{1}{2}$ for the class with 2 nodes and $\frac{1}{4}$ for the class with 4 nodes. Because of this, the proportion of votes for class 1 increases to 0.57. This is justified by the fact that the central node actually receives the highest possible vote that it can from a class consisting of only two nodes.

However, if the data set is imbalanced, smaller classes have a larger effect in the iterative calculation of F^*. This prevents votes of more frequent classes to outweigh votes of less frequent classes.

4.2 Text mining inspired weights calculation

We shortly present weighting of terms in the construction of bag-of-words (BOW) vectors and explain how the same ideas can be applied to extraction of homogeneous networks from heterogeneous networks.

Term weighting in text mining. In bag-of-words vector construction one feature vector represents each document in a corpus of documents. In that vector, the i-th value corresponds to the i-th term (a word or a n-gram) that appears in the corpus. The value of the feature depends primarily on the frequency of the term in the particular document. We describe several methods for assigning the feature values. We use the following notations: $f(t, d)$ denotes the number of times a term t appears in the document d and D denotes the corpus (a set of documents). We assume that the documents in the set are labeled, each document belonging to a class c from the set of all classes C. We use the notation $t \in d$ to describe that a term t appears in document d. Where used, the term $P(t)$ is the probability that a randomly selected document contains the term t, and $P(c)$ is the probability that a randomly selected document belongs to class c.

Scheme	Formula
`tf`	$f(t, d)$
`if-idf`	$f(t, d) \cdot \log\left(\frac{\lvert D\rvert}{\lvert\{d' \in D : t \in d'\}\rvert}\right)$
`chi^2`	$f(t, d) \cdot \sum_{c \in C}\left(\frac{(P(t \wedge c)P(\neg t \wedge \neg c) - P(t \wedge \neg c)P(\neg t \wedge c))^2}{P(t)P(\neg t)P(c)P(\neg c)}\right)$
`ig`	$f(t, d) \cdot \sum_{c \in C}\left(\sum_{c' \in \{c, \neg c\}}\left(\sum_{t' \in \{t, \neg t\}}\left(P(t', c') \cdot \log\frac{P(t' \wedge c')}{P(t')P(c')}\right)\right)\right)$
`delta-idf`	$f(t, d) \cdot \sum_{c \in C}\left(\log\frac{\lvert c\rvert}{\lvert\{d' \in D : d' \in c \wedge t \in d'\}\rvert} - \log\frac{\lvert\neg c\rvert}{\lvert\{d' \in D : d' \notin c \wedge t \notin d'\}\rvert}\right)$
`rf`	$f(t, d) \cdot \sum_{c \in C}\left(\log\left(2 + \frac{\lvert\{d' \in D : d' \in c \wedge t \in d'\}\rvert}{\lvert\{d' \in D : d' \notin c \wedge t \notin d'\}\rvert}\right)\right)$

Table 1. Term weighing in text mining.

Table 1 shows different methods for term weighting. The term frequency (`tf`) weights each term with its frequency in the document. The term frequency–inverse document frequency (`tf-idf`) [10] addresses the drawback of the `tf` scheme, which tends to assign high values to common words that appear frequently in the corpus. The χ^2 (`chi^2`) weighting scheme [3] attempts to correct another drawback of the `tf` scheme (one which is not addressed by the `tf-idf` scheme) by taking also class value of processed documents into consideration. This allows the scheme to penalize terms that appear in documents of all classes, and favor terms which are specific to some classes. Information gain (`ig`) [3] uses class labels to improve term weights. The Δ-idf (`delta-idf`) [17] and the Relevance frequency (`rf`) [15] attempt to merge the ideas of `idf` and both above class-based schemes by penalizing both common and non-informative terms.

Midpoint weighting in homogeneous network construction. Let us revisit the example from Section 3.1, in which two papers are connected by one link for each author they share. The resulting network is equivalent to a network in which two papers are connected by a link with a weight equal to the number of authors that wrote both papers (Figure 1). The method treats all authors equally which may not be correct. For example, if two papers share an author that only co-authored a small number of papers, it is more likely that these two papers are similar than if the two papers share an author that co-authored tens or even hundreds of papers. The first pair of papers should therefore be connected by a stronger weight than the second. Moreover, if the papers are labeled by the research field, then two papers, sharing an author publishing in only one research field, are more likely to be similar as if they share an author who has co-authored papers in several research fields. Again, the first pair of papers should be connected by the edge with larger weight.

Both described considerations are similar to the issues addressed in the term weighting schemes in document retrieval (presented at the beginning of this section). For example, the `tf-idf` weighting scheme attempts to decrease the weight of terms which appear in many documents, while we wish to decrease the weight of links, induced by authors which are connected to many papers. The `ig` weighting scheme decreases the weight of terms which appear in variously labeled documents, while we wish to decrease the weight of links induced by authors appearing in different research areas i.e., connected to variously labeled papers.

We alter the term weighting schemes in such a way that they can be used to set weights to midpoints in heterogeneous graphs (such as authors in our example). We propose that the weight of a link between two base nodes in the first step of the methodology (see Section 3) is calculated as the sum of weights of all the midpoints they share. In particular, if we construct a homogeneous network in which nodes are connected if they share a connection to a node of type T in the original heterogeneous network, then the weight of the link between nodes v and w should be equal to

$$\sum_{m \in T : (m,v) \in E \wedge (m,w) \in E} w(m), \tag{2}$$

where $w(m)$ is the weight assigned to the midpoint m. The value of $w(m)$ can be calculated in several ways. Table 2 shows the proposed midpoint weighting heuristics corresponding to term weighting used in document retrieval (Table 1). The notation used was as follows. We denote with B the set of all nodes of the base node type, and with E the set of all edges of the heterogeneous network. When m is a node, $P(m)$ denotes the probability that a random base node is connected to the midpoint node m. We assume that nodes are labeled, each belonging to a class c from the set of all classes C. We use $P(c)$ to denote the probability that a random base node is in class c. The term $P(c \wedge m)$ denotes the probability that a random base node is both in class c and linked to midpoint m.

Scheme	Formula
`tf`	1
`if-idf`	$\log\left(\frac{\lvert B \rvert}{\lvert \{b \in B : (b,m) \in E\} \rvert}\right)$
`chi^2`	$\sum_{c \in C} \frac{(P(m \wedge c)P(\neg m \wedge \neg c) - P(m, \neg c)P(\neg m, c))^2}{P(m)P(c)P(\neg m)P(\neg c)}$
`ig`	$\sum_{c \in C} \left(\sum_{c' \in \{c, \neg c\}} \left(\sum_{m' \in \{m, \neg m\}} P(m' \wedge c') \log\left(\frac{P(m' \wedge c')}{P(c')P(m')}\right) \right) \right)$
`delta-idf`	$\sum_{c \in C} \left\lvert \log \frac{\lvert c \rvert}{\lvert \{b' \in B : b' \in c \wedge (b', m) \in E\} \rvert} - \log \frac{\lvert \neg c \rvert}{\lvert \{b' \in B : b' \notin c \wedge (b', m) \notin E\} \rvert} \right\rvert .$
`rf`	$\sum_{c \in C} \left(\log\left(2 + \frac{\lvert \{b' \in B : b' \in c \wedge (b', m) \in E\} \rvert}{\lvert \{b' \in B : b' \notin c \wedge (b', m) \notin E\} \rvert} \right) \right)$

Table 2. Heuristics for weighting midpoints in homogeneous network construction.

The `tf` weight is effectively used in [6], where all authors are weighed equally. The `delta-idf` weighting scheme, unlike other term weighting schemes, may assign negative weights to certain terms. Since link weights in graphs are assumed to be positive both by the PageRank and the link propagation algorithm, we must change the weighting scheme before it can be used to construct homogeneous networks. We propose that in the original weighting scheme, terms which receive negative values are deemed informative, as they are informative about the term *not* being typical of a certain class. Therefore it is reasonable to take the absolute values of the weights in network construction.

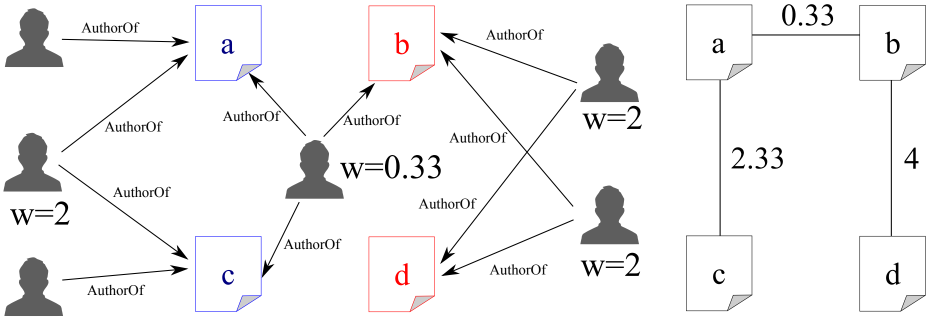

Fig. 3. The decomposition of the toy network from Figure 1 using the χ^2 heuristic. The blue color denotes that the paper belongs to class 1 and the red color denotes class 2.

Example 1. Figure 3 shows the decomposition of the network, seen in Figure 1, using the χ^2 heuristic. The weight of the central author m is calculated as the sum over both classes of

$$\frac{(P(m \wedge c)P(\neg m \wedge \neg c) - P(m, \neg c)P(\neg m, c))^2}{P(m)P(c)P(\neg m)P(\neg c)} \qquad (3)$$

When c is the first (blue) class, we can calculate the required values as $P(m \wedge c) = \frac{2}{4}, P(\neg m \wedge \neg c) = \frac{1}{4}, P(m, \neg c) = \frac{1}{4}$, $P(\neg m \wedge c) = 0$; $P(m) = \frac{3}{4}, P(\neg m) = \frac{1}{4}$, $P(c) = P(\neg c) = \frac{1}{2}$, yielding the first summand of 3 as

$$\frac{(\frac{2}{4} \cdot \frac{1}{4} - \frac{1}{4} \cdot 0)^2}{\frac{3}{4} \cdot \frac{1}{4} \cdot \frac{1}{2} \cdot \frac{1}{2}} = \frac{1}{3}.$$

When c is the second class, after calculating $P(m \wedge c) = P(\neg m \wedge \neg c) = P(m, \neg c) = P(\neg m \wedge c) = \frac{1}{4}$, we see that the second summand of 3 is 0 and the total weight of author m is $\frac{1}{3}$.

The weights of the remaining authors are calculated in the same way. In our case, none of the other authors wrote papers from both classes, so their weights are all equal to 2.

The decomposed network on the left is constructed by summing the weights of all common authors for each pair of papers. We see that the connection between papers a and b is now weaker because their common author was assigned a smaller weight.

5 Experimental setting and results

In this section, we describe the experiments used to evaluate the performance of the presented classifiers. We first describe the data set we used in our experiments. Then, we present the experiment set up and results.

5.1 Data set description

We evaluated the proposed weighting heuristics on a data set of customer purchases used in the PAKDD 2015 mining competition *Gender prediction based on e-commerce data*. The data consists of $30,000$ customers. The data for each customer consists of the gender (the target variable), the start and end time of the purchase, and the list of products purchased. A typical product is described by a 4-part string (for example: `A3/B5/C2/D8`). The strings describe a 4-level hierarchy of products, meaning that the example product is the product $D8$ (or D-level category) which belongs to (A-level) category $A3$, sub-category (or B-level category) $B5$ and sub-subcategory (or C-level category) $C3$. The category levels are consistent, meaning that if two products belong to the same B-level category, they also belong to the same A-level category. The data set is highly imbalanced: $23,375$ customers are women and $6,625$ are men.

For the purpose of our experiments, we ignored the temporal aspects of the data set and only focused on the products the customers purchased. This allowed us to view the data set as an implicitly defined heterogeneous network. The network consists of five node types: customers (the base node type) and four hierarchy levels. In this heterogeneous network, every purchase by a customer defines four edges in the heterogeneous network: one edge between the customer and each (sub)category to which the product belongs.

We constructed four homogeneous networks from the original heterogeneous network. In the first, two customers are connected if they purchased the same product (same D-level item), i.e. if they are connected by a path in the original network that goes through a D-level item. In the second, they are connected if they purchased a product in the same sub-subcategory (C-level item), in the third if they purchased the same B-level item and in the fourth if they purchased the same A-level item. The constructed networks are referred to as A-, B-, C- and D-level networks from this point on.

5.2 Experiment description

The first set of experiments was designed to determine if the results of [6], which show that a centroid classifier, trained on Personalized PageRank feature vectors, performs as good as the more complex SVM classifier. We tested the performance of the centroid classifier, the k-nearest neighbors classifier (with k set to $1, 2, 5$ and 10), and the SVM classifier. Because the data set is imbalanced we tested the SVM classifier both with uniform instance weights as well as weights proportional to the class frequencies. The tests were performed on feature vectors extracted from all four homogeneous networks. We randomly sampled $3,000$ network nodes to train all classifiers and tested their performance on the remaining $27,000$ nodes. The small size of the training set ensured that the training phase was fast.

In the second set of experiments we tested the heuristics, used in the construction of the homogeneous networks. We tested three classifiers. The first was the SVM classifier using solely the Personalized PageRank vectors extracted from the network. As the results of the first experiment showed that weights, proportional to the class frequencies, improve the classification accuracy of the SVM classifier, we used the same weights for this set of experiments. The second classifier we tested was the label propagation classifier as defined in [25], which classified the network nodes using the graph itself. The third classifier was the label propagation classifier with the starting matrix Y, adjusted for the class frequencies, as proposed in Section 3.2. The goal of this round of experiments was to both compare the label propagation classifier with the SVM classifier and evaluate whether the adjusted starting matrix Y has an effect on classifier performance. As in experiment 1, we trained the classifiers on a randomly sampled set of $3,000$ network nodes and tested their performance on the remaining $27,000$ nodes.

In the third round of experiments, we tested the performance of the label propagation and propositionalization based classifiers on all four homogeneous networks. Based on the results of the first two sets of experiments, we used the SVM classifier for the propositionalization approach and the label propagation method with the modified starting matrix Y. As explained in Section 3.2, we constructed feature vectors for SVM classifiers by concatenating feature vectors of individual homogeneous networks. We constructed the adjacency matrix for the label propagation algorithm by summing the four adjacency matrices. One of the goals of the third round of experiments was to test the performance of the classifiers when they are trained on a large data set. This motivated us to train the classifiers on 90% of the data set and test their performance on the remaining 10%.

In all experiments we evaluated the accuracy of the classifiers using the *balanced accuracy* metric. This is the metric used in the PAKDD'15 Data Mining Competition and is defined as

$$\frac{\frac{|\{\text{Correctly classified male customers}\}|}{|\{\text{All male customers}\}|} + \frac{|\{\text{Correctly classified female customers}\}|}{|\{\text{All female customers}\}|}}{2}. \tag{4}$$

Classifier:	Centroid	1-nn	2-nn	5-nn	10-nn	SVM	SVM (balanced weights)
A-level network	74.19%	63.61%	71.93	72.74%	74.36%	74.03%	74.62%
B-level network	70.78%	56.42%	59.17%	65.30%	67.73%	63.51%	72.61%
C-level network	64.71%	63.62%	67.21%	68.26%	71.65%	70.15%	75.18%
D-level network	60.08%	67.36%	70.39%	66.72%	66.06%	65.61%	71.17%

(a) Results of the first set of experiments.

Scheme	*A*-level	*B*-level	*C*-level	*D*-level
`tf`	76.61%	74.00%	77.34%	73.65%
`chi^2`	77.80%	74.17%	76.86%	68.76%
`idf`	77.80%	74.22%	77.23%	72.25%
`delta`	77.80%	74.14%	77.23%	72.52%
`rf`	77.80%	74.11%	76.81%	70.54%
`ig`	77.80%	74.12%	76.87%	68.72%

(b) Performance of the SVM classifier in the second round of experiments.

Scheme	*A*-level	*B*-level	*C*-level	*D*-level
`tf`	75.52%	64.28%	63.60%	72.44%
`chi^2`	76.02%	65.15%	71.95%	72.75%
`idf`	74.90%	63.83%	61.02%	72.48%
`delta`	74.90%	63.76%	61.05%	72.48%
`rf`	75.52%	64.28%	67.59%	72.55%
`ig`	76.02%	65.15%	72.41%	72.96%

(c) Performance of the label propagation classifier in the second round of experiments.

Scheme	*A*-level	*B*-level	*C*-level	*D*-level
`tf`	77.16%	74.75%	77.28%	73.91%
`chi^2`	77.16%	74.44%	77.61%	73.82%
`idf`	77.20%	74.70%	77.74%	73.76%
`delta`	77.20%	74.71%	77.74%	73.76%
`rf`	77.16%	74.59%	77.21%	74.03%
`ig`	77.16%	74.49%	77.59%	73.79%

(d) Performance of the balanced label propagation classifier in the second round of experiments.

Scheme	SVM	Label propagation
`tf`	81.35%	77.06%
`chi^2`	81.78%	77.10%
`idf`	82.09%	79.03%
`delta`	81.94%	79.08%
`rf`	81.49%	77.16%
`ig`	81.56%	77.12%

(e) The results of the third set of experiments showing the balanced accuracies of the SVM and label propagation classifiers on the entire data set.

5.3 Experimental results

The first set of experiments, shown in Table 3a, shows that there is a large difference in the performance of different classifiers. Similarly to Grčar et al. [6], the simple centroid classifier performs well on feature vectors extracted from several different homogeneous networks. However, the classifier is still consistently outperformed by the SVM classifier if the instance weights of the classifiers are set according to the class sizes. We conclude that the optimal classifier for the methodology, introduced in [6], depends on the data set.

The results of the second set of experiments are shown in Tables 3b, 3c and 3d. When comparing the results of the two label propagation approaches the results show that label propagation with adjusted starting matrix has large impact on the performance of the classifier, as the balanced accuracy increases by 1–2% in the case of the *A*- and *D*-level network and even more in the case of *B*- and *C*-level networks. This result confirms the intuition that, in Section 3.2, motivated the construction of the adjusted starting matrix.

Different heuristics used in construction of homogeneous networks also affect the final performance of all three classifiers. No heuristic consistently outperform the others, meaning that the choice of heuristic is application dependent. The last conclusion of the second round of experiments is that the computationally demanding propositionalization method does not outperform the label propagation method. In all four networks choosing the correct heuristic and correct weights for the starting matrix allows the label propagation method to perform comparably to the SVM classifier.

Table 3e shows the results of the third set of experiments. In this experiment, the propositionalization-based approach clearly outperforms the label propagation algorithm. It is possible that this effect occurs because the network propositionalization approach, in particular the SVM classifier, require more training examples (compared to the network propagation classifier) to perform well. A second explanation may come from the way the four networks were combined in our experiments (i.e. the concatenation approach in the case of network propositionalization and the matrix sum in the case of label propagation). By summing the adjacency matrices before performing label propagation, we implicitly assumed that the connections between customers that purchased the same D-level product, are equally important as connections between customers that purchased the same A-level product. This may cause the amount of A-level edges to everwhelm the effects of the D level edges, causing the resulting network to be very similar to the original A level network. The propositionalization based approach is less prone to an error of this type, as the idea of the SVM algorithm is to define correct weights for elements of the feature vectors. The SVM algorithm is therefore flexible enough to assign larger weights to the features, produced by the D-level network, if it estimates that these features are more important in classification.

The second conclusion we can draw from the third set of experiments is that the effect of using weighting heuristics in the construction of the homogeneous networks is still obvious. With both classification methods the adjusted `delta` and `idf` heuristics perform best.

6 Conclusions and further work

While network analysis is a well established research field, analysis of heterogeneous networks is much newer and less researched. Methods taking the heterogeneous nature of the networks into account show an improved performance [2]. Some methods like RankClus and others presented in [20] are capable of solving tasks that cannot be defined with homogeneous information networks (like clustering two disjoint sets of entities). Another important novelty is combining network analysis with the analysis of node data, either in the form of text documents or results obtained from various experiments [5, 8, 6].

The contributions of the paper are as follows. By setting the weights of the initial class matrix proportionally to the class value frequency, we improved the performance of the label propagation algorithm when applied to a highly imbalanced data set. We adapted heuristics, developed primarily for use in text mining, for the construction of homogeneous networks from heterogeneous networks. Our results show that the choice of heuristics impacts the performance of both label propagation classifier and classifiers based on the propositionalization approach of [6]. We also present a variation of the label propagation approach, described in [25].

In future work, in-depth analysis of the network construction heuristics and their performance in classifiers applied to homogeneous networks will be pursued. We plan to design efficient methods for propositionalization of large data sets and decrease the computational load of PageRank calculations by first detecting communities in a network. Such "pre-processing" should reduce the size of a network on which PageRank calculations are to be performed.

We plan to test the methods, described in this paper, on publicly available data sets such as the DBLP, Cora and CiteSeer databases. The presented heuristics shall be evaluated on the methodology for mining text enriched heterogeneous networks presented in [6]. For that, one has to construct a heterogeneous network in which the central node represents genes, connected to the response of plants against an infection. We will enrich the nodes with papers from the PubMed database which mention the genes.

Acknowledgement

The presented work was partially supported by the European Commission through the Human Brain Project (Grant number 604102). We also acknowledge the support of research projects funded by the National Research Agency: the Knowledge Technologies research programme and the project Development and applications of new semantic data mining methods in life sciences.

References

[1] Burt, R. and Minor, M. (1983). *Applied Network Analysis: a Methodological Introduction*. Sage Publications.

[2] Davis, D., Lichtenwalter, R., and Chawla, N. V. (2011). Multi-relational link prediction in heterogeneous information networks. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 281–288.

[3] Debole, F. and Sebastiani, F. (2004). Supervised term weighting for automated text categorization. In *Text mining and its applications*, pages 81–97. Springer.

[4] D'Orazio, V., Landis, S. T., Palmer, G., and Schrodt, P. (2014). Separating the wheat from the chaff: Applications of automated document classification using support vector machines. *Polytical Analysis*, 22(2):224–242.

[5] Dutkowski, J. and Ideker, T. (2011). Protein networks as logic functions in development and cancer. *PLoS Computational Biology*, 7(9).

[6] Grčar, M., Trdin, N., and Lavrač, N. (2013). A methodology for mining document-enriched heterogeneous information networks. *The Computer Journal*, 56(3):321–335.

[7] Han, E.-H. and Karypis, G. (2000). Centroid-based document classification: Analysis and experimental results. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 424–431. Springer.

[8] Hofree, M., Shen, J. P., Carter, H., Gross, A., and Ideker, T. (2013). Network-based stratification of tumor mutations. *Nature Methods*, 10(11):1108–1115.

[9] Jeh, G. and Widom, J. (2002). SimRank: A measure of structural-context similarity. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543. ACM.

[10] Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21.

[11] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.

[12] Kondor, R. I. and Lafferty, J. D. (2002). Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322.

[13] Kralj, J. (2015). Mining heterogeneous citation networks. In *Proceedings of the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 672–683.

[14] Kwok, J. T.-Y. (1998). Automated text categorization using support vector machine. In *Proceedings of the 5th International Conference on Neural Information Processing*, pages 347–351.

[15] Lan, M., Tan, C. L., Su, J., and Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(4):721–735.

[16] Manevitz, L. M. and Yousef, M. (2002). One-class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154.

[17] Martineau, J. and Finin, T. (2009). Delta TFIDF: An Improved Feature Space for Sentiment Analysis. In *Proceedings of the Third AAAI Internatonal Conference on Weblogs and Social Media*, San Jose, CA. AAAI Press.

[18] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

[19] Storn, R. and Price, K. (1997). Differential evolution; a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359.

[20] Sun, Y. and Han, J. (2012). *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers.

[21] Sun, Y., Yu, Y., and Han, J. (2009). Ranking-based clustering of heterogeneous information networks with star network schema. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 797–806.

[22] Tan, S. (2006). An effective refinement strategy for KNN text classifier. *Expert Syst. Appl.*, 30(2):290–298.

[23] Vanunu, O., Magger, O., Ruppin, E., Shlomi, T., and Sharan, R. (2010). Associating genes and protein complexes with disease via network propagation. *PLoS Computational Biology*, 6(1).

[24] Zachary, W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473.

[25] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16(16):321–328.