Analysis of text-enriched heterogeneous information networks

Jan Kralj[1,2], Anita Valmarska[1,2], Miha Grčar, 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

Abstract. This chapter addresses the analysis of information networks, focusing on heterogeneous information networks with more than one type of nodes and arcs. After an overview of tasks and approaches to mining heterogeneous information networks, the presentation focuses on text-enriched heterogeneous information networks whose distinguishing property is that certain nodes are enriched with text information. A particular approach to mining text-enriched heterogeneous information networks is presented that combines text mining and network mining approaches. The approach decomposes a heterogeneous network into separate homogeneous networks, followed by concatenating the structural context vectors calculated from separate homogeneous networks with the bag-of-words vectors obtained from textual information contained in certain network nodes. The approach is show-cased on the analysis of two real-life text-enriched heterogeneous citation networks.

1 Introduction

The field of *network analysis* has its roots in two research fields: mathematical graph theory and social sciences. Network analysis started as an independent research discipline in the late seventies (Zachary, 1977) and early eighties (Burt and Minor, 1983), when sociologists became increasingly aware that the study of social relations—and not only individual attributes—is necessary for in-depth analysis of human societies. Since this early research, network analysis has grown substantially: the field now covers not only social networks but also general networks originating from any (scientific) discipline.

In recent years, analysis of *heterogeneous information networks* (Sun and Han, 2012) has gained momentum. In contrast to standard *homogeneous* information networks, heterogeneous information networks describe heterogeneous types of entities and different types of relations. Moreover, in *enriched heterogeneous information networks*, nodes of certain type contain additional information, for example in the form of experimental results or documents. After an

overview of tasks and approaches to mining heterogeneous information networks, we focus on *text-enriched heterogeneous information networks*. We present a particular approach to mining text-enriched heterogeneous information networks, together with its application in two complex real-life domains. In the first example, video lectures from the VideoLectures.NET website, forming a network of lectures, authors and viewers, are enriched with their abstracts. The results show that using both structural context vectors and bag-of-words vectors improves category prediction compared to using only one type of vectors. In the second example, scientific publications forming a network of publications and authors, are enriched with their abstracts. The results show that increasing the network size and combining text and network structure information improves the accuracy of paper categorization.

The chapter is structured as follows. Section 2 introduces the concepts of homogeneous and heterogeneous information networks and presents examples of such networks. Section 3 presents data analysis tasks applicable in homogeneous or heterogeneous networks. Section 4 presents an approach to the analysis of text-enriched information networks. Sections 5 and 6 present the applications of the described methodology in two real-life domains: a network of video lectures and their authors and a citation network of psychology papers, respectively. The chapter concludes with a summary and opportunities for further work.

2 Information networks

This section introduces the area of *information network analysis*, illustrated with some real-world examples of information networks.

Standard data sets used in data mining and machine learning are usually available in a tabular form, where a data instance (corresponding to a row in the data table) is characterized by its properties described in terms of the values of a selected set of attributes (each corresponding to a table column). In contrast, the motivation for information network mining is due to the fact that information may exists both at the instance level and in the way how the instances interact.

Intuitively, an information network is a network composed of entities (for example, web pages) that are in some way connected to other entities (one page may contain links to other pages). In mathematical terms, such structures are represented by graphs.

Definition 1. *A* graph $G = (V, E)$ *is a mathematical object, composed of a set of vertices* V *and a set of edges* E *connecting the vertices. Set of edges* E *is the union of two sets,* $E = E_U \cup E_D$*, where set* E_U *contains undirected edges* $\{x, y\}$ *and set* E_D *contains directed edges* (x, y) *between pairs of vertices* x, y*.*

- *If all edges present in* E *are undirected, we call the graph* undirected*. If all edges are directed, the graph is* directed*. Graphs containing both directed and undirected edges are sometimes referred to as* mixed *graphs.*
- *A graph with no loops (edges connecting a node to itself) and no multiple edges (meaning that a pair of nodes is connected by at most one edge) is called a* simple *graph.*

Graphs are a convenient way to represent relations between different entities, but do not contain any real data themselves. An *information network* is a graph in which each vertex has certain properties. Networks are a richer way of representing data than using either graphs or tables, but can lack the power to represent truly complex interactions between entities of different types. To this end, we define the concept of a heterogeneous information network.

Definition 2. *A* heterogeneous information network *is a tuple* $(V, E, \mathcal{A}, \mathcal{E}, \tau, \phi)$*, where* $G = (V, E)$ *is a directed graph,* $\mathcal{A}$ *a set of object types,* $\mathcal{R}$ *a set of edge types and* $\tau : V \to \mathcal{A}$ *and* $\phi : E \to \mathcal{R}$ *are functions satisfying the conditions: if edges* $e_1 = (x_1, y_1)$ *and* $e_2 = (x_2, y_2)$ *belong to the same edge type* $(\phi(e_1) = \phi(e_2))$*, then their start points and their end points belong to the same vertex type* $(\tau(x_1) = \tau(x_2)$ *and* $\tau(y_1) = \tau(y_2))$*.*

Remark 1. In many information networks, vertices of two types a_1 and a_2 are only connected by edges of one type. In this case, the edge type is uniquely defined by the type of its starting and ending vertex type and is not explicitly stated. It is common to view the elements of the set $\mathcal{A}$ as disjoint sets of vertices from V, instead of abstract types. This gives rise to the style of writing, where for type $t_1 \in \mathcal{A}$, and vertex $v \in V$, we write $v \in t_1$ instead of the usual $\tau(v) = t_1$ to denote the fact that v is of type t_1.

Remark 2. A heterogeneous information network may also be represented in a relational, RDF-like form as a set of triplets. Edge types, in this representation, would be represented as relations. In this representation, network schemas of heterogeneous information networks correspond to RDF schemas. The constraints put on edge types in heterogeneous information networks (i.e. that all edges of a certain type start in nodes of the same type) can be encoded using `rdfs:domain` and `rdfs:range` properties.

Sun and Han (2012) note that sets $\mathcal{A}$ and $\mathcal{E}$, along with the restrictions imposed by the definition of a heterogeneous information network, can be seen as a network as well, with edges connecting two vertex types if there exists an edge type whose edges connect vertices of the two vertex types. The authors call this 'meta-level' description of a network a *network schema*.

Definition 3. *For a heterogeneous information network* $G = (V, E, \mathcal{A}, \mathcal{E}, \tau, \phi)$*, a* network schema of G*, denoted* T_G*, is a directed graph with vertices* $\mathcal{A}$ *and edges* $\overline{\mathcal{E}}$*, where edge type* $t \in \mathcal{E}$ *whose edges connect vertices of type* $t_1 \in \mathcal{A}$ *to vertices of type* $t_2 \in \mathcal{A}$*, defines an edge in* $\overline{\mathcal{E}}$ *from type* t_1 *to* t_2*.*

Given such a broad definition of heterogeneous information networks, a large amount of human knowledge can be expressed in the form of networks. Some examples are listed below.

Example 1. Bibliographic information networks or *citation networks*, such as the DBLP network examined by Sun and Han (2012) or the network examined by Grčar et al. (2013) are networks connecting authors of scientific papers with

their papers. Thus, in their elementary form, they contain at least two types of entities (authors (A) and papers (P)), and at least one type of edges, connecting authors to the papers they have (co)authored. On top of this, the network may also include several other entity types, including journals and conferences (which can be merged into one type, *venue*), institutions and so on. Along with the entity types, the list of edge types is also expanded: papers are, for example, written by authors, published at venues and contain certain terms. Papers may cite other papers, meaning that a paper in the network can be connected to entities of all other entity types in the network.

Example 2. Online social networks model the structure of popular online social platforms such as Twitter and Facebook. In the case of Twitter the network entity types are *user, tweet, hashtag*, and *term*. The connections between the types are as follows: users *follow* other users and *post* tweets, tweets *reply* to other tweets and *contain* both terms and hashtags.

Example 3. Biological networks represent a starting point for a large number of different heterogeneous information networks and can contain entity types such as species, genes, Gene Ontology (Consortium, 2000) annotations, proteins, metabolites and so on. The types of links between such mixed entities are diverse. For example, genes can belong to species, encode proteins, be annotated by a GO annotation, and so on.

3 Analysis of information networks

We present some analytic tasks which can be applied to information networks. First, we present general tasks that can be applied to homogeneous networks, followed by approaches to mining heterogeneous networks.

3.1 Tasks in homogeneous information network analysis

The field of information network analysis covers a wide variety of tasks. Some of them are listed below.

Classification. Classification of network data is a natural generalization of classification tasks encountered in a typical machine learning setting. The problem formulation is simple: given a network and class labels for some of the entities in the network, predict the class labels for the rest of the entities in the network. Another name for this problem is *label propagation*. A common approach used for this task is the algorithm proposed by Zhou et al. (2004). The approach finds a probability distribution f of a vertex v_i being labeled with label 1 (as opposed to 0). The classification problem in this case is a binary classification problem. The method finds f by minimizing the function

$$f^T(I - D^{-1/2}MD^{-1/2})f + \mu||f - y||^2, \tag{1}$$

where M is the adjacency (or weight) matrix of the network and D is a diagonal matrix defined as $d_{ii} = \deg(v_i) = \sum_j m_{ij}$. The two summands in Equation (1) represent two demands that have to be fulfilled: (i) the label distribution for vertices which are connected (especially the ones connected with strong edges) must be similar, and (ii) the distribution must be close to the original distribution of the data already labeled. Parameter μ determines the strength of two influences on the result: a large value of μ results in a labeling that closely matches the known labels while a small value of μ strongly penalizes connections between differently labeled vertices. This approach was tried by Vanunu et al. (2010), where the method was used to discover new genes, associated with a disease.

Link prediction. While classification tasks try to discover new knowledge about network entities, link prediction focuses on unknown connections between the entities. The assumption is that not all network edges are known. The task of link prediction is to predict new edges that are missing or likely to appear in the future. A common approach to link prediction is assigning a score $s(u, v)$ to each pair of vertices u and v which models a probability of the vertices being connected. Approaches used include calculating the score as a product of vertex degrees (Barabási et al., 2002) and (Newman, 2001a), or using the number of common neighbors of two vertices, $|N_u \cap N_v|$ (Newman, 2001b). The latter approach can be modified to the Jaccard coefficient of N_u and N_v, which is defined as $\frac{|N_u \cap N_v|}{|N_u \cup N_v|}$ with values in $[0, 1]$. This normalization prevents high degree vertices to overshadow low degree vertex pairs, which may have a large share of common neighbors. The Adamic/Adar measure, used in (Adamic and Adar, 2003), further increases the impact of low degree vertices by calculating the distance as follows: $\sum_{n \in N_u \cap N_v} \frac{1}{\log(|N_n|)}$.

Community detection. There is a general consensus on what a network community is, however, there is no strict definition of the term. The idea is well summarized in the definition by Yang et al. (2010): a community is a group of network nodes, with dense links within the group and sparse links between the groups. An extensive overview of community detection methods is presented in (Plantié and Crampes, 2013).

Ranking. The objective of ranking in information networks is to asses the relevance of a given object either globally (with regard to the whole graph) or locally (relative to some object in the graph). A well known ranking method is PageRank (Page et al., 1999), which was used in the Google search engine. The idea of PageRank—frequently abbreviated as PR—is simple: for a given network with the adjacency matrix M, the score of the i-th vertex is equal to the i-th component of the dominant eigenvector of M'^T, where M' is the matrix M with rows normalized so that they sum to 1. This is motivated by two different views.

The first is the random walker approach: a random walker starts walking from a random vertex v of the network and in each step walks to one of the neighboring vertices with a probability proportional to the weight of the edge traversed. The PageRank of a vertex is then the expected proportion of time the walker spends in the vertex, or, equivalently, the probability that the walker is in the particular vertex after a long time. The second view of PageRank is the view of score propagation. The PageRank of a vertex is its score, which it passes to the neighboring vertices. A vertex v_i with a score $PR(i)$ transfers its score to all its neighbors. Each neighbor receives a share of the score proportional to the strength of the edge between it and v_i. This view explains the PageRank with a principle that in order for a vertex to be highly ranked, it must be pointed to by many highly ranked vertices.

Other methods for ranking include Personalized-PageRank (Page et al., 1999)—frequently abbreviated as P-PR—that calculates the vertex score locally to a given network vertex, SimRank (Jeh and Widom, 2002), diffusion kernels (Kondor and Lafferty, 2002), hubs and authorities (Kleinberg, 1999) and spreading activation (Crestani, 1997).

3.2 Tasks in heterogeneous information network analysis

Most data mining tasks in homogeneous information networks can be applied to heterogeneous networks by simply ignoring the heterogeneous structure. This, however, decreases the amount of information available in subsequent steps and can therefore decrease the performance of algorithms (Davis et al., 2011). Approaches that take the heterogeneous network structure into account are therefore preferable.

Authority ranking. Sun and Han (2012) introduce *authority ranking* to rank the vertices of a bipartite network, where vertices are comprised of a set of authors $X = \{x_1, \ldots, x_m\}$ and a set of papers $Y = \{y_1, \ldots, y_n\}$. There are two edge types: links from papers to authors and links from authors to papers. The adjacency matrix of the network can therefore be written as

$$M = \begin{bmatrix} 0 & M_{XY} \\ M_{YX} & 0 \end{bmatrix}$$

where M_{YX} contains weights of edges pointing from authors to papers and M_{XY} contains weights of edges pointing from papers to authors.

The concept of authority ranking is a generalization of PageRank for bipartite networks, defining two functions r_X (ranking the set X) and r_Y (ranking the set Y) to rank papers and authors separately. The functions are defined as follows:

$$r_X(x_i) = \sum_{j=1}^{n} r_{ij}^{XY} r_Y(x_j) \tag{2}$$

$$r_Y(y_j) = \sum_{i=1}^{m} r_{ji}^{YX}(j) r_X(y_i) \tag{3}$$

where r_{ij}^{XY} is the weight of the edge between vertices i and j. The weights the matrix R, obtained from the matrix M by normalizing the row sums to 1, as in the PageRank approach. The above equations can be rewritten as an eigenproblem for a block matrix, since vectors r_X and r_Y satisfy $r_X = R_{XY} r_Y$ and $r_Y = R_{YX} r_X$ or, in matrix form:

$$\begin{bmatrix} r_X \\ r_Y \end{bmatrix} = \begin{bmatrix} 0 & R_{XY} \\ R_{YX} & 0 \end{bmatrix} \begin{bmatrix} r_X \\ r_y \end{bmatrix}.$$

Similarly, Sun et al. (2009b) define authority ranking on a star heterogeneous network with a central type Z, where instead of propagating authority directly from a node of type X to a node of type Y, authority is propagated indirectly through a node of type Z, yielding equations $r_X = R_{XY} R_{ZY} r_Y$ for all pairs of types X and Y.

Ranking based clustering. While both ranking and clustering can be performed on heterogeneous information networks, applying only one of the two may sometimes lead to results which are not truly informative as there is a high risk of apples-to-pears comparisons being made. For example, simply ranking authors in a bibliographic network may lead to a comparison of scientists in completely different fields of work which may not be comparable. Sun and Han (2012) propose joining the two seemingly orthogonal approaches to information network analysis (ranking and clustering) into one. They propose two algorithms: RankClus (Sun et al., 2009a) and NetClus (Sun et al., 2009b), both of which cluster entities of a certain type (for example, authors) into clusters and rank the entities within clusters. Algorithm RankClus is tailored for bipartite information networks, while NetClus can be applied to networks with a star network schema.

The RankClus algorithm starts with a starting clustering of elements, which it then iteratively improves. The ranking of objects within each type is used to define ranking functions $r_{Y|X_i}$, which rank elements of type Y only taking into account elements of type X, belonging to cluster X_i. For the next step, the algorithm considers the ranking $r_{Y|X_i}$ as values proportional to probabilities that objects from Y belong to cluster X_i. This is justified by the fact that when the clustering is discovered, the elements of Y will only have a high rank within the cluster they belong to. Using this view the algorithm constructs a mixture model (using the EM algorithm (Bilmes, 1997)) to evaluate the probabilities of *links* belonging to each of the clusters. Using this knowledge, new clusters of type X are constructed and the process is repeated until convergence. The NetClus algorithm shares its idea with the RankClus. Instead of applying probabilities to links, as in RankClus, the role of links in NetClus is replaced by objects belonging to the central type in the star network.

Classification through label propagation. The problem of classification is generalized from homogeneous to heterogeneous networks: given a network

and class labels for some of the entities in the network, predict the labels of the remaining entities in the network. In (Hwang and Kuang, 2010) the idea of label propagation used by Zhou et al. (2004) is expanded to include multiple parameters μ_{ij} in place of a single parameter μ appearing in Equation (1). A similar approach is taken by Sun and Han (2012). Ji et al. (2010) propose the GNETMINE algorithm which uses the idea of knowledge propagation through a heterogeneous information network to find probability estimates for labels of the unlabeled data. A strong point of this approach is that it has no limitations on the network schema, meaning it can be applied to both highly complex heterogeneous and homogeneous networks.

Ranking based classification. Building on the idea of GNETMINE, Sun and Han (2012) propose a classification algorithm that relies on within-class ranking functions to achieve better classification results. The idea is that nodes, connected to *high ranked* entities belonging to class c, most likely belong to the same class. This idea is implemented in the RankClass framework for classification in heterogeneous information networks.

Ranking and classification in RankClass are interlinked, since only elements *within* each class are ranked rather than the whole set. The methodology consists of two steps which are applied successively until the convergence. In the ranking step, the network elements are ranked according to the authority ranking principle. Then, given the rankings of elements, the EM algorithm calculates new estimates of probabilities that elements belongs to a certain class. Edges connecting elements likely to belong to the same class are increased and within class rankings are recalculated.

Multi-relational link prediction. Expanding the ideas of link prediction for homogeneous information networks, Davis et al. (2011) propose a link prediction algorithm for each pair of object types in the network. The score is higher if the two objects are likely to be linked. Two objects o_1 and o_2 of types t_1 and t_2 have a high score if there exist many common neighbors of o_1 and o_2, which are neighbors to connected objects of types t_1 and t_2 (for example, if two authors often attend the same conferences, and it is common for authors at a conference to be paper co-authors, it is probable that the two authors are going to become co-authors of a paper).

Semantic link association prediction. Chen et al. (2012) constructed a heterogeneous network consisting of $295,897$ nodes and $727,997$ edges from 17 publicly available data sources about drug target interaction, including semantically annotated knowledge sources in the form of ontologies. The constructed heterogeneous network contains 10 node types and 12 edge types. Two most important node types are target nodes, representing individual genes, and chemical compound nodes. These two node types are connected by two edge types: a chemical compound can bind to a certain target gene or can change the expression of the gene. In addition to these two link types, target nodes are linked to

nodes representing Gene Ontology (Consortium, 2000) concepts, KEGG (Kanehisa and Goto, 2000) pathways, tissues and diseases. Chemical compound nodes are linked to nodes representing chemical ontology concepts, chemical substructures, medical side effects and diseases.The authors developed a statistic model called Semantic Link Association Prediction (SLAP) to measure associations between network elements. Scores are calculated for drug-target pairs for each possible meta path between the two. The scores are normalized for each meta path, with the sum giving an actual association score between the elements. Element pairs with significant scores (smaller p-values) are then discovered.

3.3 Data-enriched network analysis

The methods described in Sections 3.1 and 3.2 rely solely on the network structure to extract information. However, as an information network includes both the network structure and the data itself, it is sensible to include the data attached to each node into the network analysis process as well.

Network Guided Forest. Dutkowski and Ideker (2011) present a method based on decision trees to analyze a protein-protein interaction network. They analyze gene expression data from several studies on human cancers. The data consists of gene expression levels, obtained through microarray experiments and contains a series of expression levels, one for each gene of each sample. The proposed method, named Network-Guided Forests (NGF), constructs a forest of trees to classify an example into the appropriate class according to the expression levels of the examined genes. The final result is obtained through the aggregation of all results. The NGF method is similar to the random forest method (Breiman, 2001) as it constructs several decision trees and each decision tree classifies examples according to their gene expression. The difference is that in NGF the construction of trees is guided by the underlying network of protein-protein interactions, which helps to find the best gene to split the data in each tree node. This approach is interesting from a conceptual point of view, as it is composed of both network analysis methods and standard statistical and data mining algorithms. It can be viewed as either mining data enriched with a network component, or analysis of networks enriched with experimental data.

Two-step clustering. Hofree et al. (2013) also combine network analysis and data mining. They analyze a complex network of gene-gene interactions to analyze cancer patient data. The data consists of a large binary matrix F with values indicating if a given gene is mutated for a patient. They propose a two-step patient clustering method. First, a network propagation approach based on (Zhou et al., 2004) (see Section 3.1) is applied to the network which transforms the original binary matrix into a matrix with values in $[0, 1]$. In the second step, authors use non-negative matrix factorization (Lee and Seung, 1999) to find candidate features to use in clustering.

Network extraction using text mining. While human readable documents may contain a lot of information, this information is not conveniently structured for data analysis. As information networks are a better way of representing knowledge, several methods and applications converted databases of scientific articles into large (and usually heterogeneous) information networks. One of the first attempts is described by Jenssen et al. (2001), where a network of human genes is constructed from titles and abstracts of over 10 million MEDLINE records. Kok and Domingos (2008) use relational clustering to cluster both vertices and edges and construct a semantic network from the text. In (Chen and Sharp, 2004), a NLP-based text mining approach called Chilibot was introduced. The methodology can construct networks about biological entities using articles collected from PubMed. Van Landeghem et al. (2013) used a semantic network extracted from PubMed articles with protein–protein and regulatory interactions from experimental databases to discover clusters of tightly connected genes.

4 Mining text-enriched heterogeneous information networks

This section introduces the methodology of mining text-enriched information networks first described by Grčar et al. (2013). The methodology uses both text mining and network analysis of text-enriched heterogeneous information networks (such as the citation network of scientific papers) to construct feature vectors which describe both, the location of nodes in the network and internal structure of nodes.

4.1 Data structure

The data in a text-enriched heterogeneous information network is a fusion of two diffedrent data types: heterogeneous information networks and texts. Our data thus comprises of a heterogeneous information network with different node and edge types, where nodes of one designated type are text documents. An example of a heterogeneous citation network in which the text documents are papers is shown in Figure 1 and its network schema is presented in Figure 2.

Remark 3. In a directed heterogeneous network, an edge from vertex v to vertex w (for example, an author *writes* a paper) implicitly defines an 'inverse' edge going from vertex w to vertex v (a paper is *written by* an author).

4.2 Network decomposition

The first step of the methodology focuses on the network structure. The original heterogeneous information network is decomposed into a set of homogeneous networks. Each homogeneous network is constructed from a circular walk in the original network schema. If a sequence of node types $t_1, t_2, \ldots, t_n$ forms a circular

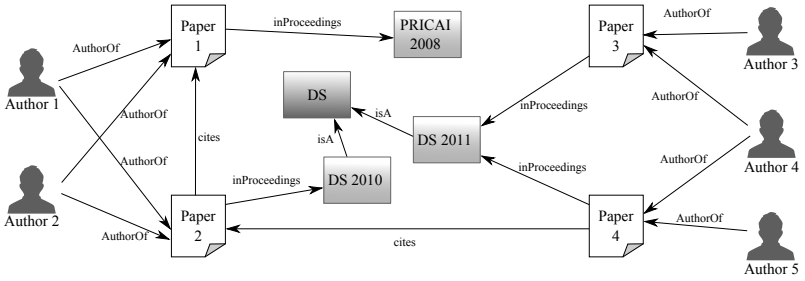


Fig. 1. An example of a citation network (from Grčar et al. (2013)).

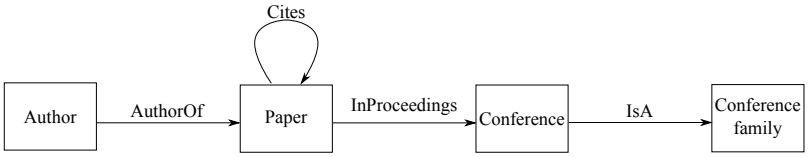


Fig. 2. The network schema of the citation network, shown in Figure 1.

walk (meaning that $t_1 = t_n$) in the network schema, then two nodes n and m are connected in the decomposed network if there exists a walk $n_1, n_2, \ldots n_n$ such that $n_1 = n$, $n_n = m$ and each node n_i in the walk is of type t_i.

Take for example the network shown in Figure 1. From it (using the implicitly defined inverse edges, described in Remark 3), we construct three homogeneous networks of papers, shown in Figure 3:

- The first network (Figure 3a) is constructed using the walk Paper *HasAuthor* Author *AuthorOf* Paper, i.e., two papers are connected if they share a common author.
- The second network (Figure 3b) is constructed using the walk Paper *inProceedings_of* Conference *isA* Conference family *hasConference* Conference *containsPaper* Paper and two papers are connected if they appeared at the same conference family.
- The third network (Figure 3c) is constructed using the walk Paper *cites* Paper and two papers are connected if one paper cites another.

This step of the methodology is the only one which cannot be made fully automatic. For each heterogeneous network, different meta-paths can be considered, and expert judgment is required to asses their importance. Usually meta-paths of heterogeneous networks have a real-world meaning, so field experts may provide insights into paths importance.

4.3 Feature vector construction

In the second step of the methodology, a set of feature vectors is calculated for each node in the original heterogeneous network: a bag-of-words vector con-

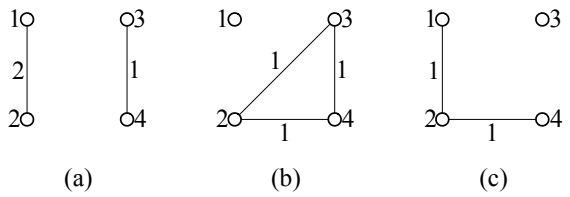


Fig. 3. The decomposition of the network from Figure 1 according to the (a) paper-author-paper, (b) paper-conference family-paper and (c) paper-paper meta-paths. The nodes in the decompositions correspond to the papers in the original network. The weights, assigned to the edges in this example, are obtained by simply counting the number of paths in the original network which correspond to a link in the decomposed network. The weight in the Paper-Author-Paper decomposition corresponds to the number of authors, shared by two papers.

structed from the text document, and feature vectors constructed from all homogeneous networks.

In the bag-of-words (BOW) construction, each text is processed using standard natural language processing techniques. Typically the following steps are performed: preprocessing using a tokenizer, stop-word removal, stemming, construction of N-grams of certain length, and removal of infrequent words from the vocabulary. The resulting vectors are normalized according to the Euclidean metric.

For each homogeneous networks, obtained through network decomposition, the personalized PageRank (P-PR) algorithm (Page et al., 1999) is used to construct feature vectors for each node in the network.

Personalized page rank of node v (P-PR$_v$) in a network is a vector, which for each other node w of the network, tells how simple it is to randomly walk from v to w. It is defined as the stationary distribution of the position of a random walker which starts its walk in node v and at either selects one of the outgoing connections or travels to his starting location. The probability (denoted p) of continuing the walk is a parameter of the algorithm and is usually set to 0.85. The resulting PageRank vectors are normalized according to the Euclidean norm to make them compatible with the BOW vector calculated for the same document.

The PageRank vector is calculated iteratively. In the first step, the rank of node v is set to 1 and the other ranks are set to 0 to construct r^0, the 0-th estimation of the PageRank vector. Then, at each step, the rank is spread along the connections of the network using the formula

$$r^{(k+1)} = p(A^T r^{(k)}) + (1-p)r^{(0)}. \tag{4}$$

In Equation 4, $r^{(k)}$ is the estimation of the PageRank vector after k iterations, and A is the coincidence matrix of the network, normalized so that the elements in each row sum to 1. If all elements in a given row of the coincidence matrix are zero (i.e., if a vertex has no outgoing connections), all values in that row are

set to $\frac{1}{n}$, where n is the number of vertices (this simulates the behaviour of the walker when jumping from a node with no outgoing connections to any other node in the network).

Remark 4. Continuing from Remark 2, if heterogeneous information networks are viewed as RDF-graphs, we can consider the feature vector construction as a further enrichment of the RDF-graph. Bag-of-words vectors can be represented as a set of triplets using the hasTerm relation, as seen in Lytras and Sheth (2010), and $P-PR$ vectors may be represented as a set of triplets using a distanceFrom relation with a numeric property. In this way, the PageRank vector for node v would be encoded in all relations of the type distanceFrom which start in v.

4.4 Data fusion

The result of running both the text mining procedure and the P-PR algorithm is a set of vectors $\{v_0, v_1, \ldots, v_n\}$ for each node v, where v_0 is the BOW vector, and where for each i ($1 \leq i \leq n$, where n is the number of network decompositions), v_i is the personalized PageRank vector of node v in the i-th homogeneous network. In the final step of the methodology, these vectors are combined to create a final feature vector. Using positive weights $\alpha_0, \alpha_1, \ldots, \alpha_n$, which sum to 1, a unified vector is constructed describing the node v. The vector is constructed as

$$v = \sqrt{\alpha_0} b \oplus \sqrt{\alpha_1} v_1 \oplus \cdots \oplus \sqrt{\alpha_n} v_n,$$

where the symbol $\oplus$ represents the concatenation of two vectors. The values of weights α_i can be determined automatically.

A simple way to automatically set weights is to use an optimization algorithm such as the multiple kernel learning (MKL), presented in (Rakotomamonjy et al., 2008), in which the feature vectors are viewed as linear kernels. For each i, the vector v_i corresponds to a linear mapping $\overline{v_i} : x \mapsto x \cdot v_i$. The concatenated vector v then represents the linear mapping

$$[x_0, x_1, \ldots, x_n] \mapsto \alpha_0 x_0 \cdot v_0 + \alpha_1 x_1 \cdot v_1 + \cdots + \alpha_n \cdot v_n.$$

Another possibility is to determine the optimal weights using a general purpose optimization algorithm, e.g., differential evolution (Storn and Price, 1997).

4.5 Scalability issues

While the calculation of bag-of-words vectors can be done in a single pass over the data, the calculation of $P-PR$ vectors has to be adapted when the number of basic nodes becomes too large. The iterative process converges to a stationary distribution of the rank after several steps. In our experiments, the required number of steps ranged from 50 to 100, and since each step requires a matrix-vector multiplication, the calculation of a single P-PR vector may take several seconds for a large network, making the calculation of tens of thousands of P-PR vectors computationally difficult. Here, we present some ideas to handle the rising computational complexity of large networks.

To reduce the size of the network on which PageRank vectors are calculated we calculated $P-PR_v$ by performing the PageRank algorithm on a subnetwork of the original network, composed of nodes that have a path leading from v to them in the original network (Kralj et al., 2015). The $P-PR$ value for all other nodes is set to 0. We can also limit the size of the graph on which the $P-PR$ method is applied by calculating only PageRank values of nodes in a local neighborhood of a given node, setting PageRank values for nodes that are too far from the start node to 0. This in some cases decreases the computation time, but the decrease will not occur in many real world networks, especially *small world networks* (Watts and Strogatz, 1998), in which the shortest path between any two nodes may be very short.

Alternatively, a community detection method (Plantié and Crampes, 2013) can be used as a preprocessing step in the calculation of $P-PR$ vectors. Once the communities in the graph are discovered, one can calculate $P-PR_v$ by only calculating its values on a subgraph containing all the nodes of the same community as v and links between them. We can treat the remaining communities as non-existent by setting the PageRank value of their nodes to 0, or treat them as a single entity by replacing the entire community with one node v. For a node w, the weight of the edge between v and w can be calculated as the sum, average, or maximum of all weights leading from v to the community.

5 VideoLectures.NET categorization case study

The network propositionalization approach, described in Section 4, was applied to a network of $3,520$ lectures from the VideoLectures.NET website. The aim of the experiment was to develop a method that can assist in categorization of lectures, hosted on the site. This functionality was required due to the rapid growth of the number of hosted lectures (150–200 lectures are added each month) as well as due to the fact that the categorization taxonomy is fine-grained, making manual categorization difficult.

5.1 Data set

Of the $3,520$ lectures $1,156$ lectures were manually categorized into 129 categories (one lecture may belong to more than one category) by the curators of the website. The data included $2,706$ lecture authors, events at which the lectures were filmed and $62,070$ user clicks. From this data we constructed a heterogeneous network containing lectures, authors, events and portal users as nodes.

Each lecture contained a title and possibly an abstract which were used to create the BOW vector for each lecture. The heterogeneous network was decomposed into three homogeneous networks: the *lecture-event-lecture* network, the *lecture-author-lecture* and the *lecture-viewer-lecture* network, in which links between two lectures were weighed in proportion to the number of viewers that viewed both lectures.

5.2 Experiment description

In the first set of experiments, a pure text mining approach was used to classify the lectures. The lectures were processed using a standard text mining approach using both TF and TF-IDF weighing. The n-gram length, the minimum term frequency and the cut-off percentage were varied to provide several benchmark performance measures. For each parameter setting the centroid classifier was used on the resulting vectors to predict the categories of individual video lecture.

In the second set of experiments, the vectors obtained through text mining were used to train two classifiers: the k-nearest neighbors classifier and the SVM classifier. For the k-NN classifier, k was set to 20, and for SVM, the SVM-Multiclass (Joachims et al., 2009) was used with the termination criterion set to 0.1 and the trade-off between error and margin set to $5,000$. In addition to the text mining vectors, the SVM and k-NN classifiers were also applied to diffusion kernels (DK) (Kondor and Lafferty, 2002) calculated on the three homogeneous graphs.

The third set of experiments used the methodology proposed in Section 4. The method was deployed on each of the three homogeneous graphs from Section 5.1. For each homogeneous graph, the three classifiers from the first two sets (the centroid classifier, the SVM classifier and the k-NN classifier) were applied to the resulting feature vectors. Next the feature vectors were combined as described in Section 4.4. The feature vectors were combined (a) using equal weights for all feature vectors, or (b) using a stochastic optimizer called differential evolution (DE) (Storn and Price, 1997).

5.3 Evaluation and results

In the experiments described in Section 5.2 the performance of classifiers was evaluated by matching predictions to the pre-categorized classes. Classification accuracy was measured on the top 1, 3, 5 and 10 categories, proposed by the classifier. For each experiment, a 10-fold cross validation was performed. The results are given in Table 1.

The results of the first set of experiments show that using a TF-IDF weighing improves the accuracy of the centroid classifier compared to using TF weights. Varying the minimum frequency, n-gram length and cut-off values resulted in smaller improvements to the performance. The most efficient setting was using $2-$grams and the minimum term frequency of 1, so this setting was used in all BOW constructions in the successive experiments.

The results of the second set of experiments show that the text mining approach performs relatively well and outperforms both the classifier based on the same-event network and the classifier based on the same-author network. The same-author graph contains the least relevant information for the categorization task. The most relevant information is contained in the viewed-together graph. It is noteworthy that the choice of the classification algorithm is less important than the data source from which the similarities between objects are inferred.

Setting	Top 1	Top 3	Top 5	Top 10
First set (text mining)	Accuracy (%)			
TF, $n = 1$, min-freq $= 1$, cut-off $= 0$	53.97	69.46	74.48	81.74
TF-IDF, $n = 1$, min-freq $= 1$, cut-off $= 0$	58.99	75.34	79.50	85.55
TF-IDF, $n = 2$, min-freq $= 1$, cut-off $= 0$	59.60	75.34	80.27	85.20
TF-IDF, $n = 3$, min-freq $= 1$, cut-off $= 0$	59.42	75.77	80.10	85.20
TF-IDF, $n = 2$, min-freq $= 2$, cut-off $= 0$	59.51	76.21	80.79	85.46
TF-IDF, $n = 2$, min-freq $= 3$, cut-off $= 0$	58.13	75.86	80.62	85.20
TF-IDF, $n = 2$, min-freq $= 2$, cut-off $= 0.1$	58.99	75.34	79.15	84.25
Second set (Text mining + DK)	Accuracy (%)			
Text mining + SVM	59.16	73.09	78.28	82.96
Text mining + k-NN	58.47	72.74	78.28	83.91
Text mining + centroid	59.51	76.21	80.79	85.46
DK on viewed-together + SVM	70.75	86.93	90.92	93.68
DK on viewed-together + k-NN	72.74	87.80	90.83	93.94
DK on same-event + SVM	32.00	49.04	54.67	58.65
DK on same-event + k-NN	31.92	47.66	53.37	61.07
DK on same-author + SVM	18.94	27.51	31.22	36.24
DK on same-author + k-NN	19.81	31.74	36.24	43.59
Third set (enriched networks)	Accuracy (%)			
viewed-together + SVM	70.41	85.46	89.71	93.60
viewed-together + k-NN	70.75	84.60	89.36	93.34
viewed-together + centroid	74.91	89.01	92.13	95.33
same-evend + SVM	31.74	50.17	55.97	59.95
same-evend + k-NN	32.34	50.43	55.96	64.79
same-evend + centroid	27.59	46.62	53.63	65.05
same-author + SVM	15.83	24.22	27.33	33.04
same-author + k-NN	15.48	23.70	27.94	32.52
same-author + centroid	14.79	25.52	31.74	42.73
combined - equal weights + centroid	65.73	83.21	87.97	93.42
combined - DE calculated weights + centroid	78.11	91.43	94.03	95.85

Table 1. Accuracies of the algorithms when classifying video lectures.

The results of the third set of experiments showcase the performance of the methodology presented in this Section. Just as in the second set of experiments, the results show that the choice of the classification algorithm results in only minor changes in the classification accuracy compared to the choice of the network decomposition method. The final two rows of the results show that setting equal weights to all feature vectors is far from optimal, as it decreases the accuracy to *below* that of the best individual feature vector. Using differential evolution, on the other hand, improves the performance, as this classifier, using optimized weights and all feature vectors, consistently outperforms other classifiers.

6 Psychology publications categorization case study

We also applied the methodology, presented in Section 4, on almost one million scientific publications from the field of psychology. Like the video lectures, the publications belonged to at least one category from a large set of possible categories. The size of the constructed network allowed us to measure how classifier performance increases as we increase the size of the network on which it is trained. Our motivation was to construct a classifier capable of predicting all categories of a publication with more probable categories listed first. Such a classifier may be used to assist in the manual classification of new psychology articles.

6.1 Data collection

The first step in the construction of a network is data collection. Because there is no central database containing publications in the field of psychology, we decided to crawl the Wikipedia pages connected with psychology.

We collected the information about psychology publications from the reference section of the articles connected to the category Psychology on English Wikipedia. Due to citation formatting inconsistencies, we extracted only the references containing their DOI (Digital Object Identifier).

We examined the hierarchical tree of Wikipedia categories, belonging to the category Psychology. Categories in lower levels of the hierarchy reveal articles that are connected to psychology, but are also strongly connected to other disciplines. Examples include pages from the categories Religion, Evolution, Biology, etc. We decided to stop our collection at level 5. The decision was based on the difference between the number of visited categories and the number of yet uncollected articles at depths 4, 5 and 6. Our final collection therefore includes all Wikipedia subcategories and pages reached from the top level Psychology category in 5 or less steps.

Once we collected the set of DOIs connected with psychology on Wikipedia, we needed a suitable free citation tool that includes academic publications from the field of psychology, allows a crawling script and offers DOI search. Microsoft Academic Search (MAS) satisfied these conditions and was selected as our citation tool. We queried the MAS for each of the collected DOIs. If a publication was found on MAS, we collected the information about the title, authors, year of publication, the journal, ID of the publication, IDs of the authors, etc. Whenever possible we also extracted the publication's abstract. Additionally, we collected the same information for all the publications that cite the queried publications.

6.2 Data set

The result of our data collection process was a network consisting of $953,628$ publications of which $63,862$ 'core publications' were obtained directly from Wikipedia pages. Other publications were cited by the core publications. The publications were linked by $1,539,563$ citation links and had $1,589,144$ authors.

The core publications are labeled with the Wikipedia page referencing it. The remaining publications are labeled with the labels of the core papers citing them. Each publication may be labeled by several different articles. The publications were linked by $1,539,563$ citation links and had $1,589,144$ authors. We collected 93 977 abstracts of the publications, of which 4551 belong to the core publications.

The goal of our experiment was to examine the accuracy of a classifier predicting the labels of publications. To do that, we first decreased the number of labels. Originally, the publications were labeled with Wikipedia pages, resulting in $71,606$ different labels. The Wikipedia pages were replaced with the Wikipedia categories listing them, however, this still left us with $3,173$ labels, of which many, especially the categories visited in the final step of crawling, were rather obscure. Because of this, we decided to only allow the categories visited at levels 0, 1 and 2 to represent labels of publications. The categories at level 3, 4, 5 and 6 were transformed into publication labels by climbing up the category hierarchy to the level 2 categories that link to them. The result is a data set in which every publication is labeled with one or more Wikipedia categories it is associated to.

The heterogeneous network was decomposed into three homogeneous networks: the *paper-author-paper* (PAP) network, the *paper-cites-paper* (PP) network and a symmetric copy of the PP network in which directed edges are replaced by undirected edges (PPS).

Remark 5. It is not fair to use all homogeneous networks for the prediction of publication categories. Because the non-core publications were labeled with labels of core publications that cited them, using the citation graphs (the PP and the PPS graph) would yield too optimistic error estimation, because it would use the very structure that was used to label the publications.

Remark 6. We use both the directed and undirected citation network because both contain information about the publications, but may have very different effects on the PageRank calculation. In the directed network, a publication will share its rank with all publications it is citing, while in the undirected case, it will also share its rank with publications it is cited by. Because the resulting vectors may contain different information about the publication, we decided to calculate both and evaluate their performance.

6.3 Experiment description

The settings used to obtain feature vectors is the same as in Section 5. As in (Grčar et al., 2013), n-grams of size up to 2 and a minimum term frequency of 0 was used to calculate the BOW vectors. For the calculation of $P - PR$ vectors the damping factor was set to 0.85, as this is the standard setting also used in (Page et al., 1999). Where more than one feature vector was calculated, the vectors were concatenated using weights optimized using the differential evolution optimization algorithm (Storn and Price, 1997). In all experiments the

calculated featured vectors were used with a centroid classifier using the cosine similarity distance. This classifier first calculates the centroid vectors of each class (or category) by summing and normalizing vectors belonging to indices of that class. For a new instance with feature vector w, it calculates the cosine similarity distance

$$d(c_i, w) = 1 - c_i \cdot w,$$

which represents the proximity of the instance to class i. We classify the instance into the class for which the distance is the smallest. We also examine the 'top n' classifier, where the classifier predicts that the instance belongs to one of the n classes with the smallest distances. Just like in (Grčar et al., 2013), we consider a classifier successful if it correctly predicts at least one class with which the instance is labeled.

We used the centroid classifier for two reasons. First, the experiments presented in Section 5, show that it performs just as well as the SVM and the k-nearest neighbor classifier, and second, because for large networks, calculating all $P-PR$ vectors is computationally too demanding. As shown in (Grčar et al., 2013), the centroids of classes can be calculated in one iteration of the PageRank algorithm.

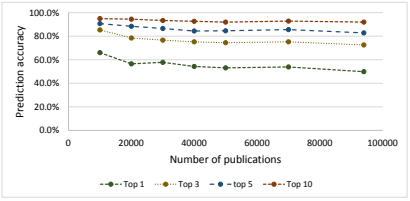
In the first set of experiments we use the publications for which an abstracts were available. Because most of the 93,977 selected publications are not core publications, we construct only two feature vectors for each publication: a bag-of-words (BOW) vector and a $P-PR$ vector obtained from the PAP network. We examine how the predictive power of the classifier increases as we use more publications. We used 10,000, 20,000, 30,000, 40,000, 50,000, 70,000 and 93,977 publications.

In the second set of experiments we use only the core publications for which abstracts are available. While this is the smallest data set, it allows us to use all feature vectors the methodology provides: the BOW vectors and the $P-PR$ obtained from all three networks (PP, PPS and PAP).

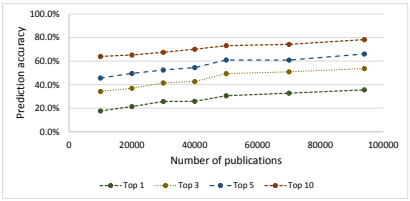
In the third set of experiments all collected papers are used. Because the papers were labeled using citations, the PP and PPS networks are not used. Since abstracts are not available for most of the papers, only the $P-PR$ vectors obtained from the PAP network are used in the classification.

6.4 Evaluation and results

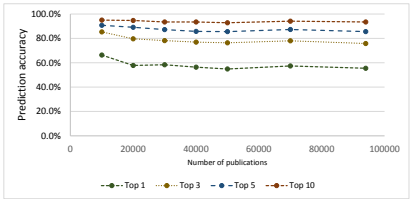
With each of the experiments described in Section 6.3 we predict the labels of publications. Classification accuracy is measured on the top 1, 3, 5 and 10 labels, proposed by the classifier. For each experiment the data set is split into a training, validation, and test set. Centroids of classes are calculated using the training set and concatenated according to the weights optimized using the validation set. The accuracy of the algorithm (the percentage of papers for which the label is correctly predicted) is estimated using the test set. The results are given in Table 2 and Figure 4.



(a) The centroid classifier using BOW.



(b) The centroid classifier using PAP.



(c) The centroid classifier using both BOW and PAP.

Fig. 4. The classification accuracy of classifiers using different amounts of publications to predict labels.

The results of the first set of experiments are shown in Figure 4. The performance of the classifier using BOW vectors does not increase with more data, while the classifier using PAP vectors is steadily improving as more and more publications are added. The classifier using both BOW and PAP vectors consistently outperforms both individual classifiers, showing the utility of combining structural information of the network and the content of the publications. As the performance of the PAP classifier increases, the gap between the BOW classifier and the classifier using both vectors also increases. The accuracies obtained with all $93,977$ publications are also shown in the first part of Table 2.

The results obtained in the second set of experiments are shown in the second part of Table 2. Because more information was extracted from the network, this is the most comprehensive overview of the methodology. The results show that using a symmetric citation network (PPS), i.e. spreading the PageRank in both directions of a citation yields better results than using a non-symmetric citation network (PP). Combining both the PP and PPS vectors does not improve the performance of the classifier, which means that the vectors, obtained from the PP network, carry no information that is not already contained in the PPS network. The same is not true for other vectors. The results consistently show that including more vectors into the classification increases the prediction accuracy: using both BOW and PAP is better than simply using BOW, but adding PP increases the performance even further.

The performance of the PAP classifier on the full network (calculated in experiment 3 and shown in the last row of Table 2) is higher than PAP results

Setting	Top 1	Top 3	Top 5	Top 10
First set	Accuracy (%)			
BOW + PAP	55.5	75.8	85.6	93.5
PAP	35.6	53.7	66.0	78.3
BOW	49.9	72.6	82.8	92.0
Second set	Accuracy (%)			
All	78.6	92.4	94.1	97.4
all but BOW	47.7	62.2	71.7	83.0
all but PAP	45.4	57.9	60.4	96.9
all but PP	44.7	74.3	81.7	93.0
all but PPS	59.4	75.9	80.7	94.4
BOW + PAP	78.7	93.0	95.4	97.5
BOW + PP	79.8	93.0	95.5	97.4
BOW + PPS	79.6	93.0	95.5	97.5
PAP + PP	44.5	58.9	69.4	82.3
PAP + PPS	47.5	61.9	70.6	82.0
PP + PPS	44.4	58.4	68.3	78.9
BOW	78.3	92.9	95.6	97.5
PP	40.7	56.9	67.1	77.7
PPS	44.9	59.6	67.9	80.8
PAP	27.5	45.4	58.2	74.7
Third set	Accuracy (%)			
PAP	38.8	59.3	71.0	81.4

Table 2. Accuracies of the algorithms classifying publications from the field of psychology.

for all other networks, demonstrating that increasing the network size does help the classification. However, the performance is still lower than that of the BOW classifier on smaller networks. It appears that authors in the field of Psychology are not strictly limited to one field of research, making predictions using co-authorship information difficult.

7 Conclusion and further work

While network analysis is an established field of research, analysis of heterogeneous networks is a much newer research area. Methods taking the heterogeneous nature of the networks into account show improved performance, as shown by, e.g., Davis et al. (2011). Some methods like RankClus and others presented in (Sun and Han, 2012) are capable of solving tasks that cannot even be defined on homogeneous information networks (like clustering two disjoint sets of entities). Another important novelty is merging network analysis with the analysis of data, either in the form of text documents or results obtained from various past experiments presented in (Dutkowski and Ideker, 2011; Hofree et al., 2013; Grčar et al., 2013).

This chapter presents a methodology for mining text-enriched heterogeneous information networks which combines the information from heterogeneous networks with textual data. Compared to the methods described in Section 3, the presented methodology combines aspects of network analysis with aspects of text mining. The methodology is applied to text-enriched heterogeneous networks and does not present an alternative, but rather an expanded way of data analysis, compared to these methods. Thus, many other network analysis techniques, especially those that focus on discovering information about *nodes* in the network, can be modified to use it. The presented methodology is comparable to the methods described in Subsection 3.3, in which data enriched networks are analyzed with methods that consider both the network structure and the data enriching the nodes. However, unlike those methods, our methodology deals with textual information enriching network nodes and thus requires a combination of network analysis and text mining. While there are many applications in which text analysis is combined with network mining, they usually apply text analysis to extract knowledge in the form of a network (see Section 3.3) and then apply network analysis methods to further analyze it. Unlike these approaches, the approach presented in this Chapter combines two separate knowledge sources and joins them into a single representation.

We show-case the performance of the methodology on two data sets. The results from the VideoLectures.NET data show that using the methodology increases classification accuracy compared to using only texts or only structural information about the instances. The results from the psychology papers experiment show that the relational information hidden in the network structure is beneficial to classification and that its usefulness increases for larger networks.

In our experiments on publications from the field of psychology, we only used a part of information collected about psychology publications. In future, we plan to examine how to incorporate temporal information into the described methodology. We have already collected the year of publication which allows us to observe the dynamics of categories. This additional information may also be used to improve the classification accuracy.

Our approach uses network enrichment with text data and heterogeneous network decomposition and then combines the produced vectors into a single score. Alternative is to use Cartesian product of multiple vector spaces to form a tensor representation as presented by Nickel (2013). The tensor space grows exponentially with the number of dimensions but recently several decompositions have been proposed which allow processing in a big data setting (Vervliet et al., 2014; Cichocki, 2014). The suggested decompositions allow multi-relational learning, which is a path we want to test in our future work.

In further work we plan to use a combination of network analysis and data mining on a problem of biological networks enriched with experimental data and texts. An experimental data-enriched heterogeneous network centered around genes can be constructed in which network information will be enriched with papers mentioning the genes.

Acknowledgments

The presented work was partially supported by the European Commission through the Human Brain Project (Grant number 604102) and by the Slovenian Research Agency project "Development and applications of new semantic data mining methods in life sciences" (Grant number J2-5478).

References

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social Networks*, 25(3):211–230.

Barabási, A. L., Jeong, H., Néda, Z., Ravasz, E., Schubert, A., and Vicsek, T. (2002). Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3-4):590–614.

Bilmes, J. (1997). A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, ICSI.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

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

Chen, B., Ding, Y., and Wild, D. J. (2012). Assessing drug target association using semantic linked data. *PLoS Computational Biology*, 8(7).

Chen, H. and Sharp, B. M. (2004). Content-rich biological network constructed by mining pubmed abstracts. *BMC Bioinformatics*, 5:147.

Cichocki, A. (2014). Era of big data processing: A new approach via tensor networks and tensor decompositions. *arXiv preprint arXiv:1403.2048*.

Consortium (2000). Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature genetics*, 25(1):25–29.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6):453–482.

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.

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

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

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.

Hwang, T. and Kuang, R. (2010). A heterogeneous label propagation algorithm for disease gene discovery. In *Proceedings of SIAM International Conference on Data Mining*, pages 583–594.

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.

Jenssen, T.-K., Laegreid, A., Komorowski, J., and Hovig, E. (2001). A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28.

Ji, M., Sun, Y., Danilevsky, M., Han, J., and Gao, J. (2010). Graph regularized transductive classification on heterogeneous information networks. In *Proceedings of the 25th European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 570–586.

Joachims, T., Finley, T., and Yu, C.-N. J. (2009). Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59.

Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30.

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

Kok, S. and Domingos, P. (2008). Extracting semantic networks from text via relational clustering. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, ECML PKDD '08, pages 624–639. Springer-Verlag.

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.

Kralj, J., Valmarska, A., Robnik Šikonja, M., and Lavrač, N. (2015). Mining text enriched heterogeneous citation networks. In *Proceedings of the 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining*.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791.

Lytras, M. and Sheth, A. (2010). *Progressive Concepts for Semantic Web Evolution: Applications and Developments*. IGI Global.

Newman, M. (2001a). Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102.

Newman, M. E. J. (2001b). The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*, 98(2):404–409.

Nickel, M. (2013). *Tensor Factorization for Relational Learning*. PhD thesis, Ludwig–Maximilians–Universitaet Muenchen.

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

Plantié, M. and Crampes, M. (2013). Survey on social community detection. In Ramzan, N., Zwol, R., Lee, J.-S., Clüver, K., and Hua, X.-S., editors, *Social Media Retrieval*, Computer Communications and Networks, pages 65–85. Springer London.

Rakotomamonjy, A., Bach, F., Canu, S., and Grandvalet, Y. (2008). SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521.

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.

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

Sun, Y., Han, J., Zhao, P., Yin, Z., Cheng, H., and Wu, T. (2009a). RankClus: integrating clustering with ranking for heterogeneous information network analysis. In *Proceedings of the International Conference on Extending Data Base Technology*, pages 565–576.

Sun, Y., Yu, Y., and Han, J. (2009b). 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.

Van Landeghem, S., De Bodt, S., Z.J., D., Inze, D., and Van de Peer, Y. (2013). The potential of text mining in data integration and network biology for plant research: A case study on arabidopsis. *The Plant Cell*, 25(3):794–807.

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).

Vervliet, N., Debals, O., Sorber, L., and De Lathauwer, L. (2014). Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis. *Signal Processing Magazine, IEEE*, 31(5):71–79.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442.

Yang, B., Liu, D., and Liu, J. (2010). Discovering communities from social networks: Methodologies and applications. In *Handbook of Social Network Technologies and Applications*, pages 331–346. Springer.

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

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.