Universiteit
Antwerpen

Faculty of Applied Economics

Dissertation

Classification within network data with a bipartite structure

Thesis submitted in order to obtain the degree
of
Doctor in Applied Economics

Author:
Marija Stankova

Supervisor:
Prof. dr. David Martens

September, 2016

Supervisor:
Prof. dr. David Martens (University of Antwerp, Belgium)

Members of the Examination Committee (in alphabetical order):
Prof. dr. Toon Calders (University of Antwerp, Belgium)
Prof. dr. Tina Eliassi-Rad (Northeastern University, USA)
Prof. dr. Herbert Peremans (University of Antwerp, Belgium)
Prof. dr. Foster Provost (New York University, USA)
Prof. dr. Johan Springael (University of Antwerp, Belgium)

ISBN 978-90-8994-154-1
9 789089 941541

Cover design: University of Antwerp
Printing and binding: Universitas, Antwerp

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my advisor David, for giving me the opportunity to pursue a Ph.D. He has been a tremendous support over the past four years and an example of a successful researcher, professor and entrepreneur. His guidance and dedication helped me grow as a research scientist and learn a lot in the field of data science. I appreciate all his contributions of ideas, time and funding that made this thesis possible. Furthermore, I would like to thank him for also making this whole Ph.D. experience a lot of fun through the many dinners, drinks and conferences with the Applied Data Mining team.

I am also very thankful to Foster, to whom a owe a lot. It has been my pleasure to collaborate with one of the leading researchers in our field since the beginning of my Ph.D. I have learned a lot from our enriching discussions and his profound knowledge of the topic. Moreover, I am immensely grateful for welcoming me to conduct a part of this research at New York University.

I would like to extend my gratitude to the other members of my examination committee, Tina Eliassi-Rad, Toon Calders, Herbert Peremans and Johan Springael, for taking their time to carefully evaluate my work. Their expertise, valuable comments and insightful questions helped me improve the quality of this thesis.

Furthermore, I would like to thank my family for all their love and encouragement. To my mom and dad, who unconditionally supported me in all my pursuits in life and to my closest family - grandma, grandpa, Kiki, Gabi, uncles Kole and Orce and aunt Suze for always being there for me. You have been my greatest support in life and I dedicate this book to you.

This Ph.D. experience would not have been the same without my wonderful ENM colleagues from the fifth floor to whom I have a lot to thank for. To my fellow data scientists Jellis and Enric, for making our B519 office such a fun and enjoyable workplace, Julie for the sincere friendship and the immense support even from the other side of the world, Daniel for the many inspiring conversations over dinners and drinks, Jelle, Jaco, Alexander, Renata, Sanne, Annelies, Kevin, Christof, Sofie, Ellen, Florian, Jochen, Nicholas, Marco, Luca, Floor, Jasmine, Dorien, Christine,... for all the great discussions and fun times during our team building weekends, housewarming parties, board game evenings and more. Beside my colleagues from the University of Antwerp, I was also very fortunate to have a great group of colleagues at NYU Stern, that helped making my stay in NYC one of the best experiences so far. Vilma, Jessica, Apostolos, Konstantin, Rob,... thank you for all the fun talks at the office or during our lunch and coffee breaks at Washington Square Park. I am also thankful to Vinayak and the whole Lenddo team for our collaboration on the credit scoring paper and the pleasant afterwork dinners and drinks.

My time in Belgium was also made enjoyable in large part due to the many great people that I have met outside of the University. I am particularly grateful to Kole and Gogi for being such good friends and an incredible support during the last difficult stages of the Ph.D. Moreover, Bake, Viktor (x2), Boco, Igor, Vera, Giovanni, Jan, Andres, Gonzalo, Peter, Rafael, Thanasis..., thank you for all the unforgettable memories throughout the years. I am also truly happy to have met my lovely roommates from Ghent, Kiki and Isa, for our many adventures and long lasting friendship. Furthermore, I am indebted to Karel, whose care and endless encouragement is so appreciated. During the time spent in New York, I have met a lot of remarkable people that made my stay truly pleasant. For this, I am largely thankful to Ane for introducing me to many people and to Martin for always being ready to explore the vibrant NYC nightlife scene with me. Lastly, I am very grateful for my group of lifelong friends: Simona, Anci, Ema, Tanja, Caci, Sanja, Eli, Caki, Deki, Kire, Diko, Tode, Viksa, Aksi, Zoki, Fico,... for their invaluable support and all the beautiful moments we have shared so far.

Finally, I would like to thank the Flemish Government for providing me with financial support through the University Research Fund (BOF) to conduct my research. I am also very thankful to the personnel of the Faculty of Applied Economics, especially to An and Esmeralda, for helping me out and guiding me through all the administrative procedures.

Marija Stankova
Antwerp, 2016

English preface

During the last two decades, data science has emerged as one of the main areas of interest for many companies, governments and academia. While it used to be a rather difficult and time consuming process to collect data, today we witness the unpreceded wealth of information generated from social network interactions, search queries, payment transactions, clickstreams, logs, health records, mobile phone usage, sensors, etc that can be automatically collected and analysed. Moreover, the reduced costs for storing the data in combination with the increased computational power and the development of new technologies have lead to significant advances in the field of data science. Many companies today are relying on data science to gain useful knowledge and insights from data. Leveraging this knowledge, they can make more informed decisions and use this advantage to position themselves better in the market. Furthermore, data science can help the public sector work more efficiently, create better policies and offer personalised services to the citizens. While this presents the significant gains that can be achieved with data science, the potential dangers include sensitive ethical and privacy issues that should be carefully and properly managed.

Many of the real-world datasets used in the field correspond to bigraph settings, such as data about users rating movies or people visiting locations. Although some work exists regarding such bipartite graphs (bigraphs), no general network-oriented methodology has been proposed yet to perform node classification. Prior literature has generally seen classification within this type of data from a classical perspective as classification with massive and sparse feature data. We, on the other hand, propose an alternative network based formulation, i.e. a three-stage framework for doing classification in bipartite data via projection (see **Chapter 3**). This projecting approach transforms the bigraph into a weighted unigraph version that preserves information about the underlying bigraph and allows the practitioners to make use of the wealth of unigraph techniques already available. The framework opens up the design space for experimenting with existing or new methods in the different stages and creating new techniques by mixing-and-matching the choices. Furthermore, we take a theoretical approach to extend this work into a multiple-stage framework that includes information about the bigraph link weights. As we discuss later on in **Chapter 4**, this results in creating more representative projections, which in turn improves the prediction results. Additionally, we validate our designs with two application studies. In **Chapter 5**, we discuss how we can use this network based formulation to help detect companies that fraudulently reside outside of Belgium for tax benefits. Moreover, in **Chapter 6** we apply the framework to bigraphs of

Facebook behavioural data in order to help assess the creditworthiness of microloan applicants. Lastly, in **Chapter** 7 we discuss the problem of classification within bigraph data from the standard perspective with high-dimensional features and elaborate on how helpful it is to consider the nonlinearities in the data through higher order interaction features.

Dutch preface

Gedurende de laatste twee decennia hebben datawetenschappen zich gemanifesteerd als een van de belangrijkste interessedomeinen voor bedrijven, overheden en academische instellingen. Terwijl het vroeger een moeilijk en tijdrovend proces was om data te verzamelen en te verwerken, zien we vandaag een weelde aan informatie die gegenereerd wordt door sociaal netwerk interacties, zoekopdrachten, betalingsverkeer, clickstreams, logs, gezondheidsgegevens, mobiele telefonie, sensoren, enz. die automatisch kunnen verzameld en geanalyseerd worden. Daarenboven dragen gereduceerde dataopslagkosten in combinatie met een toenemende rekenkracht en de ontwikkeling van nieuwe technologieij tot significante ontwikkelingen in het domein van de datawetenschappen. Tegenwoordig maken vele bedrijven gebruik van dit onderzoeksdomein om bruikbare kennis en inzichten te bekomen uit hun databronnen. De toepassing van deze kennis kan resulteren in meer geormeerde beslissingen en finaal leiden tot een betere positionering in de markt. Daarenboven kunnen datawetenschappen bijdragen tot een efficiere werking van de publieke sector, aanleiding geven tot betere beleidsmaatregelen en het aanbieden van een verbeterde persoonlijke dienstverlening. De bovenstaande opsomming illustreert de vele winsten die kunnen worden bereikt aan de hand van datawetenschappen. Daartegenover dient er zorgvuldig te worden omgesprongen met ethische en privacy knelpunten.

Vele van de hedendaagse datasets gebruikt in dit domein corresponderen met een bipartiete structuur, zoals data in verband met gebruikers die films raten of personen die zekere locaties bezoeken. Hoewel er reeds onderzoek is verricht met betrekking tot zulke bipartiete grafen (bigraphs), werd er nog geen algemene netwerk-georieerde methodologie voorgesteld om classificatie van knooppunten uit te voeren. Voorgaande literatuur heeft, in het algemeen, classificatie van dergelijke data beschouwd vanuit het klassieke perspectief van classificatie met immense en ijle (spaarse) attribuutmatrices. Wij daarentegen stellen een alternatieve netwerk-gebaseerde formulering voor, zijnde een drie-stadia raamwerk voor knooppuntclassificatie in bigraphs door projectie (zie **Hoofdstuk 3**). Deze aanpak via projectie transformeert de bigraph in een gewogen netwerk (unigraph) die informatie met betrekking tot de onderliggende bigraph bewaart en aan beoefenaars de mogelijkheid geeft om gebruik te maken van de immense weelde van reeds bestaande unigraph technieken. Het raamwerk laat verschillende experimenten met bestaande of nieuwe methoden toe in de verschillende stadia en cret nieuwe technieken door de ontwerpkeuzes te mengen en te koppelen. Daarenboven hanteren we een theoretische aanpak om dit werk uit te breiden tot een meer-stadia raamwerk die informatie met betrekking tot de gewichten van de verbindingen in de bigraph incorporeert. Zoals we verder bespreken in **Hoofdstuk** 4, resulteert dit in de creatie van meer representatieve projecties die de predictieresultaten verbeteren. De validatie van onze ontwerpen gebeurt op basis van twee toepassingsgerichte studies. In **Hoofdstuk** 5 bespreken we hoe deze op netwerk gebaseerde formulering kan bijdragen tot het detecteren van bedrijven die domiciliefraude plegen. Daarenboven, in **Hoofdstuk** 6, wordt het raamwerk toegepast op bigraphs van Facebook gedragsdata om de kredietwaardigheid van microkrediet aanvragers te beoordelen. Tenslotte bespreken we het probleem van classificatie in bigraph data vanuit het standaard perspectief met hoog-dimensionale attributen in **Hoofdstuk** 7. Daar gaan we dieper in op het nut van de beschouwing van niet-lineariteiten in de data door opname van hoge orde interactie variabelen.

Contents

Part I

INTRODUCTION

1

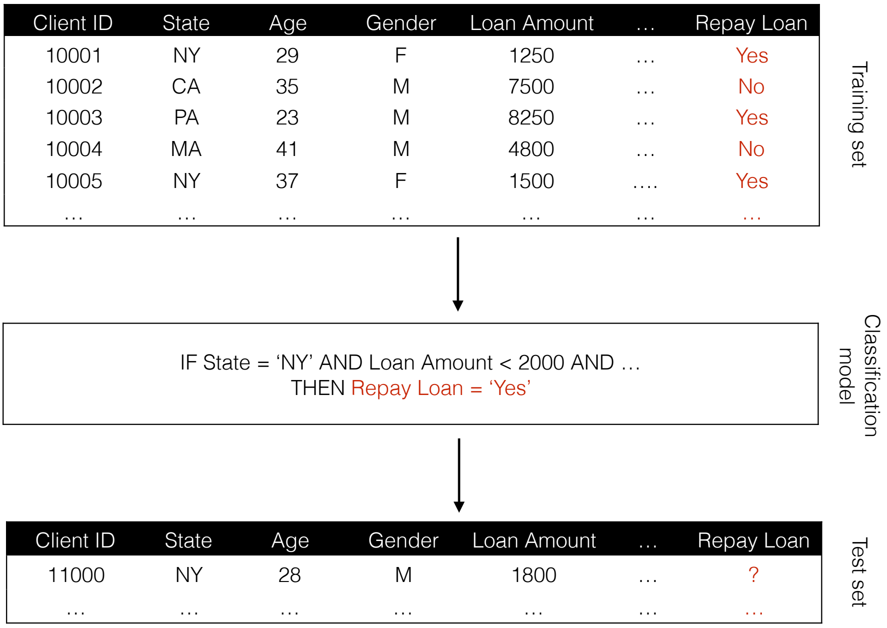
Introduction

This introductory chapter provides a brief overview of the basic notations and techniques that are used throughout the book. As stated previously, data science involves the use of automated methods to extract useful information and knowledge from data. Although the term data science is often interchangeably used with data mining, as discussed by Provost and Fawcett (Provost and Fawcett 2013), data science can be considered as the set of principles that underlay the process of extracting meaningful patterns from data. Data mining on the other hand, entails the use of techniques that implement these principals for the aforementioned goal. Nowadays, many datasets used in the field are *Big Data*, meaning they are so big and complex that they can not be processed using traditional data processing tools (Jacobs 2009). Another definition describes the characteristics of Big Data with the help of the three *V*s as (Laney 2001): having such large sizes that it is difficult to store them in traditional databases (Volume), the data are arriving in continuous streams and need to be processed in real-time (Velocity) and there is a need to harness different types of unstructured data (Variety) [1].

There exist many data mining techniques that can be used for different *tasks*, such as classification, regression, similarity matching, clustering, co-occurrence grouping, profiling, link prediction, data reduction and causal modelling (Provost and Fawcett 2013). The focus of this work is on the task of *classification*, where historic data about many *instances* are used to find patterns and build models. In a typical setting, we have data about multiple variables called *features* or *attributes*. Based on these data, we would like to determine the value (*class label*) of the categorical *target* variable(s) [2]. Each instance is represented with a fixed-length vector of feature values, known as a *feature vector*. We use a *training set* for which the features and the target variable(s) are known, to build a classification model that enables us to determine the label of new, unseen instances. The classification techniques can yield a discrete class as an

[1] In recent literature several other Vs have been added to the definition (Marr 2015).

[2] In case of a numerical target variable, the data mining task is called *regression*.



Client ID	State	Age	Gender	Loan Amount	…	Repay Loan
10001	NY	29	F	1250	…	Yes
10002	CA	35	M	7500	…	No
10003	PA	23	M	8250	…	Yes
10004	MA	41	M	4800	…	No
10005	NY	37	F	1500	….	Yes
…	…	…	…	…	…	…

IF State = 'NY' AND Loan Amount < 2000 AND …
THEN Repay Loan = 'Yes'

Client ID	State	Age	Gender	Loan Amount	…	Repay Loan
11000	NY	28	M	1800	…	?
…	…	…	…	…	…	…

Figure 1.1.: Using a classification technique to predict whether a new customer will fully repay its microloan.

output or they can give a probability estimate (*score*) that the instance belongs to a certain class.

As a running example we consider a microfinance application (see Figure 1.1), where the goal is to predict the trustworthiness of the microloan applicants. Based on the bank or company's historic data about previous customers, a classification model can be built from the clients' socio-demographic data, as well as additional features engineered from their social network accounts. The target variable represents whether the loan applicant would fully repay the loan or not. When a new customer applies for a loan, the lender can use the model to make an automated decision based on the customer's characteristics. In prior literature, classification has been used for other applications as well including churn prediction (Moeyersoms and Martens 2015; Verbeke, Martens, and Baesens 2014), fraud detection (Bhattacharyya et al. 2011), credit scoring (Huang, Chen, and Wang 2007), targeted marketing (Shaw et al. 2001), medical diagnosis (Soni et al. 2011), to name just a few. Furthermore, there is a wealth of different classification techniques that can be used to solve classification problems like logistic regression (Friedman, Hastie, and Tibshirani 2001), Support Vector Machines (SVM) (Scholkopf and Smola 2001), *k* - nearest neighbour (kNN) (Friedman, Hastie, and Tibshirani 2001), Artificial Neural Networks (ANN) (Hornik, Stinchcombe, and White 1989), decision trees like C4.5 (Witten and Frank 2005), etc. The classification techniques that are relevant to this thesis are shortly described in Section 1.1. One of the most important factors when choosing a learning technique is based on the predictive performance on a separate, hold out set of data (*test set*). Using an out of sample data for testing the performance of the classifier ensures that we are not overfitting on the training set and that the model can generalise well on new data. To gain larger confidence in the performance of the model, we can use a *k*-fold *cross-validation* procedure, to create k different splits of the data in a training and test set and calculate the average performance over all folds. The evaluation of the models can be done with different performance measures as described in Section 1.2. Additionally, in some domains like medical diagnosis or credit scoring the models need to be *comprehensible*, since the decisions made by the classifier also need to be interpreted and verified (Gregor and Benbasat 1999; Martens, Baesens, et al. 2007; Martens and Provost 2014b). Another requirement might be the *justifiability* of the model, that is the extent to which the model is aligned with the existing domain knowledge (Martens and Baesens 2010).

1.1 Classification techniques

In what follows, we give an overview of several classification techniques that are used in this work. We must note that these summaries of the techniques are not exhaustive and serve only as introductions in the field. We do, however, refer to relevant textbooks that can provide more details to the interested readers. Furthermore, we also discuss three performance measures for evaluating the classification techniques. These measures are widely used in the data mining literature and are an appropriate choice for our applications.

k-Nearest Neighbour

k-Nearest Neighbour is a classification technique that determines the class label of an instance by using class information about the *k* most similar instances to it (nearest neighbours) (Friedman, Hastie, and Tibshirani 2001; Provost and Fawcett 2013). The similarity between the instances can be measured using different types of vector similarity or distance [3] metrics such as Euclidean, Manhattan, Jaccard, Cosine distance, etc. The most basic strategy is to classify the instance as having the most common class among these neighbouring instances. However, this simple approach gives an equal importance to all the neighbours, neglecting the fact that some of them might be more similar to the instance. One way to overcome this problem is to use weighted voting instead of majority voting, where the neighbouring class labels are weighted by the similarity between the instances. As for the number of

[3] The distance score can be transformed into similarity using the inverse of the squared distance or some other commonly used measures.

neighbours k that should be considered for classification, the best choice can be experimentally determined on a separate validation set for every dataset. One of the mayor drawbacks of the k-Nearest Neighbour technique is its inability to scale to very large dimensions, something that we discuss into more details in Section 3.1.1.

Support Vector Machines

The Support Vector Machine (SVM) is one of the most commonly used classification algorithms (Bishop 2006; Scholkopf and Smola 2001). The idea behind the classifier is to divide the instances from the different classes, which are mapped as points in high-dimensional space, with a gap that is as wide as possible. For this, we need to find an optimal decision boundary that maximizes the distance between this boundary and the nearest data point of each class (margin) (see Figure 1.2 (left)). The new instances are then classified depending on which side of the gap they are mapped.

More specifically, if we have data about n training instances with input vectors of features $\mathbf{x_1}$,$\mathbf{x_2}$...$\mathbf{x_n}$, and corresponding target variables y_1,y_2...y_n ($y_i \in$ {-1,1}), the classification problem can be described with Equation 1.1, where $\phi(\mathbf{x})$ is a feature-space transformation, $\mathbf{w}$ is a weight vector and b is a bias term. The class of a new instance can be determined as the sign of $y(\mathbf{x})$.

$$y(\mathbf{x}) = \mathbf{w^T}\phi(\mathbf{x}) + b \tag{1.1}$$

In order to control the sensitivity of the SVM to possible outliers, the instances are allowed to be misclassified. However, each misclassification introduces a penalty in the form of a slack variable ξ_i ($\xi_i \geq 0$), that increases its value the further the point i is from the boundary (see Figure 1.2 (right)). Thus, $\xi_i = 0$ when the instance is correctly classified and $\xi_i = |y_i - y(\mathbf{x_i})|$ in case of misclassification (Bishop 2006). This goal of maximizing the margin, while having a low penalty for misclassification can be formulated with the following optimisation problem (Equation 1.2), where C ($C > 0$) is a tuning parameter that controls the trade-off between the two objectives. For more details we refer to (Bishop 2006).

$$\begin{aligned} &\text{min} && \frac{1}{2}\mathbf{w^T}\mathbf{w} + C\sum_{i=1}^{n}\xi_i && && (1.2)\\ &\text{subject to} && y_i(\mathbf{w^T}\phi(\mathbf{x_i}) + b) \geq 1 - \xi_i, && i = 1, \ldots, n. && (1.3)\\ & && \xi_i \geq 0, && i = 1, \ldots, n. && (1.4) \end{aligned}$$

Beside linear classification, the SVM classifier can perform non-linear classification by implicitly mapping the inputs into high-dimensional feature spaces (also known as the "kernel trick"). This means that the training instances which are not linearly separable in the original space, can be linearly separable in a non-linear feature space

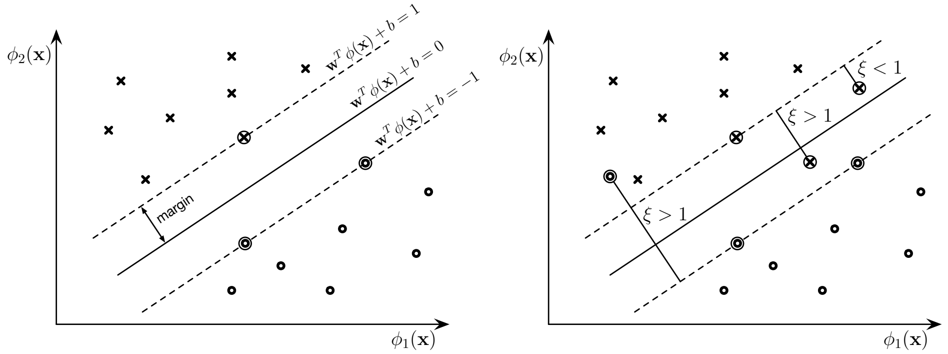


Figure 1.2.: Classification with Support Vector Machines.

implicitly defined by a kernel function $K(\mathbf{x_i},\mathbf{x}) = \phi(\mathbf{x_i})^T\phi(\mathbf{x})$. It can be used with the dual form of Equation 1.1, as represented in Equation 1.5, where the αs ($\alpha_i \geq 0$) are known as Lagrangian multipliers. There exist many kernel functions that satisfy the Mercer theorem, however, in what follows we will only consider the basic linear kernel (Equation 1.6) and the RBF (Gaussian) kernel (Equation 1.7).

$$y(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x_i},\mathbf{x}) + b \tag{1.5}$$

$$K(\mathbf{x_i},\mathbf{x}) = \mathbf{x_i^T}\mathbf{x} \tag{1.6}$$

$$K(\mathbf{x_i},\mathbf{x}) = exp(-\frac{||\mathbf{x}-\mathbf{x_i}||^2}{2\cdot\sigma^2}) \tag{1.7}$$

1.2 Performance measures

Confusion Matrix

A confusion matrix is a simple, tabular representation of the classifier performance (Fawcett 2006). Table 1.1 depicts a confusion matrix for a binary classification problem [4], where the columns signify the actual classes of the instances and the rows represent the predicted classes by the classifier. In cases where the output of the model is a probability estimate, different thresholds can be applied in order to determine the class membership of the instances. The diagonal entries in the matrix

[4] In case of a multiclass problem with n classes, the confusion matrix has n x n entries (Provost and Fawcett 2013).

Table 1.1.: A confusion matrix for a binary classification problem. The entries along the main diagonal represent the correctly predicted instances and the off-diagonal entries signify the errors.

		Actual class	
		Positive	Negative
Predicted class	Positive	True Positives (TP)	False Positives (FP)
	Negative	False Negatives (FN)	True Negatives (TN)

give the number of correctly classified positive (true positives (TP)) and negative (true negatives (TN)) instances in the dataset. The errors of the classifier are given as the false positives (negative instances that are classified as positive (FP)) and false negatives (positive instances classified as negative (FN)). By combining these statistics, we can obtain several commonly used performance measures. For instance, the *accuracy* which represents the fraction of correctly classified instances, can be calculated as the sum of the diagonal elements (TP+TN) over the total number of instances. This measure is not recommended to be used with highly imbalanced datasets since it can provide misleading results (Provost and Fawcett 2013). Furthermore, the *sensitivity* can be calculated as the fraction of correctly classified positive instances (Equation 1.8) and *specificity* is defined as the fraction of correctly classified negative instances (Equation 1.9).

$$sensitivity = \frac{TP}{TP + FN} \tag{1.8}$$

$$specificity = \frac{TN}{TN + FP} \tag{1.9}$$

Area Under the ROC - Curve (AUC)

The Receiver Operating Characteristics (ROC) Curve provides a graphical representation of the classifier's performance, by plotting the gains (sensitivity on the y-axes) versus the costs (1-specificity on the x-axes) (Fawcett 2006). In Figure 1.3 we depict the ROC curves for three classifiers that have probability estimate outputs. The curves are obtained by plotting the stats (sensitivity and 1-specificity) for different thresholds as various points in the graph. Note that several points in the ROC space have a special meaning. The lowest left point (0, 0) signifies a conservative model that never classifies an instance as positive (there are no TP or FP in the confusion matrix). Conversely, the upper right point (1, 1) is produced by a very liberal classifier that predicts only positive outcomes. A classifier that perfectly predicts the classes of all instances has a value of (0, 1) and a classifier with random performance has a value that lies within the diagonal line y = x of the ROC space. In order to compare the performances of different classifiers, we calculate the *area under the ROC curve (AUC)*, which is a scalar value between 0 and 1. As discussed by Fawcett (Fawcett 2006), the AUC value of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. Using this measure, we can conclude that in Figure 1.3 Classifier A with an AUC value of 0.6995 has the best performance, followed by Classifier B with an AUC of 0.6438. They both perform better than a random classifier, which has an AUC value of 0.5.

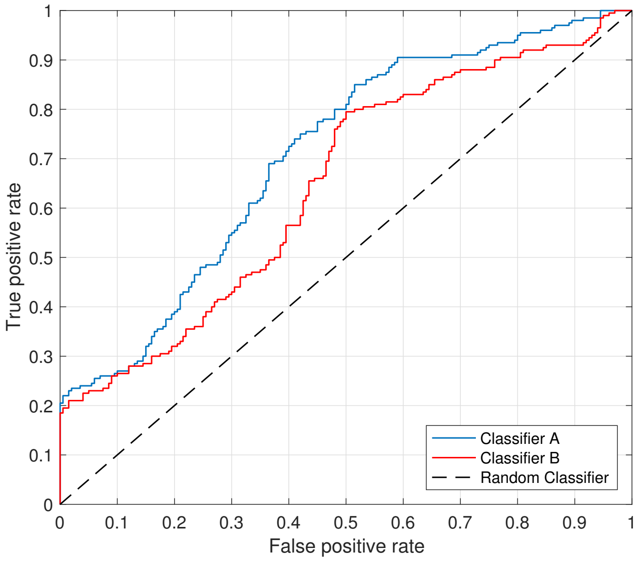


Figure 1.3.: The receiver operating characteristics (ROC) curves for three classifiers.

Lift Curve

The lift curve is a more intuitive representation of the classifier's performance than the previously discussed ROC curve. It can be interpreted as the improvement that the classifier achieves compared to a random classifier. Lift is calculated as the ratio of the positives rate of the top ranked instances by the classifier and the positives rate when randomly selecting the same number of instances. The lift curve plots the lift of the classifier as a function of the percentage of instances considered (Provost and Fawcett 2013). In Figure 1.4 we depict the lift curves for the previously discussed classifiers [5].

[5] Note that the dataset used in this example has 50% positive and 50% negative instances.

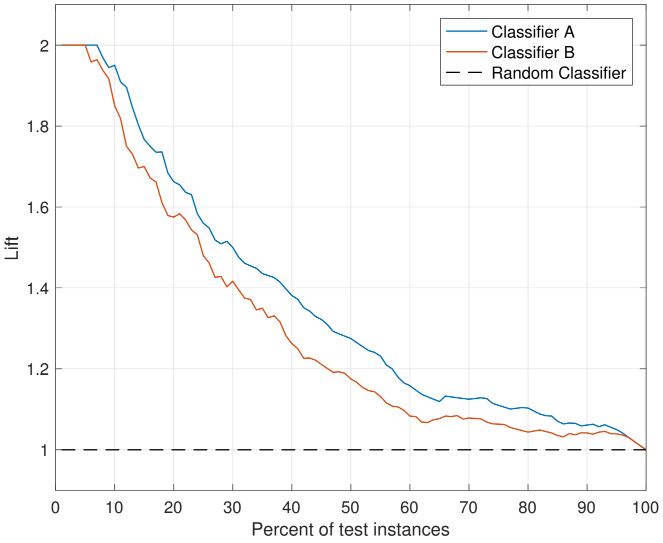


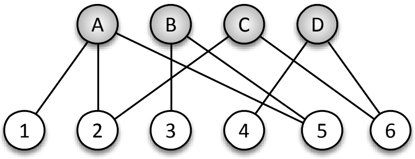
Figure 1.4.: The lift curves for three classifiers.

2

Bipartite graphs (bigraphs)

In the previous chapter, we discussed how the classification techniques can be applied over instances represented as feature vectors. These vectors have the same structure and are typically assumed to be independent and identically distributed (i.i.d.) (Jensen, Neville, and Gallagher 2004). This means that it is possible to infer the class membership of an instance without taking into account the labels of the other instances, because they are independent. Such an assumption represents a simplification of most real-world datasets, as in many cases the entities are somehow related. If we go back to our running example and consider only vectors of socio-demographic data about the microloan applicants, then we neglect the valuable information about how the applicants are connected on Facebook. In many cases, these Facebook friendships can uncover important insights as people tend to be similar to their friends (a principle known as homophily in the social network literature (Easley and Kleinberg 2010; McPherson, Smith-Lovin, and Cook 2001)). As we discuss later in Chapters 5 and 6, taking into account both structured and relational data yields significantly better predictions than considering only one type of data.

Relational (network) data can be modelled with graphs, where the entities are represented as *nodes* and the relationships between them as *links (edges)*. In this work we focus on a special type of relational datasets that can be modelled as bipartite graphs (bigraphs, sometimes also referred to as 2-mode or affiliation networks). They are defined as graphs with two types of nodes (bottom and top nodes) and links that connect only nodes of different type (Latapy, Magnien, and Del Vecchio 2008). Think for example of relationships based on companies' board members (Seierstad and Opsahl 2011), users meeting at locations or events (Eagle and Pentland 2006), users rating different products (Ziegler et al. 2005), consumers making payments to merchants (Martens and Provost 2011), mobile devices visiting locations (Provost, Martens, and Murray 2012), authors collaborating on scientific papers (M. E. Newman 2001b), people communicating on online forums (Opsahl 2011), actors playing in the same movies (Guillaume and Latapy 2006), words occurring in the same sentence/search query (Cancho and Solé 2001; Guillaume



$$\begin{array}{c|cccc} & A & B & C & D \\ \hline 1 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 3 & 0 & 1 & 0 & 0 \\ 4 & 0 & 0 & 0 & 1 \\ 5 & 1 & 1 & 0 & 0 \\ 6 & 0 & 0 & 1 & 1 \end{array}$$

Figure 2.1.: Bigraph and its corresponding adjacency matrix representation.

and Latapy 2006), proteins involved in the same metabolic processes (Guillaume and Latapy 2006), etc. In many cases, there also exists some strength or capacity associated with the relationships between the entities that can be modelled as the *weight* of the bigraph links.

A bigraph (Figure 2.1 (left)) can formally be defined as the triplet $G = (\top, \perp, E)$, where $\top$ denotes a set of *top nodes*, $\perp$ is a set of *bottom nodes* and $E \subseteq \top \times \perp$ is a set of *links (edges)*. It can also be represented by an adjacency matrix (Figure 2.1 (right)), where the rows correspond to the bottom nodes and the columns present the top nodes. An element x_{ij} in the adjacency matrix has a value of 1, if the corresponding bottom node i and top node j are connected and otherwise 0. Although there exist many notations for the analysis of the standard graphs with one type of nodes (*unigraphs*), the literature regarding bigraphs is very scarce. Latapy et al. (Latapy, Magnien, and Vecchio 2008) introduced several basic methods for describing the bigraphs, which we also use throughout the book. For each bigraph dataset $G = (\top, \perp, E)$, $n_\top$ denotes the number of top nodes $n_\top =\mid \top \mid$, $n_\perp$ the number of bottom nodes in the bigraph $n_\perp =\mid\perp\mid$ and m the total number of edges. The average degree of the top and the bottom nodes can be calculated as $k_\top = \frac{m}{n_\top}$ and $k_\perp = \frac{m}{n_\perp}$ respectively, and the total average degree k over the whole bigraph as $k = \frac{2m}{n_\top + n_\perp}$. The density of the graph, which represents the probability that two randomly chosen nodes from the distinct node sets are connected, is equal to $\delta(G) = \frac{m}{n_\top \cdot n\perp}$. These notations are basically extensions of the existing measures for unigraphs. For more details on other adapted measures to the bigraph case, including clustering coefficient and centrality measures, we refer to the studies of Borgatti and Halgin (Borgatti and Halgin 2011) and Latapy et al. (Latapy, Magnien, and Vecchio 2008).

In order to use the wealth of techniques and notations available in the literature for unigraphs, many researchers turn to a second approach of transforming the bigraph into a unigraph (*bigraph projection*) and then applying the tools for unigraph analysis. A projection is created by connecting the nodes of one of the two sets of the bigraph, if they share at least one neighboring node from the other set of nodes (see Figure 2.2). This means that the projection of the bottom nodes ($\perp$ projection), defined as $G' = (\perp, E')$ with a set of edges $E' \subseteq \perp \times \perp$, can be obtained

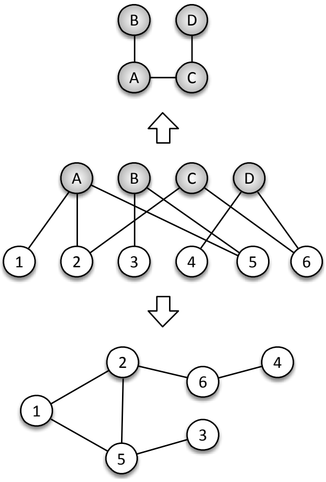


Figure 2.2.: A bigraph and its top and bottom node projections.

by connecting the nodes in $\perp$ which share at least one common neighbor in $\top$. The projection of the top nodes can be defined similarly, but for consistency in what follows we will only consider the bottom node projection. Since every top node with a degree d creates a clique in the $\perp$ projection with $d(d-1)/2$ links (analogously for the bottom nodes in the $\top$ projection), the process of projecting the bigraph can result in very dense projections, even in cases where the bigraph itself is not very dense (Latapy, Magnien, and Vecchio 2008). Guillaume and Latapy (Guillaume and Latapy 2006) have looked at the projections of random bipartite graphs and observed that the projections have a low-average distance between the nodes (”small world effect”), with a diameter in the order of $\theta(log|\perp|)$ (or $\theta(log|\top|)$ for the top node projections). Moreover, if the set of bottom (or top) nodes follows a power-law degree distribution, the projection also has a power law distribution with the same exponent. The authors also observed high clustering coefficient in the projections, which suggests that this property can be seen as a consequence of the projecting step, rather than a property of the bipartite network under study.

As discussed in Chapter 1, the focus of this work is on the task of classification. More specifically, classification within bigraph data (*node classification*), where nodes for which the class is known are related to nodes for which the class must be estimated (Macskassy and Provost 2007). As an example of node classification, we consider a bigraph of users and locations, where the users are connected to the locations they have visited (e.g. logged into a WiFi IP address (Provost, Martens, and Murray 2012) or checked in using a social network app (Cho, Myers, and Leskovec 2011)). The target variable to predict is whether a user would demonstrate brand affinity actions, like visiting a brand loyalty club page or a purchase page, in order to target mobile ads. Based on the brand interest of other users visiting the same locations we can infer the (likelihood of the) class of the unknown user (Provost, Martens, and Murray 2012). Note that node classification is different from collaborative filtering in recommender systems (Koren, Bell, and Volinsky 2009), which is essentially a link prediction problem. For our bigraph of users visiting location, a collaborative filtering task would be to predict other locations that a user would be interested in visiting next. Node classification on the other hand, uses the links between the nodes (users and locations) to predict a certain *feature* of a node (the user's brand interest) and not the presence of a link between the nodes. The task of node classification within bigraph data has mainly been formulated in prior studies as a standard classification problem with high-dimensional, sparse features (Goel, Hofman, and Sirer 2012; J. Hu et al. 2007; Kosinski, Stillwell, and Graepel 2013; Raeder et al. 2012; I. Weber, Garimella, and Borra 2013). As we further discuss in Chapter 3, this often presents a challenge for many traditional techniques, as they do not take into an account the sparsity of the data. On the other hand, in this work we examine the node classification task from an alternative, network-based perspective.

Part II

CLASSIFICATION WITHIN BIPARTITE GRAPHS (BIGRAPHS)

3

Classification within bigraphs through projection

Many real-world large datasets correspond to bipartite graph data settings; think for example of users rating movies or people visiting locations. Although some work exists over such bigraphs, no general network-oriented methodology has been proposed yet to perform node classification. In this chapter, we propose a three-stage classification framework that effectively deals with the typical very large size of such datasets. First, a weighting of the top nodes is defined. Secondly, the bigraph is projected into a unipartite (homogenous) graph among the bottom nodes, where the weights of the edges are a function of the weights of the top nodes in the bigraph. Finally, relational learners/classifiers are applied to the resulting weighted unigraph. This general framework allows us to explore the design space, by applying different choices for the three stages, introducing new alternatives and mixing-and-matching to create new techniques. We present an empirical study of the predictive and run-time performances for different combinations of functions in the three stages over a large collection of bipartite datasets. There are clear differences in predictive performance with different design choices. Based on these results, we propose several specific combinations that show good accuracy and also allow for easy and fast scaling to big datasets. A comparison with a linear SVM method on the adjacency matrix of the bigraph shows the superiority of the network-oriented approach.

3.1 Introduction

As discussed in Chapter 2, many relational, behavioral and transactional datasets can be modelled as bipartite graphs (bigraphs). Up to now, the analysis of such bigraph data has been mainly limited to measuring descriptive statistics, link prediction for recommender systems, and clustering (see Section 3.5). In this work, we take a different approach and focus on the task of *node classification* within bigraphs (Macskassy and Provost 2007), by proposing a general network-based methodology. Most of the previous studies that have looked at node classification for this type of data simply formulate it as a standard classification problem with massive, sparse feature data (for instance predicting personality traits from datasets of Facebook users liking pages (Kosinski, Stillwell, and Graepel 2013), predicting demographic attributes (Goel, Hofman, and Sirer 2012; J. Hu et al. 2007) and brand interest (Raeder et al. 2012) from people's browsing history, predicting political views from history of videos watched on YouTube (I. Weber, Garimella, and Borra 2013) and etc.). In this chapter, we examine an alternative, network-based formulation.

Generally, there exist two main approaches to analyse bigraphs with the aim of obtaining summary metrics and graphs. The first one is using techniques and metrics which are specially designed for the bipartite graphs (Latapy, Magnien, and Vecchio 2008). This direct approach takes into account the bipartite nature of this particular type of graph, but unfortunately there are only a few techniques that can be applied directly on the bigraph. Therefore a second, indirect approach is often used, which is also the basis of the methodology that we propose. Let us separate the two subsets of nodes in the bigraph into a set of *top* nodes and a set of *bottom* nodes. Choosing the nodes to focus on as the bottom nodes, a bigraph can be analyzed by transforming it to a homogeneous unigraph of the bottom nodes, called a projection, where nodes are linked if they share a common top node (see Figure 2.2, left) (Latapy, Magnien, and Vecchio 2008). This projection approach allows the application of existing network analysis techniques for unigraphs to the bipartite case. It is very convenient for the problem of node classification, as numerous relational classifiers for network data exist for homogeneous graphs. A key to operating on massive bigraph data is that many of the relational classifiers make a first-order Markov assumption on the network, meaning that they will only consider the neighboring nodes.[1] As we will discuss below, the projection should be created in such a way that it can preserve as much information as possible from the original bipartite graph, for example by creating and employing link weights.

If we consider the examples of bigraphs listed before, we can see that many of them involve relationships among people and most of the relationships are based on two types of nodes: persons and so called "focal points" of social interaction or *foci* (Easley

[1] Technically, if there are neighboring nodes for whom the value of the random variable being used for prediction is unknown, relational classifiers would have to perform collective inference (Macskassy and Provost 2007). Except where explicitly discussed, we will consider the network in question to be the training data, for which all values for the target variable are known.

and Kleinberg 2010). These foci can be any type of social (e.g. events, board meetings, online forums, locations people visit etc.) or physical entities (e.g. people interested in the same books or movies, consumers making payments to the same merchants, authors collaborating on scientific papers). By visiting the same locations or being involved in the same social activities, the persons get the opportunity to meet each other and by that create a link in the projected network. In many situations, people tend to become friends with people with whom they share similar interests or characteristics—in our context people with whom they share the same foci. This is one basis for "social selection" (Easley and Kleinberg 2010). On the other hand, sharing the same foci can also be a consequence of social influence (Easley and Kleinberg 2010), where friends can influence their friends' choices of foci. Selection and social influence are the theoretical principles that explain why ties among similar people are preferentially formed (Easley and Kleinberg 2010). This results in social networks where people tend to be similar to their friends, also known as network assortativity, which is closely related to homophily (Easley and Kleinberg 2010; McPherson, Smith-Lovin, and Cook 2001).[2] We look beyond actual social networks and consider bigraphs of transactional and behavioral data in general. Our premise is that people that are similar in one domain (e.g. preferences, behavior), would act similarly in other domains as well. This concept of cross-domain similarity has been studied in prior literature as well (Martens and Provost 2011; Provost, Dalessandro, et al. 2009; Provost, Martens, and Murray 2012). For example, users that have similar preferences for some locations like specific bars or restaurants are likely also similar in age, social status and other features. These people do not need to know each other to be related, so the resulting projection can be considered as a pseudo-social network (Martens and Provost 2011). The concept of similarity is not limited to behavioral data that involve people. It can be expanded to bigraph data with any arbitrary nodes, like bigraphs of proteins connected if they are active in the same metabolic processes (Guillaume and Latapy 2006), animal species related through the plants they visit in search for food resources (Padrón, Nogales, and Traveset 2011), etc.

3.1.1 *Alternative classification techniques*

Seeing that typical bigraph datasets are very large transactional datasets, our proposed method is designed to scale up easily to millions and even billions of nodes. As an alternative to the graph approach for node prediction, one could also represent the data by the corresponding adjacency matrix, with as many rows as there are bottom nodes and as many columns as there are top nodes. Clearly this will be a very sparse matrix as most elements in this matrix will be zero. Why would our

[2] Indeed, although homophily originally corresponded to a principle of social selection, in contemporary usage it often simply refers to assortativity in social networks.

projection approach work better than simply applying classification techniques, such as support vector machines (SVM), to this dataset? [3]

Let us consider a huge transactional dataset, where for each person on the planet we keep all the locations that he/she visited over the last month with a target variable brand interest (hence using location data to target mobile ads). With a moderately fine-grained specification of location, this would result in a dataset of size 7 billion x 100 billion. Now for each individual person we want to predict the potential brand interest. The network approach would consider only the neighboring nodes in the bigraph, specifically the (for example 10 or 100) locations this person has visited, and consider all other training individuals that also visited those locations (viz., have a one in those columns). This could immediately reduce the problem of considering a few billion to only hundreds or thousands of training points. For those, the strength of the link in the projected unigraph is computed and a relational classifier is applied (details on this follow in the next sections). The *k*NN approach with typical Euclidean norm would require us to calculate the distance between the location profile of this person and every other person in the world. Even persons that visited none of the locations of the test person can have a different distance to the test person, depending on the other locations visited. Clearly, this does not scale. We will revisit creating specific, sparse distance functions below. An SVM would need to build a predictive model on the huge 7 billion x 100 billion dataset, which will not be feasible, requiring sampling and dimensionality reduction (and dimensionality reduction techniques such as singular value decomposition also scale badly to these settings) (Martens and Provost 2011). Once more, scalability issues impede the easy application of this alternative. In our empirical work, we further discuss this scalability requirement and include SVMs for the smaller datasets as a benchmark.

Let us discuss in more detail the question of how *k*NN, and other types of nearest neighbor techniques differ from our proposal. When using a nearest neighbor technique, there is a need for searching the most similar nodes. In other words, a similarity function has to be calculated for each test node over the whole training set, which in general is not scalable to high dimensions. Most common approaches in literature to this problem include nearest neighbor techniques that either reduce the training set (Devroye 1996), look for the approximate neighbors (Arya et al. 1998) or use indexing structures based on space partitioning to represent the data (Devroye 1996). For the latter, Weber et al. (R. Weber, Schek, and Blott 1998) have shown that for large enough dimensions, the performance of such methods degrades to a basic linear search. On the other hand, this need for a similarity search is eliminated when using our approach. We take advantage of the network structure, which (i)

[3] Although it is a natural, and interesting question, whether the projection approach does better than a particular traditional classification technique is not the main point. Rather, we generalize what researchers and practitioners already are doing with this sort of data, and thereby provide a family of methods with many more design options than have been considered previously. Some combinations of options may perform very well for a particular data set. The framework facilitates a systematic exploration.

provides a natural "index" to the node's neighbors, and (ii) focuses on similarity functions that consider only shared neighbors, thereby allowing for faster processing especially in a sparsely connected bigraph (as we often encounter, for example from data on human choice behavior (Junqué de Fortuny, Martens, and Provost 2013)). The main reason why nearest neighbor techniques are not suitable for these kind of datasets is the lack of scalability for traditional metrics such as Euclidean distance. One could also choose or create distance metrics that explicitly take into account the sparseness of the data, where the distance between two instances is zero if there are no columns with non-zero elements for both instances; this essentially would correspond to the bipartite network projection method we propose, where nodes are linked if they share a top node. However, several design questions remain, such as what particular metric to use when the distance is non-zero; we propose a set of possible metrics (Table 3.2).

When such a sparsity-oriented distance metric is combined with a weighted nearest-neighbor classifier (Devroye 1996), which takes into account the similarity to all nodes and in the combining function weights their contribution per class by their similarities, we derive one instance of the projection method with particular choices for the components: a particular weight assignment ($s_k = 1$), the aggregation function corresponding to the chosen distance metric (with the addition of distance equal to zero if there are no columns with non-zero elements for both instances), and wvRN (Macskassy and Provost 2007) as the relational classifier. So, the three-stage projection approach we propose can be instantiated to be a specific (non-traditional) instance of a nearest-neighbor classifier. Whether these are the best design choices for a particular problem requires empirical examination, and one of the advantages of a flexible framework is that different design choices can be compared easily and on equal footing.

To the best of our knowledge, this chapter presents a first, general study of node classification within bigraphs by transforming the bigraph into a unigraph projection. The main contributions are summarised as follows:

1. It surveys work on analysis in bipartite data via projection. There is a non-trivial amount of work, and it previously has not been collected and analysed as a specific area of study.
2. It provides a general framework for doing classification in bipartite data via projection, which is informed by the survey of prior work. It also generalizes the prior literature in an important way: by dividing the process into three stages, we can explore the design space more systematically than prior work has done. In particular, we can look specifically at the different choices for the three stages, introduce new alternatives, and mix-and-match to create new techniques.
3. It presents a large benchmark collection of bipartite classification datasets. To our knowledge, previously no one has assembled a benchmark collection

of such datasets. With such a collection we can examine the design choices empirically.

4. It provides an empirical study over the benchmark collection, examining the generalization performance and run-time performance of different methods for bipartite classification via projection. There are clear differences in predictive performance with different design choices. The best performing method is a new combination of existing techniques, and generally new combinations revealed by the 3-stage framework often are among the best performing methods.

5. It introduces a fast, comprehensible technique, called the SW-transformation, that calculates the label scores directly on the bigraph. This method allows easy scaling to big datasets of up to millions of nodes and it is very convenient for most of the today's big datasets that are very sparse, with nodes being connected to only few other nodes in the projection.

The rest of the chapter is structured as follows. We first present a range of functions for each of the framework stages (Section 3.2) and describe the datasets used in the study (Section 3.3). Further, we analyse our findings (Section 3.4), summarise the related literature (Section 3.5) and finally conclude the chapter (Section 3.6).

3.2 Methods

In order to make use of the existing relational classifiers, we can transform a bigraph into a homogeneous unigraph by using the previously discussed projection approach (Chapter 2). Projecting the bigraph gives the advantage of using powerful methods for unipartite graph analysis, but it is also an irreversible process that results in loss of information. For instance, in the projections of Figure 2.2 we lose information associated to the opposite node set, like the degree distributions, the number of shared nodes and their identity, etc. By intelligently assigning weights to the edges in the projection graph, we can incorporate information about the top nodes and better reflect the underlying structure of the bigraph. In light of this, we propose a general three-step framework for projecting and classifying bigraphs aimed at dealing flexibly with the incorporation of the appropriate information for node classification:

1. First we calculate a weight for each of the top nodes in the bigraph. This weight represents the importance of the top node and the distinctiveness it has for the target variable. All the top node weights are a function of the node degree and thus retain information about the degree distributions in the projections.

2. Next, we determine the weight of the edges in the projection by combining the weights of the shared top nodes between the bottom nodes. This additionally

includes information about the number of shared nodes in the projection's weights.

3. Finally, we use relational classifiers on the weighted unigraphs in order to predict the values for the target variables. The relational classifiers use only the graph structure to make predictions, which eliminates the need for local information about the top nodes.

We continue this section by presenting a list of specific functions for each of the framework stages and explaining the rationale behind the choices. By all means, the list is not exhaustive and can be extended with other functions as well.

3.2.1 *Determining importance of top nodes*

A set of functions for calculating the weights of the top nodes s_k, are listed in Table 3.1 and visualised in Figure 3.1. Clearly, the simplest weighting scheme would be to assign equal importance, $s_k = 1$, to all the top nodes. Although this is an easy and basic method to use, it does not make any distinction between the top nodes. Other, more complex weighting methods can be proposed based on some property of the top node k, like the number of connections (degree) d_k. One such a method is the inverse degree, referred to as "linear" by Gupte and Eliassi-Rad (Gupte and Eliassi-Rad 2012). Another one is the inverse degree frequency (Martens and Provost 2011), that is an analogy to a commonly used measure in information retrieval called Inverse Document Frequency (IDF) (Jones 1972) and is closely related to measures of entropy (Provost and Fawcett 2013). With IDF, very common terms that occur in many documents are assigned lower weights since they are less likely to be good discriminators. The inverse frequency defines the weight of a top node as a logarithmic function of the ratio between the total number of bottom nodes $n_\perp$ and the number of bottom nodes that are connected to that particular top node d_k. In the context of, for example, the users-movies network, the movies connecting fewer users provide more information for the target variable than those linking many. Users rating films noirs are more likely to have preferences in common than users rating a current blockbuster. An alternative method for weighting the top nodes is the hyperbolic tangent function. As an input to the function, we use the inverse degree of the node, based on the intuition that lower-degree nodes tend to provide higher discriminability. To our knowledge, this weighting method has not been used in prior literature and this is the first study that experiments with it. A different approach to determine the importance of the top nodes is the use of the delta function as defined in Allali et al. (Allali, Magnien, and Latapy 2011). This function takes into account that each top node has influence on the similarity between all pairs of bottom nodes which are connected to it. Therefore, a top node with a degree d, has an impact over $\frac{d(d-1)}{2}$ pairs of bottom nodes of 1/number-of-pairs. The Adamic-Adar measure (Adamic and Adar 2003) can be decomposed into

Table 3.1.: Overview of the functions for determining top nodes weight.

Top node weight function	Formula
Simple weight assignment	$s_k = 1$
Inverse degree	$s_k = \frac{1}{d_k}$
Inverse frequency	$s_k = log_{10}(\frac{n_\perp}{d_k})$
Hyperbolic tangent	$s_k = tanh(\frac{1}{(d_k)})$
Adamic and adar	$s_k = \frac{1}{log_{10}(d_k)}$
Delta	$s_k = \frac{2}{d_k(d_k-1)}$
Beta distribution	$s_k = Beta(\alpha, \beta, (\frac{max(d_k)-d_k}{max(d_k)-min(d_k)}))$
Likelihood ratio	$s_k = \frac{d_k^c}{d_k}$

a combination of the aggregation function sum of shared nodes, discussed in the next section and the associated Adamic-Adar top node function (Table 3.1).

As one can observe from Figure 3.1, all the functions discussed so far with the exception of the simple $s_k = 1$ assignment, follow the intuition that a top node with fewer connections creates stronger ties between the connected bottom nodes (Gupte and Eliassi-Rad 2012). In contrast, one might argue that the top nodes with very few edges are nothing more than noise in the data and hence should not receive a high weight. Inaccuracies in data collecting or the way the data are sampled could lead to a top node having a misleadingly high weight. A more flexible weighting scheme could automatically fit a function to choose an appropriate trade-off between specificity and noise tolerance (Martens and Provost 2011). To this end, we employ the beta distribution density function, defined by Equation 3.1, over the interval $x \in [0, 1]$, where x is the normalized top node degree (Equation 3.2). In Equation 3.1, α and β are parameters of the density function ($\alpha > 0; \beta > 0$) that define the shape of the density curve (for illustration see Figure 3.1). The beta distribution is commonly used in Bayesian analysis as a prior distribution for binominal proportions (Forbes et al. 2011). For our purpose, the beta function provides a method for tuning the "rarity" weight to fit each dataset individually. This is done by applying a grid search to find the optimal α and β parameters for the specific dataset that provide the best predictive performance (e.g., the area under the ROC curve) on a held-out validation set.

$$Beta(\alpha, \beta, x) = \begin{cases} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\cdot\Gamma(\beta)} \cdot x^{\alpha-1} \cdot (1-x)^{\beta-1}, & \text{if } 0 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

$$\Gamma(n) = (n-1)! \tag{3.1}$$

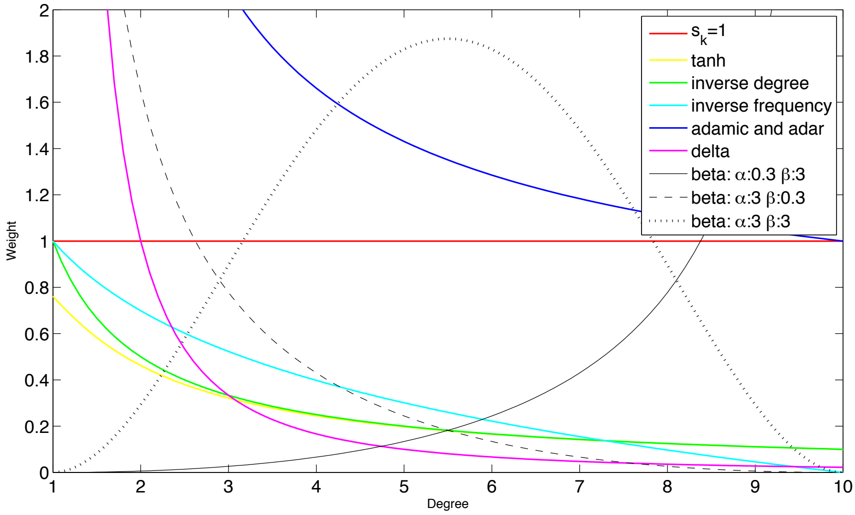


Figure 3.1.: Functions for determining the top nodes weight. Most of the functions follow the intuition that the lower-degree nodes are more discriminable for the target variable and therefore assign them higher weights.

$$x = \frac{d_k - min(d_k)}{max(d_k) - min(d_k)} \tag{3.2}$$

The likelihood ratio function, finally, takes a different approach to weighting the top nodes. It introduces supervised weighting in the projection, by taking into account how the top nodes are connected to the different classes, rather than just how they are connected in general (Martens and Provost 2011). The weight of a top node presents a ratio between the number of connected bottom nodes with positive class d_k^c and the total degree of the top node d_k.

3.2.2 *Determining the link weights in the projection*

Once we have determined the weights s_k of the top nodes, we continue with the second stage of the framework which calculates the link weights w_{ij} between the bottom nodes i and j in the unipartite projection. In Table 3.2, we introduce several methods for calculating w_{ij} as an aggregation of the weights from the shared top nodes s_k. The most straightforward way to combine the common top nodes is to simply sum their weights (Adamic and Adar 2003; Allali, Magnien, and Latapy 2011; Gupte and Eliassi-Rad 2012; Macskassy and Provost 2007; Martens and Provost 2011; M. E. Newman 2001b; Provost, Martens, and Murray 2012). Another approach is to select the maximum weight of the shared top nodes as a weight for the projection

Table 3.2.: Overview of the aggregation functions [4].

Aggregation function	Formula
Sum of shared nodes	$w_{ij} = \sum_{k \in N_b(i) \cap N_b(j)} s_k$
Max of shared nodes	$w_{ij} = \max_{k \in N_b(i) \cap N_b(j)} s_k$
Jaccard similarity	$w_{ij} = \frac{\sum_{k \in N_b(i) \cap N_b(j)} s_k}{\sum_{k \in N_b(i) \cup N_b(j)} s_k}$
Cosine similarity	$w_{ij} = \frac{\sum_{k \in N_b(i) \cap N_b(j)} s_k^2}{\sqrt{\sum_{k \in N_b(i)} s_k^2} \cdot \sqrt{\sum_{k \in N_b(j)} s_k^2}}$
Zero-one	$w_{ij} = \begin{cases} 1 & \text{if} \sum_{k \in N_b(i) \cap N_b(j)} s_k > 0 \\ 0 & \text{if } \sum_{k \in N_b(i) \cap N_b(j)} s_k = 0 \end{cases}$

edges (Gupte and Eliassi-Rad 2012). We can also use an extended, weighted version of the Jaccard index, that is defined as the sum of the weights of the top nodes that are shared by both the bottom nodes, divided by the sum of the weights of the top nodes that are connected to at least one of the bottom nodes (Allali, Magnien, and Latapy 2011; Gupte and Eliassi-Rad 2012; Provost, Martens, and Murray 2012). A problem can arise with the Jaccard index in the case when one of the bottom nodes is connected to many top nodes and the other node is connected to only a few. In that case, even when all the neighbors of one of the nodes are also neighbors of the other node, the similarity will be low. Another option for aggregating the top node weights is by employing the cosine similarity function, which calculates the similarity between pairs of vectors as the cosine value of the angle between them (Provost, Martens, and Murray 2012). Using this measure, the similarity between two bottom nodes will be the highest and equal to one when they share exactly the same top nodes and equal to zero when they don't have any neighbors in common. Finally, a very simple weighting measure assigns a value of 0 or 1 to the links in the projection, depending on whether the bottom nodes have at least one shared top node or not. This corresponds to an unweighted version of the projection graph, so it loses all the information related to the strength of the bonds between pairs of bottom nodes.

3.2.3 *Relational classifiers*

The third step of the framework for node classification within bigraphs is to use a relational (network) classifier over the unigraph projection. We consider methods for within-network classification in univariate networks, as defined in (Macskassy and Provost 2007). Specifically, in a graph where nodes with known class labels are connected to nodes with unknown class labels, relational classifiers make use of the graph structure to estimate the unknown labels. Unlike traditional non-relational models, which make use only of the local information about a node, the relational classifiers use the information about the target variable of the related nodes (their

[4] N_b refers to the neighbouring nodes in the bigraph, while N_u refers to the neighbouring nodes in the unigraph.

labels or predictions thereof) (Macskassy and Provost 2007). Macskassy and Provost compared several relational classifiers using the software package NetKit and based on their analysis, we will consider the following three relational classifiers which dominated in the study [5].

The *weighted-vote Relational Neighbor* (wvRN) classifier (Macskassy and Provost 2003) is a straightforward classifier that uses the known class labels of the related nodes (or predictions thereof) to make a probability estimation (score) of the node's own class label (see Equation 3.3). It is based on the assumption of assortativity, that the related nodes in the graph are likely to be of the same class. The classifier calculates the node's score $P(l_i = c|N(i))$ as a weighted average of the neighbour's scores.

$$P(l_i = c|N_u(i)) = \frac{1}{Z} \sum_{j \in N_u(i)} w_{ij} \cdot P(l_j = c) \tag{3.3}$$

where Z is the normalizing factor and is equal to the sum of the weights of all adjacent links ($\sum_{j \in N(i)} w_{ij}$).

The second relational classifier used in this chapter is the *class-distribution Relational Neighbor* (cdRN) classifier (Macskassy and Provost 2007). Unlike the previous classifier, it takes into account the class distribution linkages of the whole training set, and not only the immediate neighborhood, through class-specific "reference vectors". First, a class vector $CV(i)$ is created for each node as a sum of the links' weights to other nodes with each known class (l_j) (Equation 3.4). The class vectors of the training nodes are then aggregated into reference vectors for the different classes $RV(c)$ and represent an average of the $CV(i)$ for nodes known to be of class c (Equation 3.5).

$$CV(i)_c = \sum_{j \in N_u(i), l_j = c} w_{ij} \tag{3.4}$$

$$RV(c) = \frac{1}{|\perp^c|} \sum_{i \in \perp^c} CV(i) \tag{3.5}$$

where $\perp^c$ denotes the bottom nodes in the bigraph known to have label $l_i = c$.

The probability of a node i having class c can than be estimated as the normalized vector similarity between the class vector of node i ($CV(i)$) and the reference vector

[5] In this study we assume a binary target variable for the datasets. However, some of the datasets can have multiple classes, meaning that the node labels can belong to one of K possible classes. In our experimental setup, we cast these datasets to multiple bigraphs with binary labels. Alternatively, one can consider using the multivariate versions of the relational classifiers (Macskassy and Provost 2007) with these multiclass datasets. In such a case, any of the previously discussed weighting functions can be used, with the exception of the likelihood ratio function.

RV (Equation 3.6). In this chapter we employ the cosine similarity, but other functions such as L1, L2 (Bishop 2006) normalised in the range of [0,1] can also be applied.

$$P(l_i = c|N_u(i)) = sim(CV(i), RV(c)) \tag{3.6}$$

A more complex relational classifier is the *network-only Link-Based classifier* (nLB) (Lu and Getoor 2003). This classifier builds a class vector $CV(i)$ for every training node i in the network, that contains scores for each label class c. Since we only consider binary bigraphs, the class vector for a training node is a vector with two elements, that are the scores for both classes c_0 and c_1. The scores are calculated in the same way as for the wvRN classifier, also known as the count model in the study of Lu and Getoor (Lu and Getoor 2003) (Equation 3.7). In the next step, the nLB classifier builds a logistic regression model based on these class vectors (Equation 3.8).

$$CV(i)_c = \frac{\sum_{j \in N_u(i)} w_{ij} \cdot P(l_j = c)}{\sum_{j \in N_u(i)} w_{ij}} \tag{3.7}$$

$$P(l_i = c|N_u(i)) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot CV(i)_{c_0} + \beta_2 \cdot CV(i)_{c_1})}} \tag{3.8}$$

3.2.4 *Decomposition of metrics*

So far, we considered a wide range of functions for creating the weights of the projection. To the best of our knowledge, we apply all the methods that were previously used in the literature for defining the link weights in bigraph projections and that can be decomposed within our framework. In Table 3.3, we present a summary of the measures used in prior literature, divided in the three stages: top nodes weighting function, aggregation function and relational classifier. The formula for decomposition is given in Equation 3.10, where g represents the aggregation function and f the top node weighting function. As an example, the Adamic-Adar coefficient (Adamic and Adar 2003) (see Equation 3.9), can be fragmented into the Adamic-Adar top node weighting function (Table 3.1) and the sum of shared nodes aggregation function (Table 3.2). This clearly creates opportunities for combining the existing weighting functions in new ways, resulting in completely new techniques. Note that some of the combinations from prior literature do not include a relational classifier, since these studies use the unigraph projection for other tasks rather than classification, like link prediction (Allali, Magnien, and Latapy 2011; Gupte and Eliassi-Rad 2012) and measuring descriptive statistics (Guillaume and Latapy 2006; M. E. Newman 2001a,b). Moreover, some studies consider the unweighed unigraph projections (Guillaume and Latapy 2006; M. E. Newman 2001a). Since this is independent of the top node weights, we simply note "any" in the first column of Table 3.3.

$$w_{ij} = \sum_{k \in N_b(i) \cap N_b(j)} \frac{1}{\log_{10}(d_k)} \tag{3.9}$$

$$w_{ij} = g(s_{k1}, s_{k2}, ..., s_{kn}) = g(f(d_{k1}), f(d_{k2}), ..., f(d_{kn})) \tag{3.10}$$

3.2.5 *Scalability*

In the introduction we discussed how bigraphs are a natural and efficient representation for sparse feature data, and indeed the sparse representation commonly used by machine learning methods is in fact a (possibly weighted) bigraph adjacency list. The projection itself can reduce the data size, but only sometimes (e.g., when the connections are sparse and the number of top nodes is much larger than the number of bottom nodes). This notwithstanding, the scalability of the algorithms nonetheless deserves special attention since bigraph data and their corresponding unigraph projections often are very large; methods are needed that can deal with massive data. In this section we propose several techniques that enable the algorithms to scale up to very large datasets and/or improve their run-time performance.

BATCH PROCESSING Large datasets that can not be processed in memory, can be divided into smaller, processable subsets called batches. In this chapter, *batch processing* means that the label scores will be calculated by processing the batches one at a time, either sequentially or in parallel, instead of processing the whole dataset at once (cf., (Provost and Kolluri 1999)). This allows for easy scaling up using parallel and distributed computing systems.

For instance, in the case of the network-only Link-Based (nLB) classifier (see Equation 3.7), batch processing means that we create a partial projection for each batch from the training dataset and then use it to calculate a part of the class vector. Later, we assemble all the partial vectors into a whole class vector that is used by the logistic regression. In a similar manner, we create part of the scores from the test batches and then aggregate them together into a final solution. One needs to be careful when choosing the size of the batches; a size that is too large will cause the CPU to thrash, wasting substantial time swapping memory blocks between RAM and disc. On the other hand, the runtime performance can also be substantially degraded by choosing a size that is too small, since each fragment introduces additional calculation overhead. In our work, all the batch sizes were determined experimentally, by testing different sizes for each dataset.

Top node weight	Aggregation function	Rel. classifier	Ref.
any	Zero-one	-	(Guillaume and Latapy 2006; M. E. Newman 2001a)
Simple weight assign. ($s_k = 1$)	Sum of shared nodes	-	(Allali, Magnien, and Latapy 2011; Gupte and Eliassi-Rad 2012)
Inverse degree	Sum of shared nodes	-	(Gupte and Eliassi-Rad 2012; M. E. Newman 2001b)
Adamic and adar	Sum of shared nodes	-	(Adamic and Adar 2003; Gupte and Eliassi-Rad 2012)
Delta	Sum of shared nodes	-	(Allali, Magnien, and Latapy 2011; Gupte and Eliassi-Rad 2012)
Simple weight assign.($s_k = 1$)	Jaccard similarity	-	(Allali, Magnien, and Latapy 2011; Gupte and Eliassi-Rad 2012)
Inverse degree	Max of shared nodes	-	(Gupte and Eliassi-Rad 2012)
Simple weight assign.($s_k = 1$)	Sum of shared nodes	wvRN	(Macskassy and Provost 2007; Provost, Martens, and Murray 2012)
Inverse frequency	Sum of shared nodes	wvRN	(Provost, Martens, and Murray 2012)
Inverse frequency, likelihood ratio	Sum of shared nodes	wvRN	(Martens and Provost 2011)
Beta distribution, likelihood ratio	Sum of shared nodes	wvRN	(Martens and Provost 2011)
Simple weight assign. ($s_k = 1$)	Jaccard similarity	wvRN	(Provost, Martens, and Murray 2012)
Inverse frequency	Jaccard similarity	wvRN	(Provost, Martens, and Murray 2012)
Simple weight assign. ($s_k = 1$)	Cosine similarity	wvRN	(Provost, Martens, and Murray 2012)
Inverse frequency	Cosine similarity	wvRN	(Provost, Martens, and Murray 2012)
Simple weight assign.($s_k = 1$)	Sum of shared nodes	cdRN	(Macskassy and Provost 2007)
Simple weight assign.($s_k = 1$)	Sum of shared nodes	nLB	(Macskassy and Provost 2007)

Table 3.3.: Overview of measures for defining links' weight in bigraph projections used in previous literature.

sampling Another technique that enables the network-only Link-Based (nLB) classifier to scale to larger datasets and improve the run-time performance is *sampling*. Within our experiments, we observed that building the class vectors with only a subset of the training nodes (around 100 training instances) was usually sufficient for training the classifier. Therefore, in the experimental set-up we run the nLB without sampling and with sampling (denoted as nLB100 in what follows) and compare the results.

grid search Fine tuning the parameters of the optimal beta function (Equation 3.1), can require many iterations. In order to reduce their number, we can apply a *grid search* with multiple levels [6], where every successive level performs a more fine-grained search around the optimal parameter values from the previous level. If, for instance, our search on level i with step s_1 determined that x_1 is the best parameter value, then the following level $i+1$ will look for a more optimal result x_2 (x_2 may be equal to x_1) in the range $[x_1 - s_1, x_1 + s_1]$ with a smaller step s_2 [7]. As we further discuss in Section A.1 (Appendix A), we can also perform a grid search with fewer levels that results in limited performance degradation or incorporate knowledge about the most suitable beta shapes in the search procedure.

sw transformation Finally, we introduce a fast method called the *SW - transformation*, that can be used in cases when the wvRN (Equation 3.3) is combined with an aggregation function that sums the weights of the top nodes. Hence the name SW-transformation, which is an acronym of the two methods involved: the sum of shared nodes and the wvRN. As we describe next (Equations 3.11-3.19), this specific combination can be rewritten as a fast linear model over the top nodes.

We start the transformation process by substituting the projection weights w_{ij} in the wvRN formula with the corresponding aggregation function (Equation 3.11). Since the wvRN classifier considers only the neighbouring nodes' labels, in Equation 3.12 we take into account only the bottom nodes j that have an element $a_{ij} = 1$ in the unigraph adjacency matrix (see Figure 3.2, right). The link weight w_{ij} in the projection is calculated by summing the weights of the top nodes that are shared by both nodes i and j. This means that the top nodes which are not associated with node i (that have elements $x_{ik} = 0$ in the bigraph adjacency matrix from Figure 3.2, left) can be discarded in Equation 3.14. Furthermore, since we assume a binary target variable in our study, we eliminate the neighboring nodes with label $y_j = 0$ in Equation 3.16 [8]. The result is the SW-transformation (Equation 3.20), a linear model that computes the label scores directly on the bigraph and avoids the costly step of calculating

[6] We tune the beta function using a grid search with three levels.

[7] The grid we use, searches for the optimal α and β in the range between 0.1 and 12.1 with steps of 3 in the first level. We decrease the step size in each successive level by 3 times.

[8] Note that the neighboring nodes with label $y_j = 0$ are still counted when calculating the normalization factor Z in Equation 3.11

$$\begin{array}{c|cccccc} & t_1 & t_2 & \dots & t_k & \dots & t_n \\ \hline b_1 & x_{11} & x_{12} & \dots & x_{1k} & \dots & x_{1n} \\ b_2 & x_{21} & x_{22} & \dots & x_{2k} & \dots & x_{2n} \\ & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_i & x_{i1} & x_{i2} & \dots & x_{ik} & \dots & x_{in} \\ & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_m & x_{m1} & x_{m2} & \dots & x_{mk} & \dots & x_{mn} \end{array}$$

(a) $m \times n$ matrix representation of a bigraph.

$$\begin{array}{c|cccccc} & b_1 & b_2 & \dots & b_j & \dots & b_m \\ \hline b_1 & a_{11} & a_{12} & \dots & a_{1j} & \dots & a_{1m} \\ b_2 & a_{21} & a_{22} & \dots & a_{2j} & \dots & a_{2m} \\ & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_i & a_{i1} & a_{i2} & \dots & a_{ij} & \dots & a_{im} \\ & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ b_m & a_{m1} & a_{m2} & \dots & a_{mj} & \dots & a_{mm} \end{array}$$

(b) $m \times m$ matrix representation of the projected unigraph.

Figure 3.2.: Adjacency matrices of the bigraph and the projected unigraph.

the projection unigraph. In terms of implementation, the transformation optimally considers for each test instance only the weight s_k of the neighboring top nodes (where $x_{ik} = 1$ in the bigraph adjacency matrix) multiplied by the number of training instances in that column which have a label $y_j = 1$ (the positive neighbors of the node). In this manner, we directly calculate the influence of the top node, in the form of the coefficient of the corresponding linear model. The SW-transformation yields substantially faster run times (compared to calculating the whole projection) and allows easy scaling of the method to big datasets of up to millions of nodes, as we discuss further in Section 3.4. If we consider that most of the today's big dataset are very sparse, with nodes being connected to only few other nodes in the projection, we can clearly see the usefulness and the applicability of the SW-transformation.

$$\begin{aligned}
Z \cdot P(l_i = c | N_u(i)) &= \sum_{j \in N_u(i)} w_{ij} \cdot P(l_j = c) && (3.11) \\
&= \sum_{j | a_{ij} \neq 0} w_{ij} \cdot y_j && (3.12) \\
&= \sum_{j | a_{ij} \neq 0} \left(\left(\sum_{k | x_{ik} \neq 0} s_k \right) \cdot y_j + \left(\sum_{k | x_{ik} = 0} s_k \right) \cdot y_j \right) && (3.13) \\
&= \sum_{j | a_{ij} \neq 0} \left(\left(\sum_{k | x_{ik} \neq 0} s_k \right) \cdot y_j + 0 \right) && (3.14) \\
&= \sum_{\substack{j | a_{ij} \neq 0 \\ y_j = 0}} \left(\sum_{k | x_{ik} \neq 0} s_k \right) \cdot y_j + \sum_{\substack{j | a_{ij} \neq 0 \\ y_j = 1}} \left(\sum_{k | x_{ik} \neq 0} s_k \right) \cdot y_j && (3.15)
\end{aligned}$$

$$= 0 + \sum_{\substack{j|a_{ij}\neq 0 \\ y_j=1}} \left(\sum_{k|x_{ik}\neq 0} s_k \right) \cdot 1 \tag{3.16}$$

$$= \sum_{k|x_{ik}\neq 0} \left(s_k \sum_{\substack{j|a_{ij}\neq 0 \\ y_j=1}} 1 \right) \tag{3.17}$$

$$= \sum_{k|x_{ik}\neq 0} \left(s_k \sum_{\substack{j|x_{ik}\neq 0, x_{jk}\neq 0 \\ y_j=1}} 1 \right) \tag{3.18}$$

$$= \sum_{k|x_{ik}\neq 0} s_k \cdot ns_k \tag{3.19}$$

where $ns_k = |x_{jk} = 1$ and $y_j = 1|$ and s_k is a weight of a top node in Equation 3.19. In a similar manner, the normalization factor Z can be transformed into $Z = \sum_{k|x_{ik}\neq 0} s_k \cdot Zs_k$, where $Zs_k = |x_{jk} = 1|$. Finally,

$$P(l_i = c|N_u(i)) = \frac{\sum_{k|x_{ik}\neq 0} s_k \cdot ns_k}{\sum_{k|x_{ik}\neq 0} s_k \cdot Zs_k} \tag{3.20}$$

3.3 Data and Experimental Setup

For this study we collected bipartite datasets from various sources: the Koblenz Network Collection (KONECT) [9], the MIT Reality Mining Project [10], the social networks collection of The Max Plank Institute for Software Systems [11] and more. We selected all datasets where a bipartite structure is clearly present and a target variable is available to predict. Note that in some cases we discard a dataset because the class variable is related to the links in the bigraph. For instance, predicting the number of books that the users have read from a bigraph of users rating books is clearly not suitable. The datasets are summarized in Table A.2 (Appendix A) and to our knowledge comprise the first large collection of benchmark datasets for node classification over bigraphs.

The *MovieLens* dataset contains information about movie ratings from users of the MovieLens website, collected from September 1997 through April 1998 [12]. The

[9] http://konect.uni-koblenz.de
[10] http://realitycommons.media.mit.edu
[11] http://socialnetworks.mpi-sws.org
[12] http://www.grouplens.org

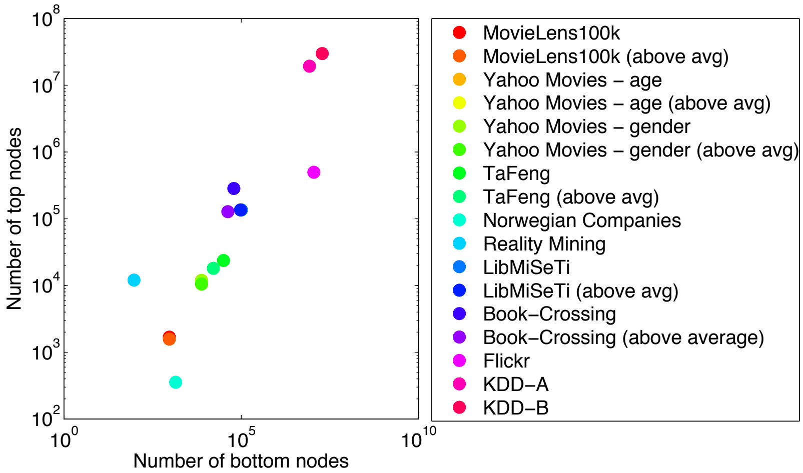


Figure 3.3.: Size of the datasets under study.

bigraph is defined between users and movies, where links are present if a user rated a movie. We focus on the task of predicting the genre of the movie, as well as the gender and the age of the user. In the first case, the movies are considered as bottom nodes and the users as top nodes, for the latter it is vice versa. For multiclass problems (as genre), we use a one-versus-all formulation and as such we define as many additional datasets as there are classes (19 in this case) [13]. The *Yahoo Movies* dataset [14] has a similar setting, where again we predict the gender and the age of users who rated movies. Likewise, *Book-Crossing* contains book ratings from the web site Bookcrossing.com (Ziegler et al. 2005) and our aim is to predict the age of the readers. The dataset collected by Opsahl and Seierstad (Seierstad and Opsahl 2011) is used for defining a bigraph between *Norwegian companies* and their board members, and the target variable is the gender of the board members. Furthermore, we use the information about mobile phone usage collected by the MIT Human Dynamics Lab (*Reality Mining* project) (Eagle and Pentland 2006) to define a bigraph of users connected to the locations (cell towers) they visited. The target variable is the affiliation of the user, being student, laboratory personnel, professor, etc. Another bigraph is defined from the *LibimSeTi* (Brozovsky and Petricek 2007) dataset, that contains data about profile ratings from users of the Czech social network LibimSeTi.cz. The prediction task in this case is the gender of the users. *Ta-Feng* is a dataset of supermarket transactions, where we predict the age of the customers, based on the products they bought (H.-S. Huang et al. 2005). Moreover, we used a dataset from the Max Plank Institute for Software Systems that contains data about several million *Flickr* pictures, to create a bigraph of pictures and the users that mark them as favorite. The target variable we predict is the number of comments on

[13] For this chapter, we only consider binary classification, where multiclass problems are cast to several one-versus-all binary classification problems.

[14] http://webscope.sandbox.yahoo.com/

the pictures. The largest datasets used in this study are from the KDD Cup 2010, where the participants were asked to predict student performance on tests. The winner of the Cup, the National Taiwan University, expanded the original dataset by converting the categorical features into sets of binary features (Yu et al. 2010) and this version of the data can be downloaded from the LibSVM website [15]. In addition to the previously described datasets, we also created new bigraphs for the rating data, where a connection exists between the nodes only if the rating was positive (defined as higher than the average rating). We annotate this type of bigraphs as “above average” in Figure 3.3 and Table A.2 in Appendix A.

Figure 3.3 gives an overview of the number of nodes present in the bigraphs under study. As shown in the plot, the size of the bigraphs differ, with some datasets having fewer than 100 bottom nodes (Reality Mining, number of people involved) and a few hundred top nodes (Norwegian companies, number of companies involved) and others with up to a few million top and bottom nodes (KDDa and KDDb datasets). In Appendix A (Figures A.1-A.4), we also examine the distributions of the probability $P(k)$ that a node has a degree k (also known as degree distributions) for the bottom and top nodes of each dataset. As one can observe, most of the datasets show a heavy-tailed degree distribution, resembling the typical power-law with different exponents. In such distributions, many nodes in the bigraph are connected only to few nodes from the opposite set, but a non-negligible number of nodes are connected to very many other nodes. The top nodes of the LibMiSeTi dataset do not follow the power law: most of the profiles in the social network get an average number of rankings from the other users, similarly for the Yahoo Movies dataset. The bottom nodes of the KDDa dataset are an exception as well, which might be due to the fact that it is an artificially created dataset.

3.4 Results

In this section, we present the predictive performance results for the techniques used by the three-stage framework (Table 3.4 and Table A.1 from Appendix A). In our empirical assessment, we examine the performance for each combination of top node weighting scheme, aggregation function and relational classifier. This leads to a total of $8 \times 5 \times 4$ combinations, that are assessed on the 58 datasets (including the casted multiclass datasets). As a benchmark technique we employ an SVM with a linear kernel on the bigraph adjacency matrices using the libLINEAR toolbox (Fan et al. 2008a). Our experimental setup consists of a 10 fold cross-validation procedure, where the reported AUC values (see Chapter 1) (Fawcett 2006) represent an average over all the folds. In each fold we use 90% of the data for training the models and 10% for assessing their predictive performance [16]. Note that each of the 10 data subsamples is used exactly once as a test set in the process. In cases where we have parameters for tuning (e.g. SVM, beta), in each run we divide the dataset into three

[15] http://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/

[16] The top node weights are calculated using solely the training data.

segments: 80% of the data is used as a training set, 10% as a held out validation set and 10% as a test set [17]. For statistical comparison of the methods, we use a Wilcoxon signed rank test (Demšar 2006) to asses the significant differences between the best performing method and the other classifiers. We emphasize the combinations that are not significantly worse from the best method (underlined) at a 5% significance level in bold and the combinations that are significantly worse at 5% but not at 1% significance level in italic. The other methods that are significantly worse at 1% significance level are shown in regular font. For the interested readers, we also provide a run-time evaluation of the methods in Section A.1 (Appendix A).

Table A.1 in Appendix A presents the predictive performance results for every combination of techniques, based on the For every dataset, we rank the performance of the techniques into a partial ranking and then combine them together into a final ranking using the Kemeny-Young optimisation (Conitzer, Davenport, and Kalagnanam 2006; Young and Levenglick 1978). The goal of the Kemeny-Young method is to find an ordering of the techniques that minimises the number of pairwise disagreements between the final ranking of the techniques and the partial rankings calculated on the individual datasets. This means that if one technique is ranked higher than another one in the final Kemeny-Young ordering, then for every dataset where the opposite is valid (the second technique outperforms the first one), the total disparity is incremented by one. This is calculated for every pair of distinct techniques and the ordering with the lowest score is selected as the final ranking. Note that when calculating the final ordering, we also consider the datasets where not all combination schemes were able to scale. More specifically, the aggregation functions Jaccard, Cosine similarity and Maximum do not scale well to datasets with high dimensions such as the Flickr dataset or larger. Also, a combination of these aggregation functions and the beta function, for which we need multiple iterations to fine tune the parameters, takes very long time to fit for datasets over 100,000 nodes. In such cases, when a dataset does not provide a ranking for one or both methods that are being compared, the dataset does not contribute to the total disparity regarding these two techniques.

From Table A.1, we can observe that the highest ranked combination that performs very well over all datasets is the hyperbolic tangent function, combined with the cosine aggregation function and the wvRN classifier. Furthermore, there are also a few alternatives that provide comparable results to this top ranked combination. If we take a closer look at the results, we can see that generally combinations that include the cosine or sum of shared nodes aggregation functions together with the hyperbolic tangent, inverse degree and occasionally the beta function provide very good and close results when combined with any of the relational classifiers. The SVM benchmark against which we compare the network projection methods, has only average performance. It is ranked on the 98th place out of 161 possible

[17] Again, each of the 10 data subsamples is used exactly once for testing and once for validation in the process.

techniques and it is significantly worse than the best method at 1% significance level. Additionally it is not able to scale up to the big KDDa and KDDb datasets.

In order to assess the quality of the functions better, we continue this section by examining the performance of the functions more carefully, by looking at each of the three stages separately. We use the ranking of the techniques from Table A.1, to create new Kemeny-Young orderings of the techniques for each stage. The partial rankings in this case represent the performance of the functions from the stage under study, ranked for every combination of techniques from the other stages.

3.4.1 *Predictive performance of the top node weight functions*

The rankings regarding the top node functions are summarized in Table 3.4, with the hyperbolic tangent and the inverse degree (both similar in shape, see Figure 3.1) providing the best performance across all domains. We should, however, be careful when interpreting these results and do not simply discard top node methods that provide poorer results over all domains. Although specific combinations that include the top node function can have very strong performances (see Table A.1), the overall rankings still get diluted by the weaker combinations (for instance, the ones containing the zero-one aggregation function). If we take a closer look at Table A.4 in Appendix A, where we list the best combinations of techniques per dataset, one can easily notice that in most cases the best performing combination contains the beta distribution as an appropriate choice. By analysing the optimal α and β coefficients for the different datasets (listed in Table A.3), we conclude that the typical shapes of the function correspond strongly to the intuition that top nodes with smaller degree are more discriminative and therefore should have higher weights. The only exception is the Flickr dataset, where the parameters define the opposite curve of the beta function. Based on the results from Table A.4, we also conclude that the supervised weighting function, likelihood ratio, exhibits very good performance for skewed datasets with a small number of positive labels [18]. What is specific about this function is that it weights only the top nodes that are connected to at least one bottom node with a positive label. This results in projections where the links to some of the neighbouring nodes with negative labels are down-weighted, since the top nodes connecting only negative bottom nodes do not contribute to the projection weights.

3.4.2 *Predictive performance of the aggregation functions*

Table 3.4 also presents an overview of the results per aggregation function, with the cosine function and the sum of shared nodes as the most suitable methods. Although both functions provide very good performances, the latter one is favourable since

[18] All the datasets for which this function performed best have only between 3.19% and 7.25% positive labels.

Table 3.4.: Kemeny - Young ranking per method on all datasets.

Kemeny Ranking	Top node func.	Aggregation func.	Relational learner
1	**tanh**	**cosine function**	**wvRN**
2	*inverse degree*	**sum of shared nodes**	**nlb**
3	*inverse frequency*	jaccard	cdRN
4	*beta distribution*	max	nlb 100
5	w=1	zero - one	
6	adamic and adar		
7	delta		
8	likelihood ratio		

it can be combined with the wvRN classifier (SW-transformation) to scale easily and fast to very large datasets. All the functions perform much better than the zero-one function, which corresponds to an unweighted version of the projection. This strongly supports the idea that adding weights to the projection reflects better the structure of the underlying bigraph and therefore results in better predictions. The Jaccard aggregation function does not perform well, as it penalizes the score if one of the nodes has a lot of links. As an example, let us consider again the bigraph of people visiting locations, with person A visiting 5 different locations; person B visiting these 5 locations and 10 more; and person C visiting the same 5 locations and 100 more. In this case, the Jaccard would penalize the AC link with a much lower score than the AB link, because of the metric's denominator which takes into account all the locations visited by at least one of the persons. This does not make sense for this setting: if we have a total of (for example) a million locations, the odds for visiting the same 5 locations by chance are very small. The max function also shows poor performance, which indicates that it is valuable to retain information for more than just one top node.

3.4.3 *Predictive performance of the relational classifiers*

In Table 3.4, we can also see the aggregated results over the relational classifiers, where the best classifier wvRN slightly outperforms the nLB. These two classifiers provide similar results in cases with relational autocorrelation over the target values in the projection. An example of positive relational autocorrelation (Jensen and Neville 2002) would be that if I like the same movies as someone else, we likely are of the same gender. Yet, the opposite can be true as well as in the case of the Norwegian companies dataset, where a man is more likely to be in a board with a female and vice verca. For this reason, the wvRN here yields a pathological average AUC (over all combination schemes) of only 0.2728. This substantially hurts the wvRN average scores, as AUCs systematically below 0.5 can be "flipped" to sometimes strong AUCs. Therefore, this result requires some extra explanation.

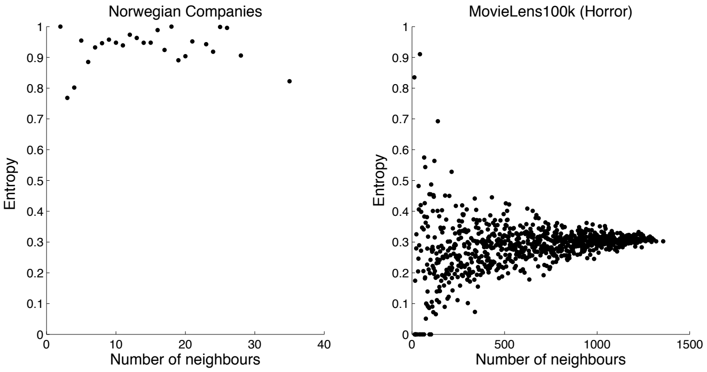


Figure 3.4.: Average entropy per number of neighbors in the projection, for the Norwegian Companies dataset (left) and the MovieLens dataset with target variable Horror genre (right).

Norway is one of the leading countries that enforces equal gender representation in companies' boards (Seierstad and Opsahl 2011), which results in the companies (top nodes) being connected to almost the same number of male and female directors (bottom nodes). In Figure 3.4, we use entropy as a measure for class imbalance (Rrnyi 1961), to calculate the heterogeneity of the target variable among the nodes' neighbours in the projection. Very high values of entropy, signify that there is nearly an equal number of neighbours from both classes, whereas low values suggest that all the adjacent nodes have the same class. The results are averaged over the nodes that have the same number of neighbours. As expected, the dataset of Norwegian companies exhibits high entropy values for all the board members (see Figure 3.4, left). In comparison, a typical dataset where the wvRN classifier performs equal or even better has much lower entropy (see Figure 3.4, right) [19]. In such cases where the class distribution of a node's neighbors is approximately 0.5, cross-validation can cause pathologies in machine-learning evaluations. Consider the following.

Most of the directors (83.6%) are members of the board of only one company and most companies (71%) have only up to 5 board members. This creates many small disconnected components in the bigraph, like the ones depicted in Figure 3.5. When the wvRN relational classifier is applied with cross-validation, it is likely that the focal node's target class will be underrepresented in the remaining neighbour nodes. For example, consider a leave-one-out-style evaluation. In case (a), a female member will be connected to only one male member in the projection, hence wvRN will predict the wrong class. In the other cases, the links' weights in the projection will be equal, leading to wvRN predicting the majority opposite class or giving a score of 0.5 when the remaining classes are balanced (denoted with question mark in the predictions under the nodes). This may possibly penalize wvRN in these cases

[19] Note that in addition to the entropy, the weights of the links also have impact on the prediction performance of wvRN.

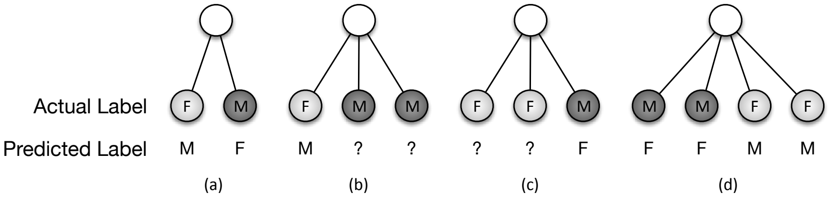


Figure 3.5.: Bigraph structures of companies (top nodes) and board members (bottom nodes). The node letter presents the actual gender of the board member and below is the predicted gender by the wvRN.

and artificially bolster the performance of the learning-based methods, or it may be exactly what we would like to happen in these cases. The learning-based classifiers are able to pick up on this: the nLB classifier provides a negative coefficient to the female class distribution for males (and again vice versa), which leads to an AUC of 0.7029 and the cdRN creates reference vectors that take into account how the training nodes are connected to the opposite class, yielding an average AUC of 0.6997. Based on the analysis, we conclude that wvRN is an appropriate choice for problems that exhibit network assortativity; however, the nLB is more powerful and can capture more complex patterns. The nLB100 is a trade-off between speed and accuracy, a much faster variant of the nLB classifier (see the run-time analysis of the methods in Figure A.10 from Section A.1 (Appendix A)), but with weaker predictive performance.

3.4.4 *SW-transformation*

The SW-transformation combines the best performing relational classifier wvRN and one of the best ranked aggregation functions, sum of shared nodes into a fast [20] linear model that scales easily to big datasets. It is the only technique in the study that scales well (or at all) to the biggest datasets KDDa with 8 million × 20 million nodes and KDDb with 19 million × 30 million nodes. An additional important aspect of the SW-transformation is the comprehensibility of the linear models it provides. A manual check of the top node coefficients (the impact they have over the target variable), can ensure whether the model makes sense [21]. Comprehensibility is needed, and even mandatory, in many domains where the decisions of the classifier need to be clearly explained and validated before the classifier can be used (Gregor and Benbasat 1999; Martens, Baesens, et al. 2007; Martens and Provost 2014b). In Table 3.5 we make an additional verification of the results, by examining the top nodes' coefficients using the combination of the beta and SW-transformation. We list the top 20 ranked instances with the highest scores/coefficient when predicting

[20] Section A.1 discusses the runtime advantages of using the SW-transformation.

[21] A study by Martens and Provost (Martens and Provost 2014a) introduces an instance-level approach for explaining documents classification, which can also be used with these type of data.

Table 3.5.: Top nodes with highest coefficient in the linear model of the SW-transformation in combination with the beta function. The higher scores indicate higher probability of being (a) male when predicting *gender* and (b) young when predicting *age* for the Yahoo Movies bigraph.

Rank	Yahoo Movies (gender)	Yahoo Movies (age)
1.	The Matrix Reloaded (2003)	Ocean's Eleven (2001)
2.	Terminator 3: Rise of the Machines (2003)	The Ring (2002)
3.	The Hulk (2003)	Scary Movie 3 (2003)
4.	X2: X-Men United (2003)	American Pie 2 (2001)
5.	Bad Boys II (2003)	American Pie (1999)
6.	The Lord of the Rings: The Two Towers (2002)	Pulp Fiction (1994)
7.	The Italian Job (2003)	The Texas Chainsaw Massacre (2003)
8.	The Matrix Revolutions (2003)	Austin Powers in Goldmember (2002)
9.	Bruce Almighty (2003)	Terminator 2 - Judgment Day (1991)
10.	28 Days Later (2003)	Gladiator (2000)
11.	Kill Bill Vol. 1 (2003)	The Lizzie McGuire Movie (2003)
12.	American Wedding (2003)	Phone Booth (2003)
13.	Freddy vs. Jason (2003)	Uptown Girls (2003)
14.	S.W.A.T. (2003)	How to Deal (2003)
15.	The Matrix (1999)	Signs (2002)
16.	The League of Extraordinary Gentlemen (2003)	Daredevil (2003)
17.	The Lord of the Rings: The Fellowship of the Ring(2001)	X-Men (2000)
18.	Terminator 2 - Judgment Day (1991)	The Matrix (1999)
19.	Seabiscuit (2003)	A Walk to Remember (2002)
20.	Star Wars (1977)	Anger Management (2003)

gender and age for the Yahoo Movies bigraph. The rankings seem indeed intuitive and enclose (i) movies that are generally targeted to a male audience (*Terminator, X-man, Kill Bill*, etc.) and (ii) movies usually preferred by younger people (*American Pie, Scary Movie, The Texas Chainsaw Massacre*, etc.)

3.4.5 *General recommendations*

We have provided an extensive empirical study of the predictive performance for a number of choices in the framework over a large collection of bipartite datasets. The results indicate that it is difficult to simply claim that a certain combination of methods performs best across all domains. Instead, based on the empirical study, we would recommend experimenting with several choices from the three stages that generally provide good results. In Figures A.12-A.19 we plot the predictive and run-time performance of all the combinations of methods on each dataset individually. The combinations of the methods we recommend (hyperbolic tangent, cosine similarity, sum of shared nodes, wvRN and nLB100) are denoted with red squares. In most cases they are among the fastest and most accurate combinations, with less than 5% AUC difference from the combination that performs best. This is not the case for several very skewed MovieLens datasets (with only 1.25%-6.5% positive labels), where we predict genres like Fantasy, Film-Noir, War or Mystery.

In such cases, as discussed before, the supervised weighting function likelihood ratio or the tunable beta function might be more appropriate choices. Furthermore, our recommendations have weak results with the Reality Mining dataset, where most of the people have visited the same places (a person on average shares all the locations he/she visited with 50% of the other people). For such datasets, where the projections are fully (or almost fully) connected, traditional classification approaches (discussed in Section 3.1.1), can be better alternatives.

3.5 Related work on bigraph data analysis

The literature regarding bigraphs has so far been focused on measuring descriptive statistics, link prediction for recommender systems and clustering. Moreover, there exist many unigraph studies that essentially use projections of bipartite data. For instance, the datasets used to create networks based on scientific collaborations (Liben-Nowell and Kleinberg 2007), co-occurrence of companies in text documents (Macskassy and Provost 2007), web page co-citation (Lu and Getoor 2003), movies linked if they share the same production company or crew (Macskassy and Provost 2003, 2007), book co-purchase (Gallagher et al. 2008), etc. in the unigraph literature are in fact bigraph projections. To our knowledge, different projection methods were not compared to maximize performance on the associated task.

There has been some initial research that explores the bigraph properties and that extends several *global network metrics* for unigraphs to the bipartite case. Centrality measures, which determine the varying importance of nodes within the graph, like betweenness, degree, closeness and eigenvector centralities (Borgatti and Everett 1997; Borgatti and Halgin 2011; Faust 1997), as well as the clustering coefficient (Latapy, Magnien, and Vecchio 2008; Lind, Gonzalez, and Herrmann 2005; Opsahl 2011; Robins and Alexander 2004) have been adapted for bigraphs. Newman (M. E. Newman 2001a,b), on the other hand, takes a different approach to measuring the network statistics of a bigraph defined between authors and papers, by applying unigraph metrics to the projection over the authors' nodes.

A second research area concerns *link prediction* in bigraphs, which aims to predict the links that will appear in the future, based on the present bigraph structure. For example, Huang et al. (Z. Huang, Li, and H. Chen 2005) use several linkage measures based on the topology of the bigraph, to predict the links that are most likely to emerge for each bottom node. Benchettara et al. (Benchettara, Kanawati, and Rouveirol 2010) describe the distinct pairs of unlinked nodes in the graph with various topological attributes and consider the pairs as positive instances if a link is created between the nodes in the training set. Subsequently, they apply supervised techniques over the pairs of nodes to predict the imminent links. A study by Allali et al. (Allali, Magnien, and Latapy 2011) considers the prediction of links that do not change the unipartite projections when added to the bigraph (also known as internal links). The internal links in the bigraph are predicted when the weights of their induced links in the projection are higher than a certain threshold. Furthermore, Kunegis et al. (Kunegis, De Luca, and Albayrak 2010) propose algebraic methods to the problem of link prediction in bigraphs and Zhou et al. (Zhou et al. 2007) consider applying methods directly on the weighted unipartite projection. Link prediction has been applied in prior literature to recommender systems for online music shops (Benchettara, Kanawati, and Rouveirol 2010), online book stores (Z. Huang, Li, and H. Chen 2005), movie recommendation (Zhou et al. 2007), etc.

Another area of study that has also been explored in the literature is bigraph *clustering*. The paper of Forunato (Fortunato 2010) provides a good overview of clustering techniques, including the ones applicable to bipartite graphs. For instance, Borgatti and Halgin (Borgatti and Halgin 2011) apply unipartite clustering techniques on the extended bigraph matrix, that contains all the nodes from the two bigraph sets in both dimensions. Blockmodeling is another popular clustering method for bigraphs (Borgatti and Halgin 2011; Doreian, Batagelj, and Ferligoj 2004). Zha et al. (Zha et al. 2001) propose a partitioning method for bigraphs that minimizes the sum of the link weights between the node pairs that are from distinct type and do not belong to the same cluster. Barber (Barber 2007) introduces a random graph model for bigraphs and extends the measure of modularity to the bipartite case, by calculating the degree to which nodes cluster into communities in respect to the null model. Sun et al. (Sun et al. 2005) employ a random walk to identify the similar nodes in the bigraph. More specifically, they calculate for every bottom node a relevance score that represents the number of times a node has been visited during the walk. The relevance scores are used to detect anomalous top nodes, by calculating a normality score as a mean over the relevance scores between the neighboring bottom nodes. Top nodes with low normality scores connect nodes that belong to different communities. Clustering has been used for discovering community structures in bigraphs of companies and board directors (Barber 2007), women attending events (Barber 2007; Doreian, Batagelj, and Ferligoj 2004), supreme court voting (Doreian, Batagelj, and Ferligoj 2004), finding similar users or genres of music from listeners and music groups bipartite graph (Lambiotte and Ausloos 2005), clustering documents based on the occurring terms (Zha et al. 2001), looking for similar actors based on the movies they have played in (Sun et al. 2005), similar authors based on the papers they collaborated (Sun et al. 2005), conferences based on the authors that published, etc.

Projecting the bigraphs into unigraphs results in loss of information (Latapy, Magnien, and Vecchio 2008). Therefore, in this chapter we propose a range of weighting functions for dealing with this problem and we assess how well they represent the relevant underlying structure by comparing the predictive performance of relational classifiers over the unigraph projection. As such, we have an objective function that determines to what extent the predictive information present in the bigraph is also contained in the projected unigraph. There exist some studies in the literature that explore the problem of how to represent the bigraph most accurately with a transformation to unipartite graph. For example, Zweig and Kaufman (Zweig and

Kaufmann 2011) take the approach of connecting nodes in the projection if they have a much higher number of occurrences of motifs (recurrent and statistically significant sub-graphs or patterns) compared to the random graph model of the given bigraph. Furthermore, Zou et al. (Zhou et al. 2007) propose a method for projecting bigraphs into asymmetrical unigraphs, where the weight from one node to another in the projection is not necessarily the same as in the opposite direction. They calculate the weights in the projection by first assigning initial weight to the bottom nodes in the bigraph and then equally distributing them over the neighboring top nodes. In the next phase, the weights are once more distributed, this time from the top to the bottom nodes. This results is a linear equation for each bottom node, where the coefficients signify the link weight in the projection with direction from the specific bottom node. Recently, Gupte and Eliassi-Rad (Gupte and Eliassi-Rad 2012) considered a wide range of measures for weighting the unigraph projections. They defined a set of axioms which approximate the intuition and examined how well the weighting measures in previous literature satisfy this characterisation. The study of Macskassy and Provost (Macskassy and Provost 2007) also discusses how the labels of the related nodes in the unigraph can be simultaneously inferred with the use of collective inference.

The philosophy of this chapter is that the best projection is the one that maximizes performance for a target task. Thus, we should have a framework for systematically exploring the design space. We have proposed various options that can be mixed and matched. Presumably there are others as well that would fit within this framework, as well as alternative possible frameworks.

In addition to the unimodal graphs discussed above that are really bigraph projections, node classification where the bigraph is considered explicitly to our knowledge has so far been limited to a few case-specific studies: predicting interest in financial products from a bigraph of consumers and merchants (Martens and Provost 2011), predicting brand interest from a bigraph of browsers visiting websites (Provost, Dalessandro, et al. 2009) and a bigraph of people visiting locations (Provost, Martens, and Murray 2012). Another exception is the work of Perlich and Provost (Perlich and Provost 2006), who consider classification for datasets with high-dimensional categorical attributes. These attributes can be for example locations which a person visited, identifiers of previously bought books (or other products) by customers and etc. If we consider the persons as bottom nodes and the products or locations as top nodes, it is clear that their approach aggregates the bigraph (or more generally the k-partite graph) information, by applying aggregation operators. This creates new features, which combined with the structured data about the persons, are used by a traditional propositional method. An interesting avenue for future work would be to combine this additional information on the bigraph into the projected unipartite network.

3.6 Conclusion

Bigraph datasets are an intuitive way to represent relational, behavioral and transactional data. The modular three-stage projection framework to leverage such data for node classification has the flexibility to compose a variety of classification methods. The comparison with support-vector machines shows encouraging results: the linear SVM has only average performance when compared to the other combinations of methods (even linear ones) and the popular implementation does not scale to the largest datasets KDDa and KDDb. In our experiments the hyperbolic tangent top node function performs best. The cosine aggregation function, followed by the sum of the shared nodes in combination with the wvRN relational classifier gives the best results. A combination of the latter two is considered by the SW-transformation, a very fast and scalable linear technique that is able to scale to datasets of up to millions of nodes easily, while providing a comprehensible model.

We do not claim to have found the best combination of elements. Rather we argue that within this framework many possibilities exist. As more and more behavioural datasets become available, the prediction over nodes in the corresponding bigraphs will likely see a similar increase in interest. However, given its speed, solid predictive performance, and comprehensibility, we suggest that the SW-transformation provides a very solid baseline method for future studies of methods for predictive modelling with (sparse) bigraph data.

Although, as described earlier, prior studies individually have applied projection to bigraph data, to our knowledge this is the first general study of predictive modelling on bigraph data using projection. Presumably future work could provide significant advances. Moreover, in this chapter we did not address cases where the original bigraph is weighted, nor the important situation where bottom nodes have local information, nor where other features of the top nodes (aggregated) can provide predictive information or where the specific top nodes are discriminative (Perlich and Provost 2006). Hopefully this framework provides a useful stepping stone to such work.

A

Appendix

Table A.1.: Kemeny-Young ranking for all the combinations of techniques.

Top node weight	Aggregation function	Classifier	Best rank	Worst rank	Avg.rank
tanh	cosine function	wvRN	**3.5**	**142.0**	**40.9**
inverse degree	cosine function	wvRN	**3.5**	**140.5**	**41.0**
tanh	cosine function	cdRN	**3.0**	**120.0**	**47.3**
tanh	sum of shared nodes	wvRN	**2.0**	**148.0**	**46.1**
beta distribution	sum of shared nodes	wvRN	**1.0**	**158.0**	**56.4**
tanh	cosine function	nlb	**8.5**	**121.5**	**45.9**
inverse degree	cosine function	nlb	*4.5*	*124.0*	*46.0*
inverse degree	sum of shared nodes	wvRN	**2.0**	**143.0**	**46.4**
inverse frequency	cosine function	wvRN	*2.0*	*140.0*	*39.3*
tanh	jaccard	wvRN	5.5	155.5	50.6
tanh	cosine function	nlb 100	*8.5*	*134.0*	*52.3*
inverse degree	cosine function	nlb 100	4.5	134.0	52.2
inverse degree	cosine function	cdRN	**6.0**	**123.0**	**48.2**
tanh	sum of shared nodes	nlb	**2.0**	**134.5**	**49.0**
inverse degree	sum of shared nodes	nlb	**2.0**	**134.5**	**49.3**
tanh	sum of shared nodes	nlb 100	**2.0**	**134.5**	**54.7**
tanh	sum of shared nodes	cdRN	**1.5**	**130.5**	**50.8**
inverse degree	sum of shared nodes	nlb 100	*2.0*	*134.5*	*55.2*
inverse degree	sum of shared nodes	cdRN	**1.5**	**130.5**	**51.4**
inverse frequency	sum of shared nodes	wvRN	**2.0**	**149.0**	**43.7**
beta distribution	cosine function	cdRN	**1.0**	**158.0**	**57.1**
inverse frequency	sum of shared nodes	nlb	*8.0*	*111.0*	*47.7*
inverse frequency	cosine function	cdRN	**10.0**	**102.0**	**43.7**
inverse frequency	cosine function	nlb	*7.5*	*104.0*	*43.2*
inverse frequency	jaccard	wvRN	*2.0*	*154.0*	*47.5*
inverse degree	jaccard	wvRN	5.5	155.5	52.1
inverse frequency	sum of shared nodes	nlb 100	*8.0*	*134.0*	*51.3*
inverse frequency	sum of shared nodes	cdRN	*9.0*	*159.0*	*51.9*
inverse frequency	cosine function	nlb 100	7.5	134.0	49.1
beta distribution	sum of shared nodes	cdRN	*3.0*	*161.0*	*65.0*
inverse frequency	jaccard	cdRN	**2.0**	**144.0**	**57.4**
tanh	jaccard	nlb	11.0	139.5	62.5
tanh	jaccard	cdRN	*2.0*	*142.5*	*61.9*
inverse degree	jaccard	cdRN	*12.0*	*142.5*	*64.0*
w=1	sum of shared nodes	wvRN	*1.0*	*144.0*	*51.8*
beta distribution	jaccard	wvRN	4.0	161.5	71.4
beta distribution	cosine function	wvRN	**4.0**	**156.0**	**58.2**
beta distribution	max	wvRN	*3.5*	*151.5*	*65.7*
beta distribution	sum of shared nodes	nlb	**4.0**	**160.5**	**72.7**
beta distribution	jaccard	nlb	**4.0**	**156.0**	**55.8**

Continued on next page

Table A.1 – *Continued from previous page*

Top node weight	Aggregation function	Classifier	Best rank	Worst rank	Avg.rank
beta distribution	cosine function	nlb	**4.0**	**156.0**	**58.1**
beta distribution	cosine function	nlb 100	2.0	161.5	68.2
inverse frequency	jaccard	nlb	4.0	145.5	58.2
beta distribution	max	nlb	*3.5*	*151.5*	*65.6*
w=1	cosine function	wvRN	2.0	141.0	52.6
adamic and adar	cosine function	wvRN	3.0	135.0	51.6
w=1	cosine function	cdRN	*7.0*	*105.0*	*56.5*
adamic and adar	sum of shared nodes	wvRN	*1.0*	*138.0*	*53.0*
w=1	jaccard	wvRN	2.0	158.0	54.9
adamic and adar	cosine function	cdRN	*10.0*	*107.0*	*58.2*
inverse degree	jaccard	nlb	10.5	142.5	64.2
delta	cosine function	wvRN	1.0	145.0	57.8
tanh	jaccard	nlb 100	5.5	157.5	76.7
inverse degree	jaccard	nlb 100	5.5	157.5	77.6
beta distribution	sum of shared nodes	nlb 100	4.0	161.0	76.0
tanh	max	wvRN	5.5	159.0	73.3
inverse degree	max	wvRN	5.5	150.5	73.3
inverse frequency	jaccard	nlb 100	4.0	158.0	72.8
w=1	sum of shared nodes	nlb	14.5	103.5	56.8
adamic and adar	sum of shared nodes	nlb	15.0	100.5	57.8
w=1	cosine function	nlb	9.0	103.0	56.1
w=1	sum of shared nodes	nlb 100	14.5	134.0	59.8
w=1	sum of shared nodes	cdRN	13.0	160.0	62.4
adamic and adar	cosine function	nlb	8.5	97.0	56.1
w=1	cosine function	nlb 100	9.0	157.5	64.8
adamic and adar	cosine function	nlb 100	8.5	157.5	64.6
beta distribution	jaccard	nlb 100	1.0	162.0	87.3
w=1	jaccard	cdRN	*4.0*	*156.0*	*65.2*
adamic and adar	jaccard	wvRN	3.0	153.0	58.1
delta	cosine function	nlb	1.5	139.0	64.6
w=1	jaccard	nlb	2.0	158.0	65.4
adamic and adar	sum of shared nodes	cdRN	13.0	158.0	61.5
adamic and adar	sum of shared nodes	nlb 100	15.0	134.0	61.8
delta	cosine function	nlb 100	1.5	157.5	73.6
delta	cosine function	cdRN	1.0	141.0	65.8
beta distribution	jaccard	cdRN	1.0	159.0	75.0
beta distribution	max	cdRN	1.0	162.0	84.9
likelihood ratio	cosine function	wvRN	3.5	146.5	68.1
w=1	jaccard	nlb 100	2.0	159.0	80.7
likelihood ratio	cosine function	nlb	3.5	146.5	69.8
likelihood ratio	cosine function	nlb 100	3.5	157.5	77.1
likelihood ratio	cosine function	cdRN	3.5	152.0	71.3
adamic and adar	jaccard	nlb	6.0	150.0	68.8
adamic and adar	jaccard	cdRN	*8.0*	*152.0*	*69.6*
delta	sum of shared nodes	wvRN	1.0	158.0	62.3
tanh	max	cdRN	13.5	156.0	84.3
likelihood ratio	sum of shared nodes	wvRN	3.5	139.0	75.8
likelihood ratio	sum of shared nodes	nlb	21.5	139.0	77.5
likelihood ratio	sum of shared nodes	nlb 100	21.5	139.0	78.9
likelihood ratio	jaccard	wvRN	1.5	158.0	78.6
likelihood ratio	jaccard	nlb	2.0	158.0	79.6
likelihood ratio	jaccard	cdRN	*1.5*	*154.0*	*78.0*
delta	sum of shared nodes	nlb	1.5	153.0	72.1
tanh	max	nlb	2.5	145.5	83.8
inverse degree	max	nlb	4.0	145.5	83.9
delta	sum of shared nodes	cdRN	1.0	154.0	72.8
tanh	max	nlb 100	15.5	145.5	88.9
SVM			1.0	162.0	91.3
inverse degree	max	nlb 100	11.0	145.5	88.6
inverse degree	max	cdRN	13.5	157.0	84.2
inverse frequency	max	wvRN	7.0	160.0	81.9
delta	sum of shared nodes	nlb 100	1.5	161.5	83.2
delta	max	wvRN	2.0	161.0	80.0
likelihood ratio	sum of shared nodes	cdRN	3.5	162.0	85.5
inverse frequency	max	nlb	2.5	135.5	87.3

Continued on next page

Table A.1 – *Continued from previous page*

Top node weight	Aggregation function	Classifier	Best rank	Worst rank	Avg.rank
inverse frequency	max	nlb 100	3.0	135.5	88.0
inverse frequency	max	cdRN	2.0	137.0	86.1
adamic and adar	jaccard	nlb 100	6.0	157.5	83.6
delta	jaccard	wvRN	1.0	159.0	77.8
likelihood ratio	jaccard	nlb 100	2.5	162.0	92.0
delta	max	cdRN	2.0	158.0	92.9
delta	max	nlb	4.0	160.0	92.3
delta	jaccard	nlb	16.5	155.5	88.9
delta	max	nlb 100	4.0	162.0	97.1
delta	jaccard	cdRN	18.0	154.0	89.3
beta distribution	max	nlb 100	24.0	162.0	103.1
adamic and adar	max	wvRN	11.0	148.0	100.7
delta	jaccard	nlb 100	16.5	162.0	106.8
adamic and adar	max	cdRN	13.0	162.0	103.7
adamic and adar	max	nlb	16.0	131.5	100.8
adamic and adar	max	nlb 100	12.0	161.0	105.6
likelihood ratio	max	wvRN	32.5	148.5	109.6
likelihood ratio	max	nlb	32.5	148.5	111.7
likelihood ratio	max	nlb 100	42.0	161.0	113.1
likelihood ratio	max	cdRN	66.5	160.0	117.3
any	zero - one	wvRN	21.0	157.0	119.7
w=1	max	wvRN	21.0	157.0	119.7
any	zero - one	nlb	9.0	157.0	118.2
w=1	max	nlb	9.0	157.0	118.2
any	zero - one	cdRN	22.0	158.0	124.9
w=1	max	cdRN	22.0	158.0	124.9
any	zero - one	nlb 100	28.0	158.0	127.7
w=1	max	nlb 100	28.0	158.0	127.7

Dataset	Target Label	l_0	l_1	$n_\top$	$n_\perp$	m	$k_\top$	$k_\perp$	k	δ
MovieLens100k	gender	273	670	1,682	943	100,000	59.45	106.04	76.19	0.063
MovieLens100k	age	448	495	1,682	943	100,000	59.45	106.04	76.19	0.063
MovieLens100k	genre	-	-	943	1,682	100,000	106.04	59.45	76.19	0.063
MovieLens100k (above average)	gender	273	670	1,574	943	82,520	52.43	87.51	65.57	0.0556
MovieLens100k (above average)	age	448	495	1,574	943	82,520	52.43	87.51	65.57	0.0556
Yahoo Movies	gender	2,206	5,436	11,915	7,642	221,330	18.57	28.96	22.63	0.0024
Yahoo Movies (above average)	gender	2,206	5,431	10,547	7,637	181,470	17.20	23.76	19.96	0.0023
Yahoo Movies	age	2,750	4,855	11,911	7,605	220,595	18.52	29.01	22.61	0.0024
Yahoo Movies (above average)	age	2,748	4,852	10,544	7,600	180,880	17.15	23.80	19.93	0.0023
TaFeng	age	17,330	14,310	23,719	31,640	723,449	30.5	22.86	26.14	9.6400e-04
TaFeng (above avg)	age	5,051	11,299	18,126	16,350	234,355	12.93	14.33	13.59	7.9078e-04
Norwegian companies	gender	513	908	355	1,421	1,746	4.92	1.23	1.97	0.0035
Reality Mining	affiliation	-	-	12,043	95	76,674	6.37	807.09	12.63	0.067
Book-Crossing	age	38,168	23,662	284,175	61,830	835,495	2.94	13.51	4.83	4.7551e-05
Book-Crossing (above average)	age	25,729	16,421	127,709	42,150	259,333	2.03	6.15	3.05	4.8177e-05
LibimSeTi	gender	43,510	57,606	135,359	101,116	13,594,717	100.43	134.45	114.98	9.932e-004
LibimSeTi (above average)	gender	40,878	54,459	135,346	95,337	8,169,662	60.36	85.69	70.83	6.3314e-04
Flickr	comments	8,177,007	3,030,449	497,472	11,195,144	34,645,469	69.64	3.09	5.926	6.2208e-06
KDDa	task performance	1,235,867	7,171,885	19,306,083	8,407,752	305,613,510	15.82	36.34	22.05	1.8828e-06
KDDb	task performance	2,684,437	16,579,660	29,890,095	19,264,097	566,345,888	18.94	29.39	23.04	9.8357e-07

Table A.2.: Descriptive statistics of the bipartite datasets: class distribution (l_0, l_1) of the bottom nodes, number of top ($n_\top$) and bottom ($n_\perp$) nodes, number of edges (m), average degree for top ($k_\top$) and bottom ($k_\perp$) nodes, average combined degree (k) and density ($\delta(G)$). The basic bipartite statistics used in this table are defined in Section 3.2.

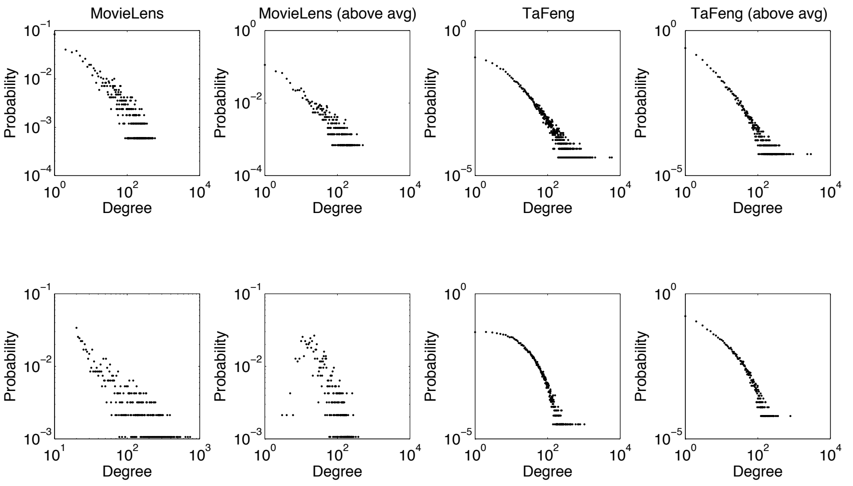


Figure A.1.: Degree distributions of the top nodes (upper row) and bottom nodes (bottom row) for different datasets.

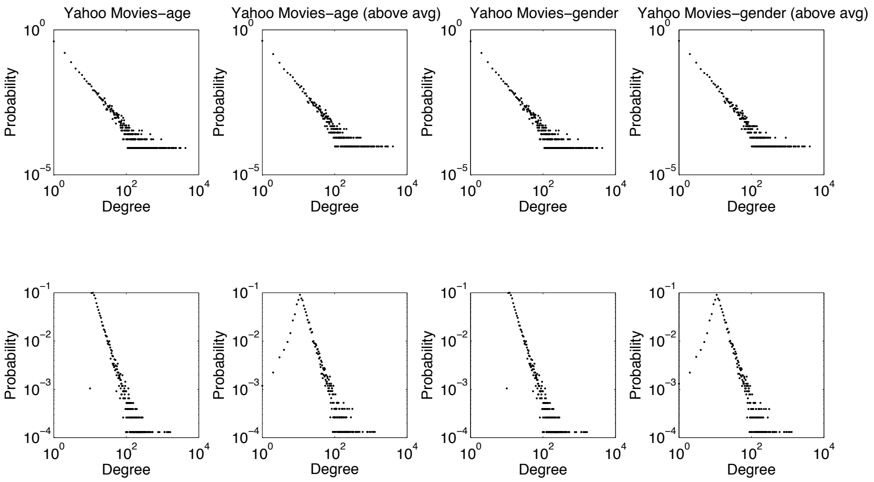


Figure A.2.: Degree distributions of the top nodes (upper row) and bottom nodes (bottom row) for different datasets.

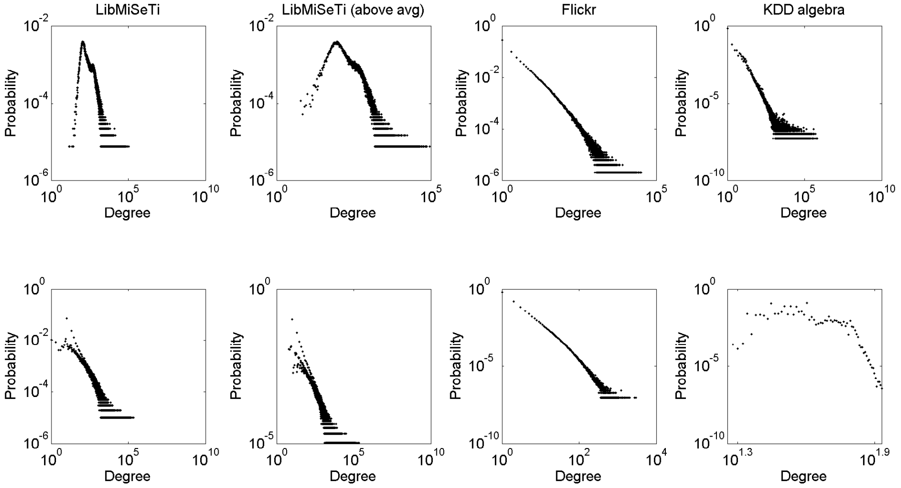


Figure A.3.: Degree distributions of the top nodes (upper row) and bottom nodes (bottom row) for different datasets.

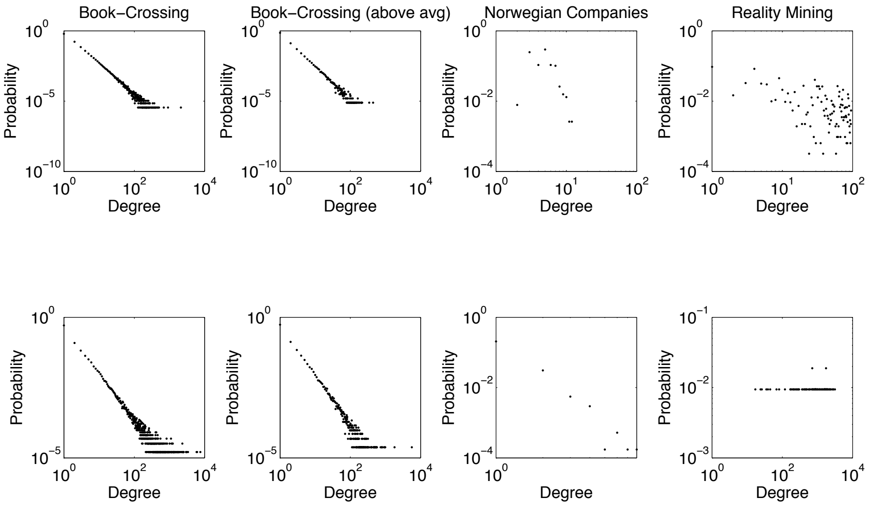


Figure A.4.: Degree distributions of the top nodes (upper row) and bottom nodes (bottom row) for different datasets.

A.1 Run-time performance

In this section we examine the run-time performance of the different techniques from the three stages. [1] We start by comparing the average durations of each of the techniques over the datasets. For this, we only consider the datasets with less than 100,000 nodes since not all the methods are able to run on the larger datasets. Figures A.6-A.9 give the detailed results of the averaged durations for each of the top node functions and aggregation functions per relational learner. In short, for each of the relational learners the maximum and the Jaccard aggregation function have the longest durations, especially when combined with the beta top node function. The beta top node function takes longest to run due to the tuning procedure. In our setting we tune the beta function with a grid search on three levels. In Table A.3 we can see how the AUC improves on different levels of the grid search for several datasets. For each dataset the best chosen parameters of the specific level are shown with the corresponding AUC value. One can observe that usually, the optimal α and β parameters give a shape of the curve such that the nodes with a smaller degree receive a higher weight. For example, in the first level for most of the datasets the optimal α is 0.1 and β is 3.1, which is exactly this shape of the curve. This may be included in the grid search for time improvement, to only take into account the parameters which give this kind of shape. Moreover, the grid search can be reduced to only one or two levels, which results in limited performance decrease (see Table A.3). We also examined tuning the α and β parameters on a smaller sample of the training data. In Figure A.5 we can see that the predictive performance for most of the datasets are stable even when the beta function is tuned on a much smaller subset of 1000 data points. This speeds up the tuning procedure up to several times, especially for the larger datasets. The rest of the top node functions are faster than the beta function, with similar durations.

The learning of the weights for the nLB classifier can also be done on a smaller sample. Therefore, we also examine the time advantage of using this approach (see Figure A.10). In our experiments even a sample of less than 100 instances was enough to tune the parameters of the nLB logistic regression. We consider this nLB classifier trained with only 100 instances as a third relational classifier, named nLB100 in the results. Although it performs slightly worse than the regular nLB classifier in terms of AUC, it can be tuned much faster. The time advantage is clearly larger on bigger datasets, like for example the BookCrossing dataset. However, when the class-label autocorrelation is uncertain and the training time is an issue, it may be better simply to use the cdRN classifier, which is fast and whose performance is quite robust to different sorts of relational autocorrelation.

The SW-transformation outperforms all the other aggregation functions in combination with any non-tuning top node function. It is able to scale to big datasets as it runs very fast. For example for the biggest dataset we used, KDDb with dimensions

[1] All experiments are conducted on a 3.40 GHz Intel i7 CPU, with 8 GB RAM and a 64-bit operating system.

Dataset	Level 1			Level 2			Level 3		
	Alpha	Beta	AUC	Alpha	Beta	AUC	Alpha	Beta	AUC
MovieLens gender	3.1000	12.1000	0.7019	3.1000	15.1000	0.7099	2.7667	16.1000	0.7110
MovieLens gender (above average)	0.1000	6.1000	0.7593	1.1000	9.1000	0.7622	0.4333	8.7667	0.7690
MovieLens age	0.1000	3.1000	0.8087	0.1000	1.1000	0.8106	0.1000	1.4333	0.8110
MovieLens age (above average)	0.1000	3.1000	0.8193	1.1000	6.1000	0.8231	1.1000	7.1000	0.8242
Yahoo Movies (gender)	0.1000	3.1000	0.7985	0.1000	1.1000	0.8046	0.4333	1.4333	0.8060
Yahoo Movies above average (gender)	0.1000	3.1000	0.7955	0.1000	1.1000	0.8026	0.1000	1.1000	0.8026
Yahoo Movies (age)	0.1000	3.1000	0.6637	0.1000	3.1000	0.6637	0.4333	3.7667	0.6698
Yahoo Movies above average (age)	0.1000	3.1000	0.6577	0.1000	1.1000	0.6594	0.1000	1.1000	0.6594
TaFeng	0.1000	3.1000	0.6861	0.1000	1.1000	0.6894	0.4333	2.1000	0.6969
TaFeng (above average)	0.1000	3.1000	0.7198	0.1000	2.1000	0.7199	0.1000	2.4333	0.7199
BookCrossing	0.1000	0.1000	0.5892	0.1000	0.1000	0.5892	0.4333	0.4333	0.5913
BookCrossing (above average)	0.1000	0.1000	0.5716	1.1000	3.1000	0.5732	0.7667	3.4333	0.5738
LibimSeTi	0.1000	3.1000	0.8461	0.1000	1.1000	0.8483	0.4333	1.7667	0.8487
LibimSeTi (above average)	0.1000	3.1000	0.8669	0.1000	1.1000	0.8676	0.1000	1.4333	0.8676
Flickr	6.1000	0.1000	0.7337	6.1000	0.1000	0.7337	5.7667	0.1000	0.7341
KDDa	0.1000	12.1000	0.7888	0.1000	15.1000	0.7892	0.1000	16.1000	0.7791

Table A.3.: Beta grid search on three levels with the optimal α and β parameters, as well as the coresponding AUC per level. The aggregation function used is the sum of shared nodes in combination with the wvRN relational classifier.

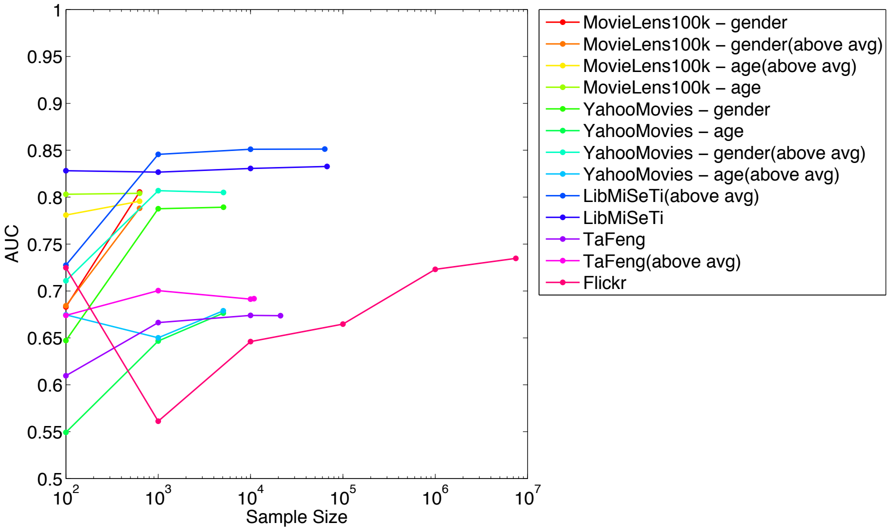


Figure A.5.: Predictive performance of the beta function in combination with SW-transformation when the parameters are tuned on a sample of the training data and trained on the full training data

of around 19 million × 30 million, the SW-transformation with a regular top node function that does not require tuning (i.e., not the beta function) takes around 9 minutes to finish. This dataset did not fit in memory, so we performed batch processing directly from disk. For the KDDa dataset (dimensions of around 8 million × 20 million) and the Flickr dataset (11 million × half a million) the SW-transformation

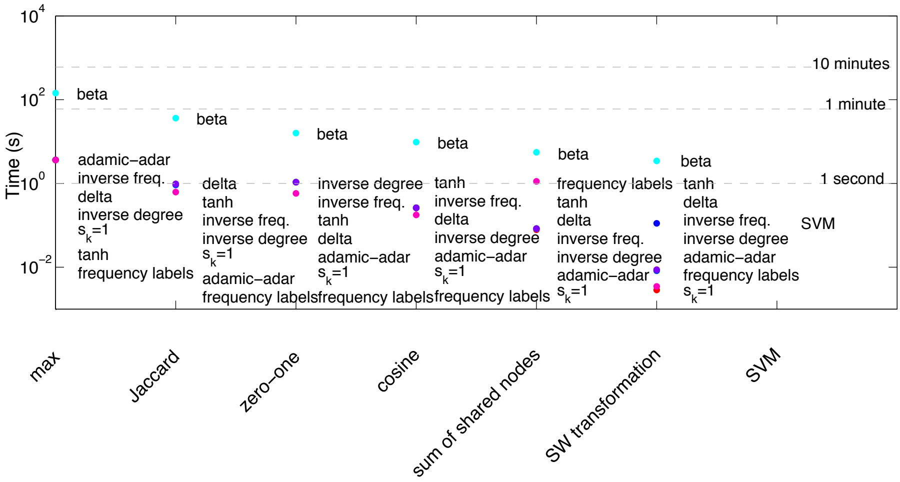


Figure A.6.: Aggregated run-time results for each of the top node and aggregation functions with wvRN (including the SW-transformation). Since most of the top node functions (except for the beta) have similar durations, the markers on the plots are very close to each other (and given in descending order). The SW-transformation outperforms all the other aggregation functions in combination with any non-tuning top node function.

takes 5 minutes and less than a minute respectively. This represents a substantial scalability and time improvement over the regular sum of shared nodes and wvRN implementations, even with batch processing. The clear time advantage of the SW-transformation over each dataset can be seen in Figure A.11, where the average time needed for the regular sum of shared nodes and wvRN over the datasets is 65.4 seconds and the SW-transformation needs only 0.5478 seconds on average.

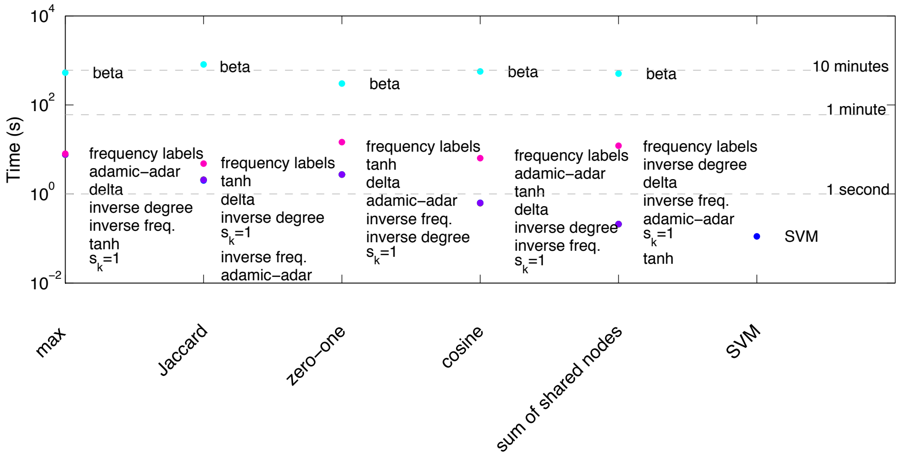


Figure A.7.: Aggregated run-time results for each of the top node and aggregation functions with the nLB classifier.

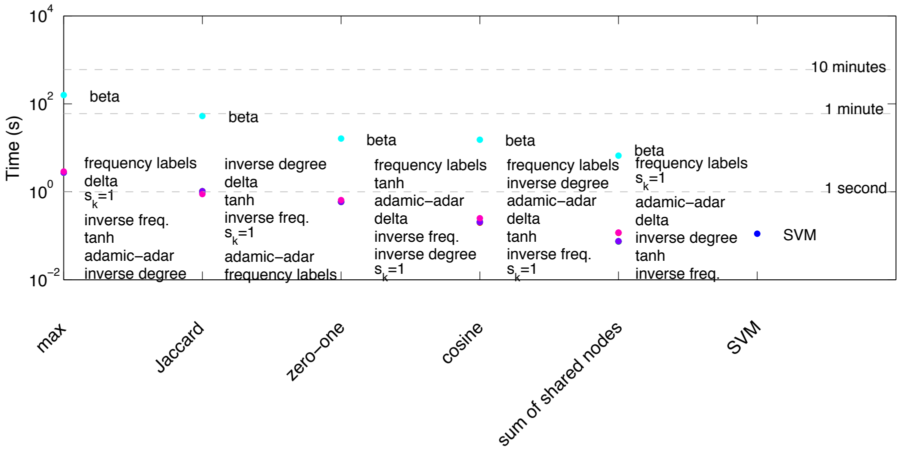


Figure A.8.: Aggregated run-time results for each of the top node and aggregation functions with the nLB100 classifier (nLB with 100 training instances).

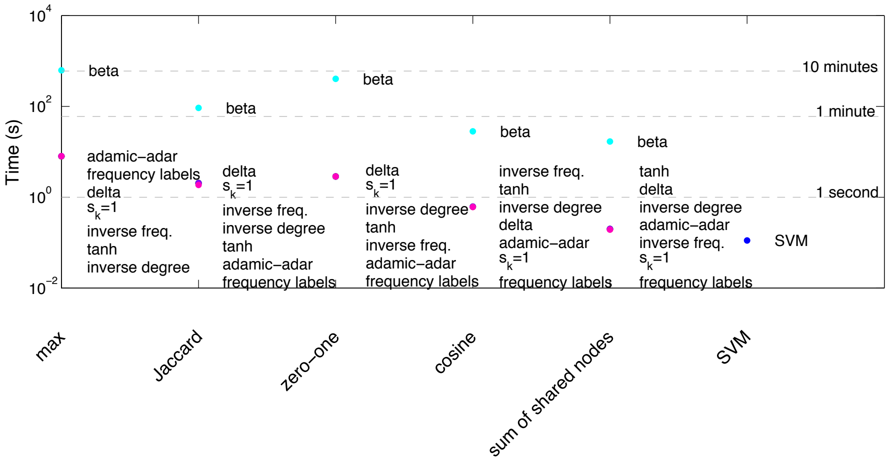


Figure A.9.: Aggregated run-time results for each of the top node and aggregation functions with the cdRN classifier.

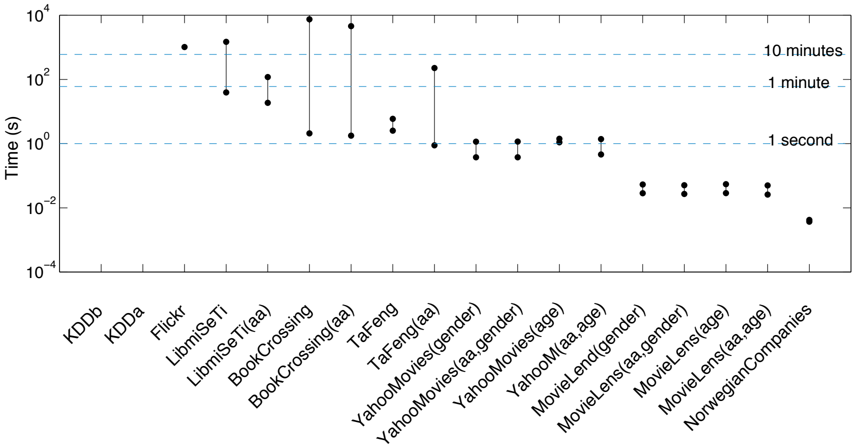


Figure A.10.: Time improvement of nLB with sampling over 100 instances as compared to no sampling for different datasets. The top of each bar represents the time needed for the nLB classifier and the bottom of each bar the time required to train the nLB with 100 instances for the specific dataset.

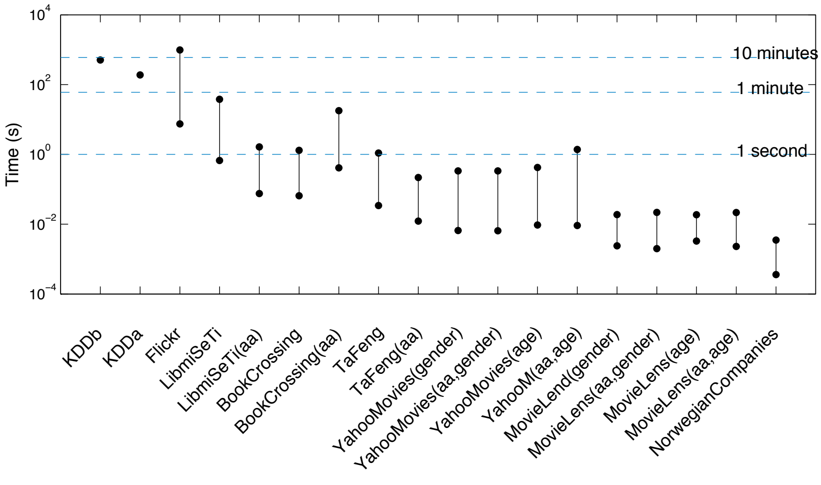


Figure A.11.: Time improvement of the SW-transformation over wvRN and sum of shared nodes for different datasets. The top of each bar represents the time needed for the wvRN classifier and the bottom of each bar the time required for SW-transformation for the specific dataset.

Dataset	Target	Top nodes function	Aggregation function	Relational classifier	AUC
KDD B		delta	sum of shared nodes	wvRN	0.8054
KDD algebra		beta distribution	sum of shared nodes	wvRN	0.7791
Flickr	target:comments	SVM			0.7602
LibmiSeTi	target:gender	tanh	cosine function	wvRN	0.8562
LibmiSeTi (above average)	target:gender	tanh	cosine function	wvRN	0.8762
TaFeng consumers products	target:age	beta distribution	sum of shared nodes	nlb	0.6785
TaFeng consumers products (above average)	target:age	delta	sum of shared nodes	nlb 100	0.7564
Yahoo Movies	target:gender	tanh	sum of shared nodes	wvRN	0.8071
Yahoo Movies (above average)	target:gender	tanh	sum of shared nodes	nlb	0.8070
Yahoo Movies	tareget:age	beta distribution	sum of shared nodes	cdRN	0.6763
Yahoo Movies (above average)	tareget:age	beta distribution	cosine function	cdRN	0.6795
MovieLens100k	target:gender	inverse degree	sum of shared nodes	wvRN	0.8071
MovieLens100k (above average)	target:gender	tanh	sum of shared nodes	wvRN	0.8104
MovieLens100k	target:age	SVM			0.8685
MovieLens100k (above average)	target:age	SVM			0.8543
MovieLens100k	target:genre [2]Action	delta	sum of shared nodes	cdRN	0.7743
MovieLens100k	target:genre [3]Adventure	beta distribution	cosine function	cdRN	0.8615
MovieLens100k	target:genre [4]Animation	beta distribution	jaccard	wvRN	0.9180
MovieLens100k	target:genre [5]Children's	likelihood ratio	jaccard	wvRN	0.8835
MovieLens100k	target:genre [6]Comedy	SVM			0.7135
MovieLens100k	target:genre [7]Crime	w=1	jaccard	wvRN	0.6632
MovieLens100k	target:genre [8]Documentary	delta	sum of shared nodes	nlb	0.6775
MovieLens100k	target:genre [9]Drama	SVM			0.7232
MovieLens100k	target:genre [10]Fantasy	beta distribution	sum of shared nodes	wvRN	0.8131
MovieLens100k	target:genre [11]Film-Noir	SVM			0.6948
MovieLens100k	target:genre [12]Horror	delta	sum of shared nodes	cdRN	0.7207
MovieLens100k	target:genre [13]Musical	likelihood ratio	jaccard	wvRN	0.9118
MovieLens100k	target:genre [14]Mystery	likelihood ratio	jaccard	wvRN	0.6166
MovieLens100k	target:genre [15]Romance	delta	cosine function	cdRN	0.6443
MovieLens100k	target:genre [16]Sci-Fi	likelihood ratio	jaccard	wvRN	0.8451
MovieLens100k	target:genre [17]Thriller	delta	cosine function	cdRN	0.6883
MovieLens100k	target:genre [18]War	beta distribution	max	cdRN	0.5502
MovieLens100k	target:genre [19]Western	tanh	sum of shared nodes	cdRN	0.8836
MovieLens100k (above average)	target:genre [2]Action	beta distribution	jaccard	nlb 100	0.8283
MovieLens100k (above average)	target:genre [3]Adventure	beta distribution	jaccard	nlb 100	0.8358
MovieLens100k (above average)	target:genre [4]Animation	beta distribution	sum of shared nodes	wvRN	0.9061
MovieLens100k (above average)	target:genre [5]Children's	w=1	sum of shared nodes	wvRN	0.8965
MovieLens100k (above average)	target:genre [6]Comedy	delta	cosine function	nlb	0.7386
MovieLens100k (above average)	target:genre [7]Crime	beta distribution	jaccard	nlb 100	0.6885
MovieLens100k (above average)	target:genre [8]Documentary	SVM			0.7523
MovieLens100k (above average)	target:genre [9]Drama	beta distribution	max	cdRN	0.7194
MovieLens100k (above average)	target:genre [10]Fantasy	beta distribution	jaccard	cdRN	0.8810
MovieLens100k (above average)	target:genre [11]Film-Noir	beta distribution	jaccard	nlb 100	0.7901
MovieLens100k (above average)	target:genre [12]Horror	delta	sum of shared nodes	wvRN	0.8038
MovieLens100k (above average)	target:genre [13]Musical	delta	jaccard	wvRN	0.8415
MovieLens100k (above average)	target:genre [14]Mystery	beta distribution	jaccard	nlb 100	0.6971
MovieLens100k (above average)	target:genre [15]Romance	delta	cosine function	wvRN	0.6972
MovieLens100k (above average)	target:genre [16]Sci-Fi	delta	sum of shared nodes	wvRN	0.8063
MovieLens100k (above average)	target:genre [17]Thriller	beta distribution	jaccard	nlb 100	0.7558
MovieLens100k (above average)	target:genre [18]War	likelihood ratio	jaccard	wvRN	0.6264
MovieLens100k (above average)	target:genre [19]Western	adamic and adar	sum of shared nodes	wvRN	0.9275
Reallity Mining	target:status[1]1styeargrad	beta distribution	jaccard	nlb	0.8505
Reallity Mining	target:status[2]mlgrad	likelihood ratio	max	wvRN	0.6255
Reallity Mining	target:status[3]sloan	delta	cosine function	cdRN	0.6710
Reallity Mining	target:status[4]mlstaff	delta	max	wvRN	0.7586
Reallity Mining	target:status[6]grad	likelihood ratio	sum of shared nodes	cdRN	0.7258
Reallity Mining	target:status[7]mlurop	likelihood ratio	sum of shared nodes	cdRN	0.7258
Norwegian companies	target:gender	tanh	max	nlb	0.7244

Table A.4.: Best combinations of methods per dataset.

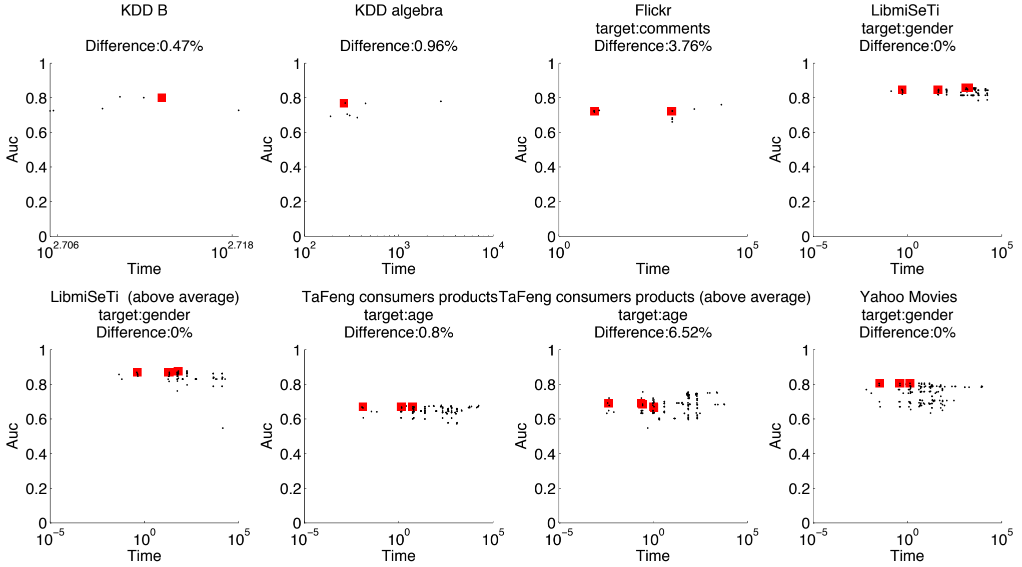


Figure A.12.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

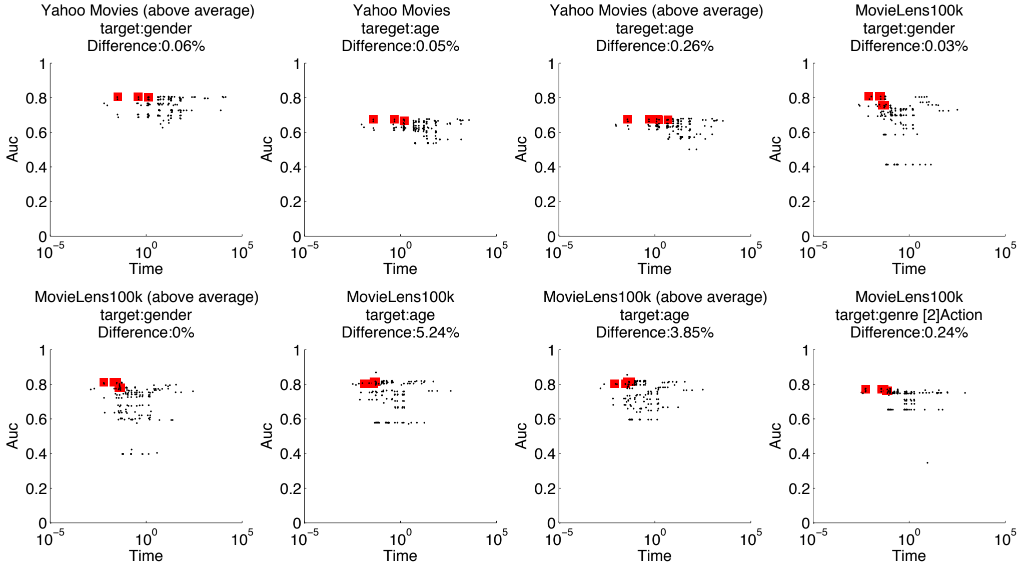


Figure A.13.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

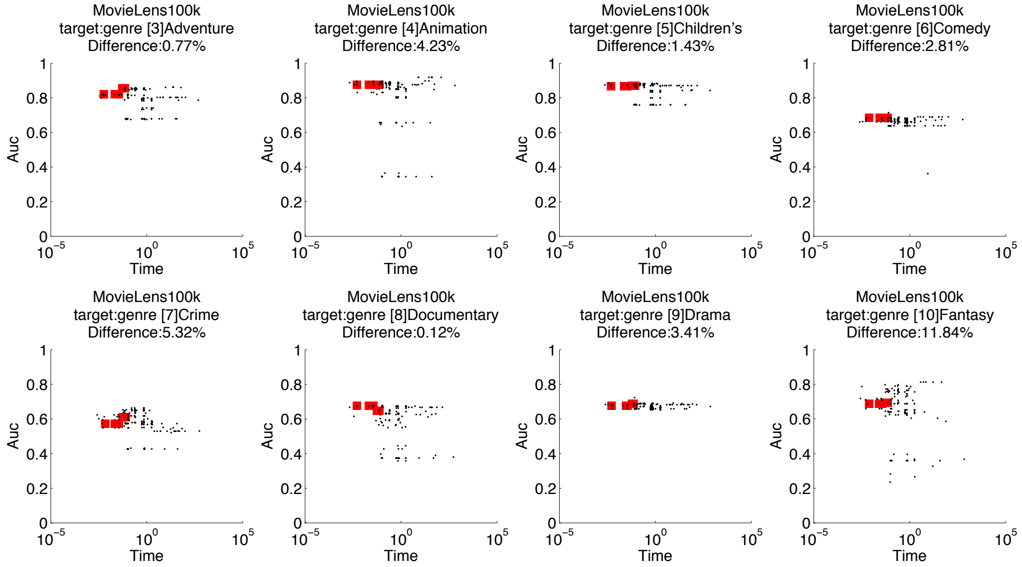


Figure A.14.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

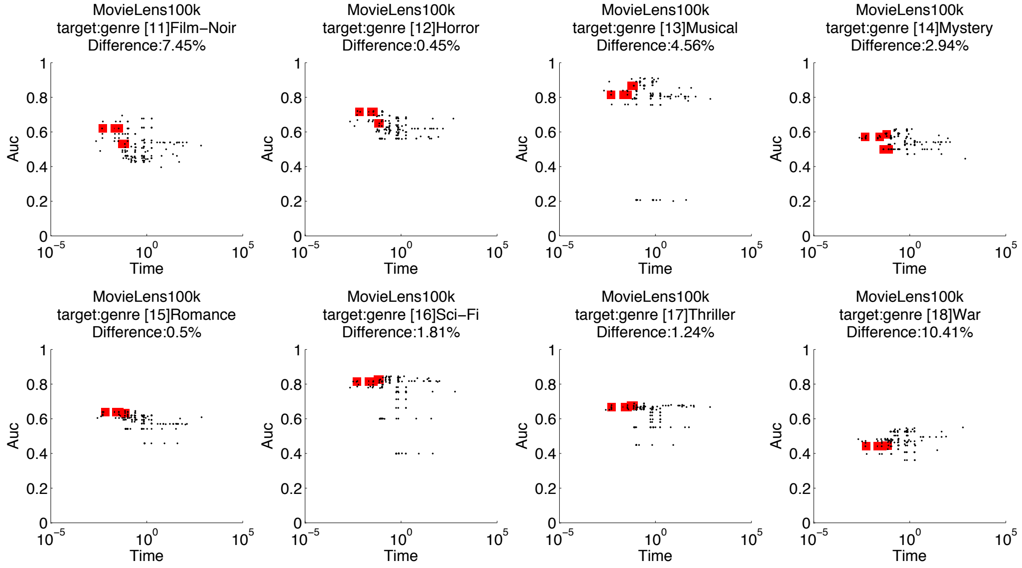


Figure A.15.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

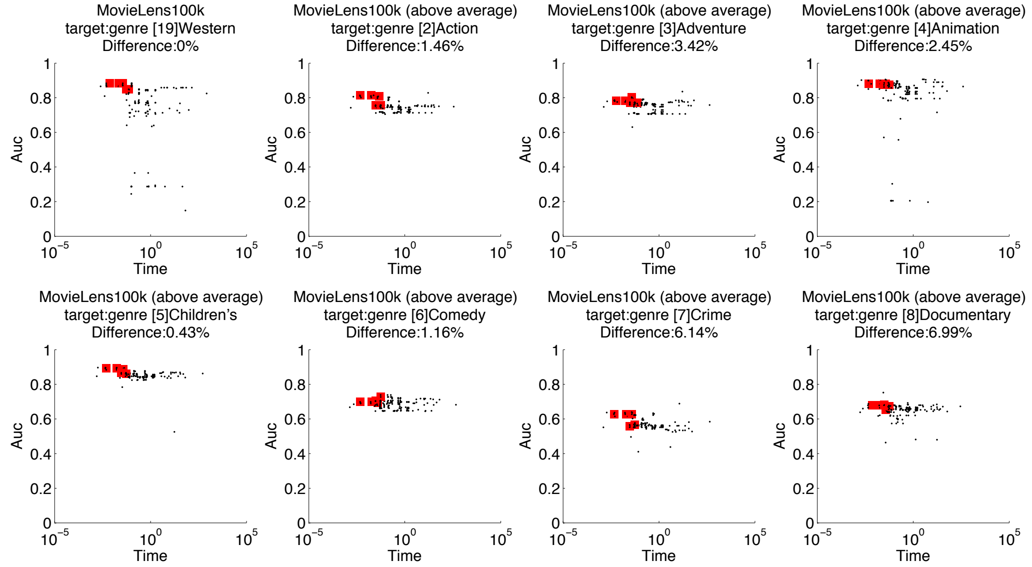


Figure A.16.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

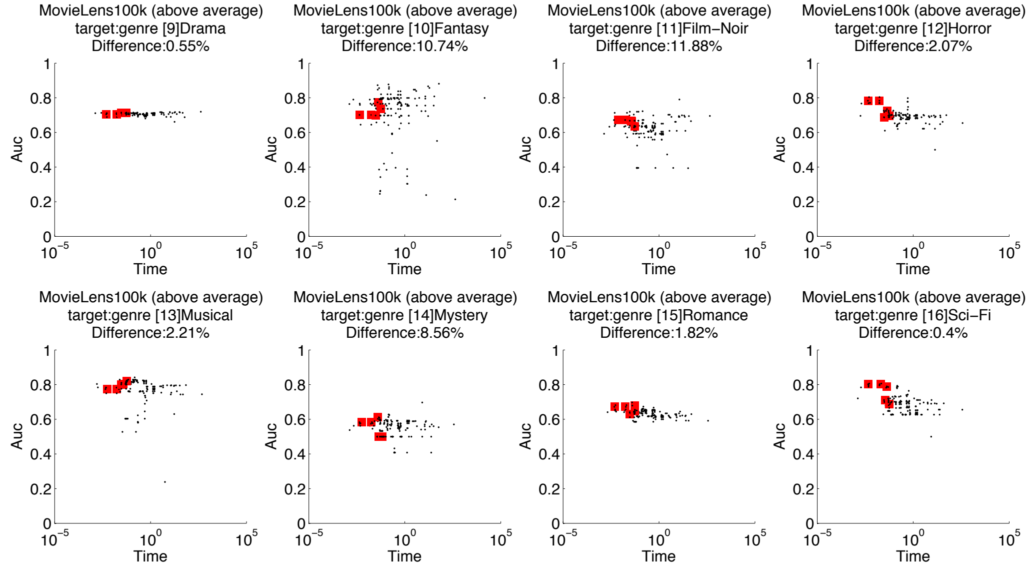


Figure A.17.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

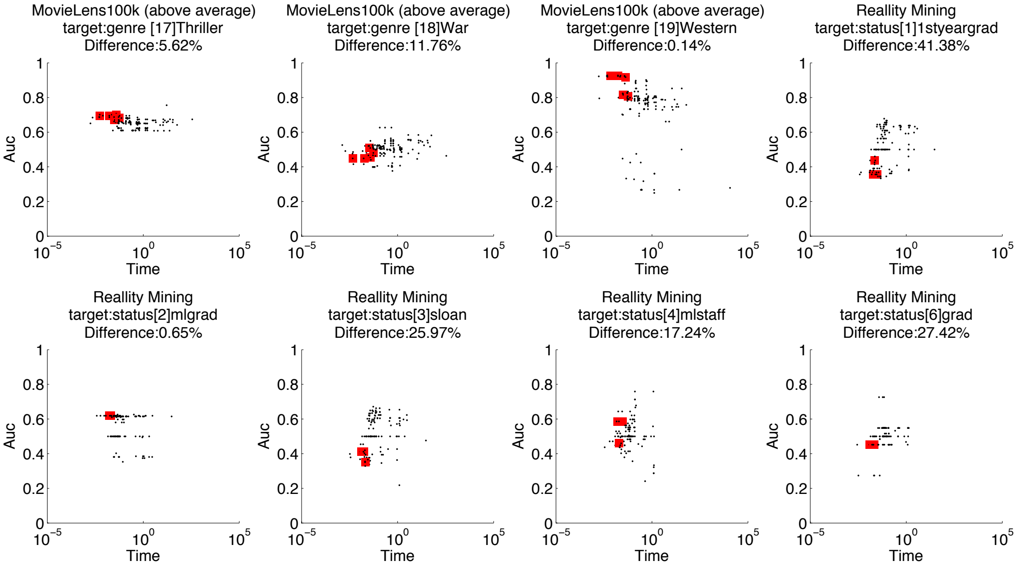


Figure A.18.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

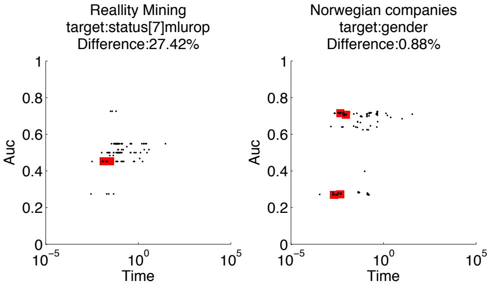


Figure A.19.: Ranking of all combinations of methods, with the proposed combinations highlighted in red.

4

Classification within weighted bigraphs through projection

As discussed in the previous chapter, many real world datasets demonstrate a bipartite structure and can be modelled as bigraphs. Moreover, most of these datasets are complex and have some strength or capacity associated with the relationships between the entities that can be represented as bigraph link weights. Since the graph mining literature (for both unigraphs and bigraphs) is mainly focused on designing metrics and techniques for the more basic case of unweighted graphs, these weights are usually either ignored or cast to binary values based on some threshold. In order to explore the added value of retaining the bigraph link weights, we build upon our previous work and propose a multiple-stage framework for node classification within weighted bigraphs in this chapter. Our empirical results show that the bigraph link weights indeed hold valuable information and utilizing them for node classification leads to significantly better predictions. Additionally, we compare the predictive and run-time performance of the techniques from our framework to two other alternatives for node classification using: (i) propositional learners over the weighted bigraph adjacency matrix and (ii) weighted k-nearest neighbour classifier or a relational learner in combination with existing vector similarity techniques. The results are encouraging and show that the predictive performance of the alternative settings is only average compared to most of the combinations from our framework. Furthermore, there is no statistically significant difference between the weighted and unweighed versions of these alternative settings.

4.1 Introduction

In the previous chapter we discussed how bigraphs are an intuitive representation for many relational, behavioural and transactional datasets. Often these bigraphs have some strength or capacity associated with the relationships between the nodes, that can be modelled as a weight of the links (or edges). Some examples include datasets of people rating different types of products where the weights represent rating scores (Bennett and Lanning 2007; Brozovsky and Petricek 2007; Koenigstein, Dror, and Koren 2011; Linden, Smith, and York 2003; Ziegler et al. 2005); frequency of visits by animal species to plants in search of food resources (Padrón, Nogales, and Traveset 2011); time spent on websites by users (Claypool et al. 2001); number of times a user played a song (McFee et al. 2012) or watched a TV-show (Y. Hu, Koren, and Volinsky 2008); number of products bought by a customer (Hsu, H.-H. Chung, and H.-S. Huang 2004); amount of money transferred between consumers and payment receivers (Martens and Provost 2011), etc. Although most of the real networks are complex by nature and have weights associated with the links, the literature so far has been mainly focused on designing methods for the more basic case of unweighted graphs (Latapy, Magnien, and Del Vecchio 2008). Due to this lack of methods for weighted graphs, they are often transformed into unweighed versions by either (i) assigning an equal weight of one to any existing link in the network or (ii) by using a predefined threshold to create binary relationships (M. Newman 2010; Wasserman 1994). For the later case, all the links that have a weight lower than the threshold are deleted and a weight of one is assigned to the remaining links [1]. In both cases, potentially valuable information is being discarded.

In order to examine the added value of retaining the link weight information, we build upon our work from Chapter 3 and propose a framework for node classification within weighted bigraphs. By not discarding the bigraph link weights, we can preserve more of the complexity and the richness of the data, which in turn could possibly lead to improved classification. The framework from this chapter is based on the same principle of projecting the bigraph into a unigraph (see Figure 2.2), so that we can use the wealth of techniques available for unigraphs. As discussed previously, the projection is created by connecting the nodes from one set of the bigraph if they share at least one node from the other set (Latapy, Magnien, and Del Vecchio 2008) (see Figure 2.2). Since this is an irreversible process, we can preserve more information about the underlying bigraph by adding weights to the projection. The earlier studies have mainly included information about the node degrees in these projection weights (see Section 3.5), to which we will refer to as topological features. Nevertheless, we would like to create a more representative projection of the bigraph by also adding information about the link weights. We propose a multiple step framework for node classification within weighted bigraphs, where we combine the two types of weights based on the bigraph topology and the link weights at a level of a top node and then aggregate them over all the shared top

[1] In Chapter 3 we employed both approaches, ignoring the link weights or discarding the links with lower than an average weight (the datasets are annotated as ”above average” in Table A.2 from Appendix A).

nodes to calculate the similarity (weight) between two nodes in the projection. Once the weighted projection is calculated, standard unigraph relational learners can be applied. For each of the framework stages, we look at several types of methods that comply with the intuition described in a set of *preferred properties*. Alternatively, we also consider two other approaches for node classification within weighted bigraphs using: (i) propositional learners over the weighted bigraph adjacency matrix and (ii) existing vector similarity techniques in combination with a weighted k-nearest neighbour classifier or a relational learner. We provide an empirical evaluation and comparison of the predictive and run-time performance of the techniques from the three different settings. As a running example, we consider the bigraph of persons visiting locations from the previous chapter, where we additionally assume that we have information about the frequency of visits, i.e. the link weights. In summary, the work presented in this chapter provides several contributions:

1. It presents a multiple-stage framework for node classification within weighted bigraphs through projection. For each of the framework stages, we propose a set of methods that can be mixed-and-matched to create a classification technique.
2. It proposes a set of preferred properties that comply with the intuition regarding how the information about the bigraph link weights should be included in the projection. These properties directly guide the design of the methods used in the different stages of the framework.
3. It evaluates two other alternative approaches for node classification within weighted bigraph: using propositional learners and vector similarity techniques in combination with a weighted k-nearest neighbour classifier or a relational learner.
4. It examines whether the bigraph link weights hold information that can be utilized for node classification. Furthermore, it assesses the added value of combining information about both the bigraph link weights and the topology of the bigraph and whether it results in better classification.

4.2 Theoretical View

Formally, a weighted bipartite graph can be defined as the set $G = (\top, \bot, E, w)$, where $\top$ denotes a set of top nodes, $\bot$ is a set of bottom nodes, $E \subseteq \top \times \bot$ is a set of links (also known as edges) and $w : \mathbb{E} \rightarrow R^+$ associates a real number called a weight to the links (Duan and H.-H. Su 2012). As defined above, we refer to the set of nodes for which we have an available target variable as bottom nodes ($\bot$) and the opposite set as top nodes of the bigraph ($\top$). Beside the graph representation, the weighted bigraph can also be depicted with a weighted adjacency matrix. This matrix is similar to the one from Figure 2.1, except that instead of binary values, the elements have a value that is equal to the corresponding link weight. In this chapter, we only consider networks with positive weights and we also assume that the weights can not be zero; a value of zero in the adjacency matrix signifies no connection between the nodes.

In the rest of this section we first introduce our framework for node classification within weighted bigraphs. We then look at the common properties of various bigraph link weights and classify the weights according to them. Based on this categorization, we propose a set of preferred proeprties that grasp the intuition of how the bigraph link weights should be included in the projection through our framework.

4.2.1 *Framework*

The framework for node classification within weighted bigraphs is outlined in Figure 4.1. It utilizes information from both the topology and the link weights of the bigraph to create a representative weighted projection, where a higher link weight signifies larger similarity between the nodes. Once we have created the weighted projection, we can apply standard relational learners for unigraphs. In this chapter, we define the basic component of a weighted bigraph, named a *module*, as a subgraph of two bottom nodes connected with weighted links to a top node. Between every pair of bottom nodes in the bigraph, we can define as many modules as there are shared top nodes. For instance, the nodes 1 and 5 from Figure 2.1 have only one module which is the subgraph 1-A-5. We use this basic building block to calculate a partial similarity score between two bottom nodes. The final weight in the projection is an aggregation of all partial similarity scores calculated for the two bottom nodes. More specifically, the framework includes the following stages:

1. Firstly, we calculate a link weight l_{ijk} for every module in the bigraph (see Figure 4.1.1). The weight captures the resemblance between the nodes i and j based on the bigraph link weights b_{ik} and b_{jk} with node k. Note that the weight is calculated only for nodes that are connected in the projection, i.e. that have top nodes in common. In the rest of this section, we present a set of preferred properties that articulate the intuition of how this weight should be created.
2. Furthermore, we calculate a weight s_k for the top nodes in the bigraph, that preserves the topological information regarding the node degree (d_k) (see Figure 4.1.2). As discussed previously in Section 3.2.1, the lower degree nodes that are connected to only a few bottom nodes are more discriminative and provide more information about the target variable. Thus, they should obtain a higher weight. In the context of our bigraph of persons visiting locations, the reasoning is that a visit to a less popular place (e.g. someone's apartment or the local organic food store) would tell us more about the shared characteristics and interests of the people, than a visit to a very busy place like Central Park.
3. In the following stage, we combine the topological weight s_k and the link weight l_{ijk} into a component weight c_{ijk} (a partial similarity score) at the

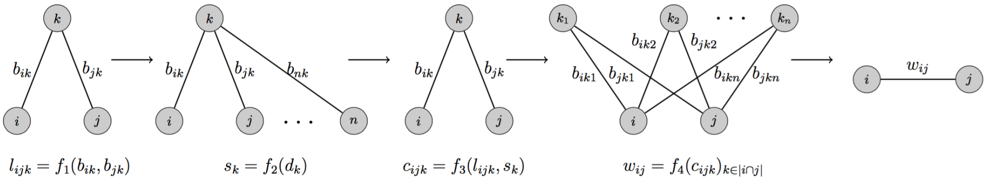


Figure 4.1.: Framework for node classification within weighted bigraphs: (1) calculate the aggregated link weight l_{ijk} from the bigraph link weights b_{ik} and b_{jk}, (2) create the top node weight s_k based on the node degree d_k, (3) combine the aggregated link weight l_{ijk} and the top node weight s_k into a component weight c_{ijk}, (4) aggregate the component weights c_{ijk} into a weight of the projection w_{ij}, (5) apply unigraph relational learners over the projection.

level of a module (see Figure 4.1.3). At this stage, an interesting question is whether both the topological and the link weights equally well demonstrate the similarity between the nodes and thus should they equally influence the combined weight. We look at this into more detail in Section 4.3 by exploring different flexible functions that can determine the importance of both weights for each dataset individually.

4. In the fourth phase, we aggregate all the component weights into a weight of the projection w_{ij}, that indicates the total similarity between the nodes i and j (see Figure 4.1.4).

5. Lastly, unigraph relational learners are applied to the weighted projection.

4.2.2 *Types of bigraph link weights*

The bigraph link weights correspond to the strength or the capacity of the relationships between the nodes. They can represent different concepts such as frequency of visits to a location (Provost, Martens, and Murray 2015b), time spent browsing a website (Claypool et al. 2001), number of products bought by a customer (Hsu, H.-H. Chung, and H.-S. Huang 2004), amount of money transferred between consumers and payment receivers (Martens and Provost 2011), rating score of a product (Bennett and Lanning 2007), etc. Although the concepts represented by these various types of weights differ, there are some properties that allow us to group the weights from a modelling aspect into two distinct categories: *preferential* and *measurable* weights.

The first category involves the weighted relationships of the so called preference networks (M. E. Newman 2003). Newman defines preference networks as bigraphs, where one set of nodes represents individuals (or groups) and the other set of nodes are entities (e.g. events, interest in books or movies, etc.) with which people are involved. The link weights in this case, called *preferential* weights, indicate a strength of the user preference and are very distinct for each individual. They are closely connected with the interests, tastes or emotions of the person and therefore can not be objectively quantified. As examples of preferential weights we consider link weights that represent ranking scores, opinions, trust, etc. Data about this type of weights are typically collected with online reviews, interviews and questionnaires (Wasserman 1994).

Unlike the preferential weights, the second category includes weights that can be explicitly measured and do not reflect subjective concepts, such as personal attitudes. Within this group we categorise link weights that represent concepts like amount of money, frequency of visits, invested time, data flow, etc. The *measurable* weights do not have a biased connotation associated with them and they can be directly measured through observations or the use of archival records like browsing history, bank statements, detailed call records, etc (Wasserman 1994). For the rest of the chapter we make the assumption that the greater values of the measurable bigraph weights signify a stronger connection to the top node. If this is not the case and the higher values are associated with a larger distance (e.g. link weights that represent concepts like the time needed for finishing a request, distance, cost, etc.), they can be converted to the former type by for example, calculating the inverse of the distance (M. E. Newman 2001c), subtracting the distance from an upper bound like the maximum link weight plus one (Brandes 2008) or calculating a negative exponent from the link weight (Brandes 2008). In some cases, the link weights can also represent a personal estimate of a measurable value. For instance, in our running example the link weights could signify how often, according to the person, she or he visits a place. Although this is a case where the weights reflect a person's opinon and are therefore clearly subjective, we categorise them in the second group since the underlying variable (frequency of visits) is still directly measurable.

4.2.3 *Preferred properties*

In this section, we discuss the intuition behind what type of methods would be suitable for each of the framework stages. We try to formalize the discussion into a set of preferred properties that articulate the properties which we want our methods to comply with. Then, based on this formulation, we propose a range of methods for every stage.

TYPE I PROPERTIES (STAGE I OF THE FRAMEWORK) The first set of properties captures the intuition on how to combine the bigraph link weights b_{ik} and b_{jk} into an aggregated link weight l_{ijk} on a level of a module (see Figure 4.1.1). We refer to the similarity between the bottom nodes based on the bigraph link weights as link weight similarity.

Intuition behind property 1.1: When are two bottom nodes *i* and *j* that share a common top node *k* most similar, given the bigraph link weights b_{ik} and b_{jk}? In the case of preferential datasets, two entities are similar if they both have similar connotations associated with the top node (e.g. both users liked or both users disliked a movie). Thus, if both preferential link weights equal the highest value or they both have the lowest value in the range, there will be a maximal similarity between the bottom nodes. For the measurable weights, the intuition is not so straightforward. We reason that when two bottom nodes "invest" more resources (e.g. time, amount of money) in the top node, they have more in common and are therefore more similar to each other. For instance, in our running example of persons visiting locations, a frequent visit to the same place increases our confidence that the visitors are similar in some domain (e.g. by age, social status or proximity of living). On the other hand, we do not have the means to draw conclusions on whether two persons that rarely visit the restaurant are similar or dissimilar in some manner. There can be various motives for not visiting the location often, which may or may not be the same for the two persons.

Preferred property 1.1 (Maximal similarity): *Given a weighted bigraph defined between two bottom nodes i and j and a shared top node k, the combined weight l_{ijk} has a maximum value when both links b_{ik} and b_{jk} have (a) either the maximal or the minimal possible value for preferential datasets and (b) the maximal possible value for the measurable weights.*

Intuition behind property 1.2: This property considers the inverse situation from Preferred property 1.1. That is, when are the bottom nodes least similar on a level of the module? In the context of preferential weights, two entities that completely differ in their attitudes (one has a strong positive and the other strong negative perspective of the top node) have a minimal similarity. However, in the case of measurable weights it is hard to define when two bottom nodes are the least similar. That might be the case of a regular customer at the local organic food store (high weight) and another person who visited the store only once and does not like it (low weight). Or it might be the case of two persons that visited the store by a random chance and have nothing in common (low and low weight). Therefore, we do not make a generalisation of this property for the measurable weights.

Preferred property 1.2 (Minimal similarity): *Given a weighted bigraph defined between two bottom nodes i and j and a shared top node k, the combined weight l_{ijk} will have a minimum value when (a) the links b_{ik} and b_{jk} have opposite values, meaning that one of the links has the highest possible value and the other one has the lowest possible value for the preferential weights.*

Intuition behind property 1.3: The next property ensures that the aggregated link weight does not depend on the labels of the nodes, but only on the bigraph structure. It is valid for both preferential and measurable weights.

Preferred property 1.3 (Commutative property): *Given a weighted bigraph defined between two bottom nodes i and j and a shared top node k, the combined weight l_{ijk} should be independent of the sequence of the links b_{ik} and b_{jk}, i.e. $l_{ijk} = f(b_{ik}, b_{jk}) = f(b_{jk}, b_{ik}) = l_{jik}$.*

Intuition behind property 1.4: With this property we reason about whether the similarity between the bottom nodes would remain the same, if we multiply the weights by a constant c ($c > 1$). This is clearly not the case for the preferential weights. As an illustration, let us look at an example of people rating movies on a scale from 1 (very bad) to 10 (very good). If two users gave similar bad scores to a movie of 1 and 3, the similarity between them should be higher than between two users that gave a bad score of 3 and a very good score of 9 (note that the pairs of scores are proportional with coefficient c=3). Similarly, the property is valid for the measurable weights too.

Preferred property 1.4 (Multiplication): *Given a weighted bigraph defined between two bottom nodes i and j and a shared top node k, the following inequality is valid for both the preferential and measurable weights: $f(b_{ik}, b_{jk}) \neq f(c \cdot b_{jk}, c \cdot b_{jk})$, where $c > 1$ and $b_{ik} \neq b_{jk}$.*

TYPE II PROPERTIES (STAGE II OF THE FRAMEWORK) This stage embodies the calculation of the top node (topological) weight s_k, which was already discussed in Section 3.2.1. Since we already covered the topic in the previous chapter, we will not further elaborate on the intuition behind this type of methods in this section.

TYPE III PROPERTIES (STAGE III OF THE FRAMEWORK) The third group of axioms formalises the intuition of how the topological and the link weights should be combined together. More specifically, we calculate a component weight c_{ijk} that merges both the top node weight s_k and the aggregated link weight l_{ijk} for each module in the bigraph (see Figure 4.1.3). Note that both weights s_k and l_{ijk} should be scaled to the same range, so that the component weight is not dominated by one of the variables. We discuss this further in Section 4.3.

Intuition behind property 3.1: Quite intuitively, the value of the component weight should be the highest when the bottom nodes are most similar regarding both the topology and the link weights.

Preferred property 3.1 (Maximal similarity) *Given a weighted bigraph defined between two bottom nodes i and j and a shared top node k, the component weight c_{ijk} will receive* *the highest value if both the topological weight s_k and the aggregated link weight l_{ijk} are maximal.*

Intuition behind property 3.2: Contrarily from our previous property, the component weight will have the lowest value when the bottom nodes are considered least similar in terms of both the link weights and the topology of the bigraph.

Preferred property 3.2 (Minimal similarity) *Given a weighted bigraph defined between two bottom nodes i and j and a shared top node k, the component weight c_{ijk} will receive the lowest value if both the topological weight s_k and the aggregated link weight l_{ijk} are minimal.*

TYPE IV PROPERTIES (STAGE IV OF THE FRAMEWORK) The last group of axioms describes the intuition of how the w_{ij} weight in the projection should be calculated, by aggregating the component weights over all shared top nodes between i and j (see Figure 4.1.4).

Intuition behind property 4.1: Again, it is intuitive that the weight w_{ij} in the projection will demonstrate the largest similarity between the bottom nodes, when all the component weights have the highest values.

Preferred property 4.1 (Maximal similarity) *Given a weighted bigraph defined between two bottom nodes i and j and one or more shared top nodes 1,2,...,k, the weight w_{ij} in the projection will be maximal if all the individual component c_{ij1},c_{ij2},...,c_{ijk} weights have the highest values.*

Intuition behind property 4.2: In the opposite manner from Preferred property 3.1, the nodes i and j can be considered as least similar when all the component weights have the lowest values.

Preferred property 4.2 (Minimal similarity) *Given a weighted bigraph defined between two bottom nodes i and j and one or more shared top nodes 1,2,...,k, the weight w_{ij} in the projection will be minimal if all the individual component c_{ij1},c_{ij2},...,c_{ijk} weights have the lowest similarity.*

Intuition behind property 4.3: Since the component weights represent positive similarity, with every added weight the tie strength between the nodes becomes stronger.

Preferred property 4.3 (Additivity: More component weights create stronger ties) *Given a weighted bigraph defined between two bottom nodes i and j and multiple shared top nodes 1,2,...,k, the weight w_{ij} in the projection increases with every added component weight ($f(c_{ij1}, c_{ij2}, ..., c_{ijk_1}) < f(c_{ij1}, c_{ij2}, ..., c_{ijk_1}, c_{ijk_2})$ where $k_1 < k_2$).*

4.3 Methodology

As discussed previously in Section 4.2.1, our node classification framework for weighted bigraphs uses existing unigraph relational learners over the weighted projection of the bigraph. The similarity weights w_{ij} in the projection preserve more information about the link weights and the topology of the bigraph. We presented an theoretical approach in Section 4.2.3 and we elaborated on the intuition behind each of the framework stages. In this section, we propose specific methods for the different stages that follow the formulated properties. This certainly by no means is the full set of possible choices, but it is a first step into creating more representative projections. Further research can include different methods for the stages and by mixing and matching compose new combinations for determining the weights.

4.3.1 *Framework for node classification within weighted bigraphs through projection*

CALCULATING THE LINK WEIGHTS Guided by the properties from Section 4.2.3, we propose a range of functions for calculating the aggregated link weight l_{ijk} in Table 4.1. This weight denotes the similarity between the bottom nodes i and j as demonstrated by the bigraph link weights b_{ik} and b_{jk}. As can be seen from the table, there is a variety of methods, most of which are based on the distance $d = |b_{ik} - b_{jk}|$ between the link weights (listed in Table 4.1). In Tables 4.2 and 4.3, we present how well the proposed methods comply with the properties.

The first three methods listed in Table 4.1 have an exponential - like shape that converts the distance d between the bigraph link weights into a similarity score, where the nodes with "closer" weights receive a higher score. The following Gaussian function (Vert, Tsuda, and Schölkopf 2004) has an additional parameter σ which fine-tunes the shape of the function to the specific dataset under study. The choice of possible σ parameters for this study creates a shape that resembles the previously discussed functions, with the additional advantage that the function slope can be further adjusted. The best parameter is chosen based on the predictive performance (the area under the ROC curve) on a held-out validation set for each dataset separately. The following two methods (Beta and Dirichlet distribution) are closely connected and defined by Equation 4.1, where $\sum_{i=1}^{K} x_i = 1$ and $0 < x_i < 1$. In case of two parameters (K=2), the Equation simplifies to the probability density function (PDF) of the Beta distribution (Forbes et al. 2011), which we employ as a function of the distance d. Unlike the previously discussed methods, the beta function can have a very adaptable shape to the specific dataset, which does not necessarily correspond to the intuition stated with the properties and thus provides more freedom. Lastly, the Dirichlet distribution with three parameters (where K=3 in Equation 4.1) is a function which is not bounded by our preferred properties and is completely defined by the dataset under study. The Dirichlet distribution is a an extension of the beta function for multiple variables and we use it as a function from the bigraph link weights b_{ik} and b_{jk}.

Function Name	Formula	Parameter range
Inverse distance	$l_{ijk} = \frac{1}{d+1}$	-
Squared inverse dist.	$l_{ijk} = \frac{1}{d^2+1}$	-
Exponential distance	$l_{ijk} = exp(-d)$	-
Gaussian function	$l_{ijk} = exp(-\frac{d^2}{2 \cdot \sigma^2})$	$\sigma = [2^{-3}, 2^3]$
Beta distribution	$l_{ijk} = B(\alpha, \beta, \frac{d - min(d)}{max(d) - min(d)})$	$\alpha = [0.1 : 0.2 : 1.3]$ $\beta = [1 : 2 : 13]$
Dirichlet distribution	$l_{ijk} = Dir(dir1, dir2, dir3, b_{ik}, b_{jk})$	$dir1, dir2, dir3 = [1 : 0.5 : 5]$

Table 4.1.: Methods for calculating the link weights. The Gaussian function, Beta and Dirichlet distribution have parameters which fine tune the shape of the functions for the specific dataset under study.

$$f(x_1, x_2, ..., x_K; \alpha_1, \alpha_2, ..., \alpha_K) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} x_i^{\alpha_i - 1} \tag{4.1}$$

$$\Gamma(n) = (n-1)! \tag{4.2}$$

calculating the topological weights In the previous chapter, we have assessed a range of methods for calculating the topological weights. Based on the empirical results, we chose to apply the hyperbolic tangent function (see Table 3.1 from Chapter 3). This function down-weights the nodes with a high number of connections (large node degree d_k) as less discriminative for the target variable. By all means, any of the other previously proposed functions in Table 3.1 for calculating the top node weights can also be used in this step of the framework.

		Preferred properties			
		Maximal sim.	Minimal sim.	Commutative property	Multiplication
Methods	Inverse distance	✓	✓	✓	✓
	Squared inverse dist.	✓	✓	✓	✓
	Exponential distance	✓	✓	✓	✓
	Gaussian function	+	+	✓	✓
	Beta distribution	+	+	✓	✓
	Dirichlet distribution	+	+	✗	✓

Table 4.2.: An overview of the methods from Stage I and the properties they satisfy for the preferential weights. The parametrized functions (Gaussian function, Beta and Dirichlet distribution) satisfy the minimal and maximal similarity properties for a specific set of parameters.[2]

		Preferred properties			
		Maximal sim.	Minimal sim.	Commutative property	Multiplication
Methods	Inverse distance	✓		✓	✓
	Squared inverse dist.	✓		✓	✓
	Exponential distance	✓		✓	✓
	Gaussian function	+		✓	✓
	Beta distribution	+		✓	✓
	Dirichlet distribution	+		✗	✓

Table 4.3.: An overview of the methods from Stage I and the properties they satisfy for the measurable weights. The parametrized functions (Gaussian function, Beta and Dirichlet distribution) satisfy the maximal similarity property for a specific set of parameters.[3]

COMBINING THE LINK AND THE TOPOLOGICAL WEIGHTS Once we have calculated both the aggregated link weights l_{ijk} and the topological (top node) weights s_k, we can combine them together into a component weight c_{ijk} using the methods listed in Table 4.4. One can assume that both types of weights equally well demonstrate the similarity between the nodes and thus assign the same importance to both of them. We look at two alternatives for achieving this, by either multiplying or summing the topological and the link weight. On the other hand, it might also be the case that one type of weight is more discriminative for the target variable than the other and therefore, should get a higher importance. For this, we introduce two parametrized functions that can be fine tuned to reflect the importance of the weights for each dataset individually. The first option is using a function with a variable parameter α in the range [0,1]. In case when the parameter is set to one, the function is flexible enough to only consider the link weight as important. In the opposite case of alpha being set to zero, only the topological weights will be taken into consideration. Any other value in the range takes into account both types of weights, with a value of 0.5 assigning an equal importance to both. The last proposed method is the previously discussed Dirichlet function (see Equation 4.1), which depending on the chosen parameters can take any arbitrary shape best suited for the specific dataset. Table 4.5 looks at the properties of the methods and examines whether they satisfy the previously discussed properties from Section 4.2.3. The multiplication and the summation are aligned with the intuition, whereas the adaptable Dirichlet and Alpha

[2] With the specific range of parameters used in the study, the Gaussian function satisfies the maximal and minimal similarity properties (denoted with + in Table 4.2).

functions are guided by the problem under study and do not necessarily follow the theory.

Function Name	Formula	Parameter range
Alpha parameter	$c_{ijk} = \alpha \cdot l_{ijk} + (1 - \alpha) \cdot s_k$	$\alpha = [0 : 0.25 : 1]$
Multiplication	$c_{ijk} = l_{ijk} \cdot s_k$	-
Summation	$c_{ijk} = l_{ijk} + s_k$	-
Dirichlet distribution	$c_{ijk} = Dir(dir1, dir2, 1, l_{ijk}, s_k)$	$dir1, dir2 = [1.5 : 0.5 : 4.5]$ $\quad dir3 = 1$

Table 4.4.: Methods for combining the link and the topological weights into a component weight.

		Preferred properties	
		Maximal sim.	Minimal sim.
Methods	Alpha parameter	+	+
	Multiplication	✓	✓
	Summation	✓	✓
	Dirichlet distribution	+	+

Table 4.5.: An overview of the methods from Stage III and the properties they satisfy.

AGGREGATING THE COMPONENT WEIGHTS The final similarity weight in the projection (w_{ij}) is created by combing all the component weights c_{ijk} together. We propose two methods for this task in Table 4.6, but other options could be used as well. The first method summation, simply sums all the component weights. The second method called maximum, takes into account only the component weight with the highest value as the total similarity between the nodes. This method does not comply with the preferred properties (see Table 4.7), since they take into account all the component weights and not simply one.

Function Name	Formula	Parameter range
Summation	$w_{ij} = \sum_{k \in N_b(i) \cap N_b(j)} c_{ijk}$	-
Maximum	$w_{ij} = \max_{k \in N_b(i) \cap N_b(j)} c_{ijk}$	-

Table 4.6.: Methods for combining the component weights into projection weight.

[3] With the specific range of parameters used in the study, the Gaussian function satisfies the maximal similarity property (denoted with + in Table 4.3).

	Preferred properties		
Methods	Maximal sim.	Minimal sim.	Aditivity
Summation	✓	✓	✓
Max	✗	✗	✗

Table 4.7.: An overview of the methods from Stage IV and the properties they satisfy.

RELATIONAL LEARNERS As discussed previously in Section 3.2.3, the relational classifiers make use of the graph structure in order to obtain probability estimations (scores) for the unknown node labels (Macskassy and Provost 2007). Although any of the classifiers used in the previous chapter can also be applied here, in order to keep the setting simple we decided to apply only the weighted-vote Relational Neighbor (WvRN) classifier (see Equation 3.3) (Macskassy and Provost 2007). We base our choice on the empirical results from the Section 3.4.3, where this classifier had comparable predictive performance to some more complex classifiers, but was much faster.

4.3.2 *Alternative settings*

In the rest of this section we discuss two other, alternative approaches for node classification within weighted bigraphs.

PROPOSITIONAL LEARNERS As an alternative to our network based approach, node classification can also be considered as a standard classification problem with high-dimensional and highly sparse features. In this setting, we represent the weighted bigraph with an adjacency matrix as explained in Section 4.2 (see Figure 2.1). To this matrix, where we consider the top nodes as features, we apply an SVM with a linear kernel using the libLINEAR toolbox (Fan et al. 2008b) [4]. For massive and sparse datasets, our network based framework has a scalability advantage as it only considers the neighbouring nodes instead of the full training set for classification (see Section 3.1.1). This means that for our previous bigraph of people visiting locations, we only need to calculate the similarity between the persons that visited the same locations (are connected in the projection). The similarity to all other nodes is set to zero. In the case of an SVM, the large dimensionality of the data would require

[4] This propositional learner was also used in Chapter 3 as a benchmark technique.

some sampling and dimensionality reduction such as singular value decomposition, which scales badly to these settings (Martens, Provost, et al. 2013).

vector similarity measures Lastly, we employ a third approach for classification within weighted bigraphs, where we calculate the similarity between the bottom nodes by using existent vector similarity metrics. To the resulting projection, we apply a weighted *k*-nearest neighbour technique or the wvRN classifier. This setting is closely related to *k*-nearest neighbour classification in memory based collaborative filtering, where bigraphs of users that rated products are used for recommending new products (see Section 4.6). Within this approach, every bottom node is represented as a vector $\overrightarrow{x}$ of size m, where m denotes the total number of top nodes in the bigraph. An element at position *i* in the vector $\overrightarrow{x}$ is equal to zero if there exist no connection between the bottom node and the *i*-th top node in the bigraph. Otherwise the element is equal to the strength (weight) of the connection. The similarity between the bottom nodes can be determined by using any vector similarity function for numerical data that takes the weights into consideration (some options are listed in (Cha 2007)). These similarity functions do not necessarily need to follow the stated intuition from Section 4.2.3. In our study we employ two vector similarity functions, namely the cosine similarity and the extended Jaccard similarity function for numeric data as defined in the work of Provost et al. (Provost, Martens, and Murray 2015b) (see Table 4.8). These two measures take into account only the bigraph link weights when calculating the similarity between the nodes. Additional information about the top node weights are included in the weighted versions of the metrics.

Once the weights in the projection are calculated, we apply a weighted *k*-nearest neighbour technique (Provost and Fawcett 2013). This is similar to the wvRN classification, with the difference that we only take into account the *k* most similar bottom nodes. The wvRN on the other hand, exploits the graph representation of the data to include all the neighbours of the node for classification. We choose the number of considered nodes *k* in *k*NN for each dataset experimentally on a separate validation set. Since the datasets are very sparse, with each node having up to only a few hundred neighbours, the search range of *k* is within [1,10,100]. Additionally, we also apply a wvRN classifier to the weighted projection.

[5] $\overrightarrow{z} = \overrightarrow{x} \circ \overrightarrow{y}$
$z_i = x_i \times y_i$
$x \cdot y = \sum_{i=1}^{m}(x_i \times y_i)$
$||x||_1 = \sum_{i=1}^{m} |x_i|$

Function Name	Formula
Cosine similarity (unweighted)	$\frac{\overrightarrow{x}_{bool,1} \cdot \overrightarrow{x}_{bool,2}}{\Vert\overrightarrow{x}_{bool,1}\Vert_1 \cdot \Vert\overrightarrow{x}_{bool,2}\Vert_1}$
Weighted Cosine similarity (link weights)	$\frac{\overrightarrow{x_1} \cdot \overrightarrow{x_2}}{\Vert\overrightarrow{x_1}\Vert_1 \cdot \Vert\overrightarrow{x_2}\Vert_1}$
Weighted Cosine similarity (combined weights)	$\frac{(\overrightarrow{w} \circ \overrightarrow{x_1}) \cdot \overrightarrow{x_2}}{\Vert\overrightarrow{w} \circ \overrightarrow{x_1} \circ \overrightarrow{x_1}\Vert_1 \cdot \Vert\overrightarrow{w} \circ \overrightarrow{x_2} \circ \overrightarrow{x_2}\Vert_1}$
Jaccard similarity (unweighted)	$\frac{\overrightarrow{x}_{bool,1} \cdot \overrightarrow{x}_{bool,2}}{\Vert max(\overrightarrow{x}_{bool,1}, \overrightarrow{x}_{bool,2})\Vert_1}$
Weighted Jaccard similarity (link weights)	$\frac{\Vert min(\overrightarrow{x_1} \circ \overrightarrow{x}_{bool,2}, \overrightarrow{x_2} \circ \overrightarrow{x}_{bool,1})\Vert_1}{\Vert max(\overrightarrow{x_1} \circ \overrightarrow{x}_{bool,2}, \overrightarrow{x_2} \circ \overrightarrow{x}_{bool,1})\Vert_1}$
Weighted Jaccard similarity (combined weights)	$\frac{\Vert min((\overrightarrow{x_1} \circ \overrightarrow{x}_{bool,2}) \circ \overrightarrow{w}, (\overrightarrow{x_2} \circ \overrightarrow{x}_{bool,1}) \circ \overrightarrow{w})\Vert_1}{\Vert max((\overrightarrow{x_1} \circ \overrightarrow{x}_{bool,2}) \circ \overrightarrow{w}, (\overrightarrow{x_2} \circ \overrightarrow{x}_{bool,1}) \circ \overrightarrow{w})\Vert_1}$

Table 4.8.: Vector similarity metrics [5].

4.4 Datasets

The datasets used in this chapter were obtained from several different sources: the MIT Reality Mining Project,[6], the Yahoo! Webscope Program [7], the Koblenz Network Collection (KONECT)[8], the PAKDD'15 Data Mining Competition [9] and the Belgian Government. They have a clear weighted bigraph structure and for each one of them a target variable is available for prediction. Descriptive statistics regarding the weighted bigraph datasets are provided in Table B.2 in Appendix B.

Several of the collected datasets include bigraphs with preferential weights, that mainly represent some type of user ratings. For instance, the MovieLens [10] and the Yahoo Movies datasets (Koenigstein, Dror, and Koren 2011) are collections of movie ratings from users of the websites MovieLens.org and Yahoo. From these data, we define bigraphs where the bottom nodes represent the users and the top nodes represent the movies. The variable that we are predicting in both cases is the gender of the users. LibimSeTi (Brozovsky and Petricek 2007) is a dataset of profile ratings from users of the Czech social network LibimSeTi.cz. Similarly, the available target variable for prediction is the gender of the users. Another bigraph with preferential weights can be defined from the BookCrossing dataset, which contains information about book ratings from the website Bookcrossing.com (Ziegler et al. 2005). Our task in this case is to predict the age of the users.

[6] http://realitycommons.media.mit.edu
[7] http://webscope.sandbox.yahoo.com/
[8] http://konect.uni-koblenz.de
[9] http://www.pakdd2015.jvn.edu.vn/
[10] http://www.grouplens.org

In this chapter, we also consider bigraph datasets with measurable weights. One example is the TaFeng dataset that includes information about transactions in supermarkets (H.-S. Huang et al. 2005), where the customers are connected to the products they bought. For this bigraph, the link weights denote the total amount of money that the customer spent on the given product (price × amount) and the target variable is the age of the customers. Another network was created from the PAKDD data science challenge data, with the goal of predicting the gender of users in an e-commerce setting. The bigraph in this setting is constructed from customers and the categories of products they browsed online. Each link weight accounts for the number of products from a given category that were viewed by the customer. Furthermore, we also used mobile phone usage data from a microlender company in order to assess the creditworthiness of the loan applicants (see Chapter 6). The bigraph in this case is defined between the loan applicants and the people they have called to or they have been called by, with the link weights denoting the number of calls. The last dataset is composed of transactional logs between companies situated in Belgium and abroad, where the link weights represent the amount of money transferred between the companies (see Chapter 5). The goal is to predict whether a company fraudulently resides outside of Belgium for tax benefits.

Since the variance of the datasets with measurable weights can be large, we need to scale the data properly so that some data do not dominate in the classification task. As a preprocessing step, we standardize the data by substantiating the mean weight at a level of a bottom node and dividing it with the standard deviation. For the datasets with preferential weights, we scale the data into a uniform range of [1,10].

4.5 Results

In this section we look at the predictive performance of all previously discussed methods from the three different settings (Section 4.3). Additionally, in Appendix B (Section B.1) we also provide an assessment of the run-time performances. Within our framework, we look at every combination of techniques from the different stages, which leads to a total of 60 unique combinations. We also asses the performance of applying the propositional learner over the weighted and unweighted adjacency matrix, as well as using the various vector similarity metrics in a combination with a weighted *k*NN technique or a relational classifier. Our experimental setting includes running a 10 fold cross validation procedure over every dataset for each method, similarly to the setup from Chapter 3. For the parametrised methods, we carefully select the optimal parameters on a held-out validation set. In cases when a method has more than one parameter that needs to be fine tuned, we apply a grid search technique on three levels (see Section 3.2.5). Hence, instead of brute-force searching the space for the optimal parameters, we reduce the number of iterations by searching more closely around the best parameters from the previous level. This reduces the required run-time significantly and at the same time provides equal granularity of the solution. All the experiments in this study are conducted on a 3.40 GHz Intel i7 CPU, with 8 GB RAM and a 64-bit operating system.

4.5.1 *Predictive performance*

The predictive performance results for all methods in terms of AUC (Area Under Curve) (Fawcett 2006) are presented in Table B.1 in Appendix B. Similarly to the previous chapter, we use the Kemeny-Young method (Conitzer, Davenport, and Kalagnanam 2006; Young and Levenglick 1978) to rank the techniques (see Section 3.4). For statistical comparison of the results we use the Wilcoxon signed rank test (Demšar 2006) to compare the results of the best performing method to all other classifiers. Unfortunately, the number of classifiers in this study is much larger than the number of dataset used, so we can not employ a more sophisticated test for comparison of multiple classifiers (e.g. Friedman test with a Nemenyi post-hoc test (Demšar 2006)). Performances that are not significantly different at a 5% confidence level (according to a Wilcoxon signed rank test (Demšar 2006)) are tabulated in bold face. In our results, significant differences at the 1% level are emphasized in italics, and differences at the 5% but not at the 1% level are reported in normal script [11]. For two combinations of techniques from our framework that include the Dirichlet distribution for determining both the link and the combined weights, we do not calculate and report the results. This is due to the long time needed for fine tuning the large number of parameters.

As can be seen from Table B.1 in Appendix B, the highest ranked techniques present a combination of methods from our framework that consider both the topological and the link weights for classification. The best performing technique uses the adaptable beta function to calculate the link weights and later combines them with the topological weights using another adjustable function, namely the Dirichlet distribution. As an aggregation step, the combined weights of all shared nodes are summed together into a final weight in the projection. The typical shape of the beta function for this combination is aligned with the intuition, that nodes with "closer" link weights are more similar. The common shape of the Dirichlet distribution in this combination is fairly balanced and gives an equal importance to both the topological and the link weights. The highest ranked technique that only considers the link weights is the Gaussian function in combination with the sum of shared nodes. This technique, as well as all the other combinations that only add information about the link weights to the projection, perform better than the unweighted projection. This proves that there is indeed information in the bigraph link weights that should be utilized for node classification. On the other hand, the results of the propositional learners are only average, with similar ranking for both cases when the SVM is applied over the weighted or the unweighted adjacency matrix. Moreover, the

[11] One of the best performing techniques from the previous chapter (tanh-ssh-wvRN), that is used as a baseline technique that only considers the topological weights is marked with **.

Table 4.9.: Kemeny - Young ranking per method on all datasets.

Kemeny Ranking	Link weight	Component weight	Aggregation func.
1	**Gaussian**	**Dirichlet**	**Sum of Shared Nodes**
2	**Beta**	**Alpha param.**	Max
3	*Inverse distance*	**Times**	
4	**Exp. distance**	*Sum*	
5	*Sq. inverse distance*	-	
6	**Dirichlet**		

performance of the vector similarity techniques is weak, with some combinations scoring even worse than the unweighted projection.

We now look at the predictive performance of the techniques from each of the three different settings in turn.

PREDICTIVE PERFORMANCE OF THE FRAMEWORK The predictive performance of the methods from the first stage of the framework (creating the link weights) are summarized in Table B.1. Although the results are statistically inconclusive and we can not claim that one technique works best over all the combinations, by looking at the rankings from Table B.1 and the summary Table B.1 in Appendix B we can say that applying the parametrized Gaussian and Beta functions yields good performance. The Gaussian function has a similar shape as the other three non parametrized functions (Inverse distance, Squared inverse distance and Exponential distance), where the lower values of d are associated with higher similarity. An additional advantage of the Gaussian function is that it is flexible enough to tune the slope depending on the dataset under study. The beta function on the other hand, is not bounded by the preferred properties and can take any arbitrary shape. Although the shape of the beta function in our experiments mainly follows the same shape as the previously discussed Gaussian function, the form of the function for two datasets (Fraud and Loans) is completely the opposite, with the higher values of d signifing larger similarity. This turns out to make the combination of methods that include the beta function, the best performing technique for both datasets (see Table B.7 in Appendix B). The highest flexibility is ensured with the use of the Dirichlet similarity, as a function of both link weights b_{ik} and b_{jk}. The shape of this function throughout our experiments is completely defined by the specific dataset, with no apparent patterns across different datasets and combinations of techniques.

From the techniques used for combining the topological and the link weights, again the parametrized functions are ranked best (see Table B.1). The best performing technique is the Dirichlet distribution, closely followed by the Alpha function. The shape of the Dirichlet function is dictated by the domain of use, with most cases having a shape that assigns similar importance to both the topological and

Rank	Link weight	Top node weight	Component weight	Aggregated weight
1	**Beta**	**Hyperbolic tangent**	**Dirichlet**	**Sum of Shared Nodes**
2	Gaussian function	-	-	Sum of Shared Nodes
3	-	Hyperbolic tangent	-	Sum of Shared Nodes
4	-	-	-	Unweighted projection

Table 4.10.: Comparison of the best performing techniques that consider only the topological weights, only the link weights, a combination of the previous two types of weights or applying a relational classifier [11] directly over the unweighed projection. The results show that combining information about both the topology and the bigraph link weights results in significantly better performance [12].

the link weights. The Alpha function is also adaptable and covers more options than the remaining two functions, Times and Summation. In fact, when $\alpha = 0.5$ the Alpha function turns into Summation. If we look at the alpha coefficients in our experiments, it seems that the topological weight is more important for the preferential datasets (in most combinations the alpha parameter is equal to zero and occasionally to 0.25). On the other hand, the value of the alpha parameter does not seem to contain a typical value for the measurable datasets. Finally, the ranking from Table B.1 shows that considering only one type of weight (not combining the link and the topological weights) or no weight yields the weakest performance.

The results from the last stage of the framework are again listed in Table B.1 and clearly show that summing the weights of all shared nodes performs significantly better than only considering the shared node with the highest weight, since more information is included in the projection.

Lastly, in Table 4.10 we include a comparison of the best combinations of techniques from our framework that consider only the topological weights, only the link weights, a combination of the previous two types of weights or applying a relational classifier directly over the unweighed projection. The results are favourable to our thesis that adding information about the bigraph link weights in the projection results in better classification: the combination of both weights outperforms the cases when solely one type of weight is considered or no information is added to the projection (unweighted projection).

PREDICTIVE PERFORMANCE OF THE ALTERNATIVE SETTINGS The vector similarity techniques from our third setting are ranked poorly in comparison to all

[11] We use WvRN as a relational classifier in all four cases.

[12] Although the combination consisted of the Beta, Hyperbolic tangent, Dirichlet and Sum of Shared Nodes functions performs best from the combining techniques, as we discuss later on, due to the large number of parameters that need to be tuned for this combination, we recommend experimenting with other combinations of functions that have a smaller number of parameters.

the other techniques (see Table B.1 in Appendix B). In Table B.4 in Appendix B we summarize the average predictive performance for all the vector similarity metrics, where the weighted cosine similarity that combines both the link and topological weights is ranked best for both cases when a *k*NN or a WvRN is applied. As can be seen from Table B.5 in Appendix B, the results are better when all the neighbours are considered for training instead of the *k* closest nodes. However, the difference in the rankings is not significant, since the datasets are very sparse with each bottom node having only up to few hundred neighbours and the typical value of *k*=100 is usually enough to cover all the neighbours. In addition, the Cosine similarity measure seems to perform better than the Jaccard similarity function (Table B.6 in Appendix B), which is in line with the results from the previous chapter and the work of Provost et al. (Provost, Martens, and Murray 2015b). For both measures, the weighted versions that take into account both the topological and the link weights are ranked better.

Unfortunately, due to the limited number of datasets from each category (4 datasets with measurable weights and 4 datasets with preferential weights), currently we can not measure significant differences in the performance of the methods for the two types of datasets. Therefore, we can not draw conclusions about what works best for each data type. We do, however, report the best performing techniques for each dataset in Tables B.7 and B.8 in Appendix B.

4.6 Related Literature

The graph mining literature regarding bigraphs in general, is limited (see Section 3.5). There are two lines of research that either focus on designing methods and metrics applicable directly to the bigraph or they employ existing unigraph techniques over the one-mode representation of the bigraph (projection). In both cases, the techniques are generally designed for the simple type of unweighted graphs, where the links are homogeneous. Therefore, for the weighted graphs the general practice is to transform them into unweighted versions by either ignoring the link weights or creating binary relationships based on some threshold (M. Newman 2010; Wasserman 1994). This leads to a loss of potentially valuable information that can help us better understand the complex networks under study.

Most of the previous work that considers techniques applicable directly to the weighted bigraphs is primarily focused on collaborative filtering for recommender systems (Y. Hu, Koren, and Volinsky 2008; H.-S. Huang et al. 2005; Koenigstein, Dror, and Koren 2011; Linden, Smith, and York 2003; McFee et al. 2012; Ziegler et al. 2005). In collaborative filtering, the bigraphs are defined between users and the items they have explicitly rated or for which they have provided some implicit feedback, like the number of views, purchases or click-throughs (Y. Hu, Koren, and Volinsky 2008; X. Su and Khoshgoftaar 2009). Unlike the link prediction techniques that search for the links that are most likely to emerge in general, the collaborative filtering techniques look for the new user-item connections in the bigraph which are most likely to occur for a given user. The vector similarity based approach that we consider as an alternative node classification solution to our framework is closely connected to the so called memory-based collaborative filtering techniques. These memory-based techniques calculate the similarity between the users (or the items (Linden, Smith, and York 2003)) using vector similarity measures (e.g. Pearson correlation, Cosine similarity) that take into account the bigraph weights. Some studies have also included information about the topology of the graph, by considering a cosine similarity measure weighted by the inverse frequency of the top nodes degree (Breese, Heckerman, and Kadie 1998), which significantly improves the results. In the next step, the ratings of the *k* nearest neighbours of a user are used for identifying the potentially interesting items. For more details about collaborative filtering, we refer to the following studies (Adomavicius and Tuzhilin 2005; X. Su and Khoshgoftaar 2009).

Node classification differs from collaborative filtering in an important way - instead of looking for new associations between the nodes, the task of node classification is to predict a certain attribute of the nodes. Research on this topic for weighted bigraphs has so far been restricted to only a few studies. For instance, the work of Provost et al. (Provost, Martens, and Murray 2015b) considers bigraphs of mobile users connected to the locations they have visited (through the IP addresses), with the aim of targeting mobile ads to the users. Each link in the bigraph is weighted by the frequency of location visits. The authors use several vector similarity metrics (including the ones listed in Table 4.8) to estimate the similarity between the users and weight the unigraph projection. To asses the metrics and see which one estimates the real resemblance between the nodes best, they look at the average lift (how many more positive cases are there than expected by a random chance) amongst the closest *k* neighbours of a node. In line with our findings, the weighted versions of the metrics that take into account the link weights, perform better than the unweighted metrics. The rankings of the metrics however, do not show that combining information about both the topology and the link weights of the bigraph always performs best. In a similar vain, Provost et al. (Provost, Dalessandro, et al. 2009) use bigraphs of browsers (as proxy for users) related to the social networks pages they have visited to find an audience that is potentially interested in a brand. The link weights of this bigraph denote the frequency of visits to the webpages. Using again vector similarity techniques (some of which take into account the link weights), the authors identify the potential consumers by selecting the ones with the highest resemblance to the known brand consumers. Since there is no single similarity metrics that ranks the potential audience better than all the other metrics, the authors suggest using the scores from these metrics as features into a higher level model. Another study by Perlich and Provost (Perlich and Provost 2006) considers classification for datasets that include high-dimensional categorical features, such as the locations that a person visited, identifiers of the previously bought products by customers and etc. These data can be seen as weighted bigraphs (or *k*-partite graphs in general), that can be aggregated using aggregation operators and added as new features in a traditional propositional model.

Our framework for node classification within weighted bigraphs is based on graph proximity measures that utilize the topology and the link weights of the graph in order to determine the similarity between the nodes and weight the projection. For the same problem, Opsahl [13] has proposed including the bigraph link weights in the projection by summing the weights directed towards the common top nodes. This results in directed weighted projections, where the link weight from one node to another in the projection is not necessary the same as in the opposite direction. Additionally, he proposes including information about the top node degree in the projection, by multiplying the topological weights with the previously described directional link based weights. Several studies have also proposed new graph-proximity measures for calculating the similarity between the nodes in weighted graphs, for the task of link prediction. For this task, a similarity score is calculated for the pairs of nodes that are not yet connected in the graph. Then, the ones with high similarity are identified as the pairs that are likely to be connected in the future. Although the graph-proximity measures from the following studies have been applied to unigraphs, we look at them into more detail since they can also be decomposed within our framework and used for creating the weights in the bigraph projection. The difference is that instead of calculating the similarity between every pair of unconnected nodes, in our setting we would calculate the similarity only between the bottom nodes. The work of Murata and Moriyasu (Murata and Moriyasu 2007, 2008) considers a link prediction task in a network of users from the Q&A website Yahoo! Chiebukuro, with the goal of identifying potential answerers to the users questions. The network is essentially a projection of a bigraph defined between users and pages , where the weights signify the number of posts (questions/answers) on the page. In the aforementioned studies, the authors propose multiplying the average of the bigraph link weights to the topological score on a module level and then summing the scores. Their findings include better link prediction results when information about the link weights is included. Similarly, Lü and Zhou (Lü and Zhou 2010) employ the same measures on three different datasets for link prediction and report inconsistent results. In their study, the link weights did not always contribute to better predictions. The authors elaborate on these results by linking them to the social network theory by Granovetter (Granovetter 1973) on weak ties, which states that the links with low weight can also contain valuable information and they can have an important role in the networks. De Sá and Prudêncio (De Sá and Prudêncio 2011) have looked at a weighted network of authors, connected if they collaborated on a paper. Again, this is a bigraph projection with the weights denoting either (i) the total number of co-authored papers or (ii) a weighted sum of the co-authored papers, where the papers with more authors get down-weighted. The authors extended several combinations of methods for calculating node similarity for weighted networks, mainly by multiplying the sum of the bigrph links to the topological weight. The similarity scores obtained from these methods are then used as features for supervised methods. Unfortunately, the results from the study are inconclusive of whether it is beneficial to consider the link weights.

[13] http://toreopsahl.com/2009/05/01/projecting-two-mode-networks-onto-weighted-one-mode-networks/

Similarly to our work, Gupte and Eliassi-Rad (Gupte and Eliassi-Rad 2012) take an axiomatic approach for the problem of how the weights in the projection should be created, but for the case of unweighed bigraphs. They define a set of axioms which approximate the intuition and examine how well the existing weighting measures in literature satisfy this characterization.

4.7 **Conclusion**

In this chapter, we build upon our work from Chapter 3 and propose a multi-stage framework for node classification within weighted bigraphs. The results show that by including information about the bigraph links, we can create more representative projections and thus improve the prediction results. The framework is build systematically, by framing the intuition behind every stage in a set of preferred properties that guide the design of the proposed methods. Although based on the results, we can not claim that any of the methods works best in every domain, we can name several combinations that have solid performances. From the first stage, we would recommend experimenting with the parametrized Gaussian and Beta functions, that can be adapted to the dataset under study. Since the results clearly show that the best predictive performance is achieved when information about both the topology and the link weights is included in the projection (better than considering only one type of weight or no weight at all), we propose that the weights are combined with the flexible Alpha function. Even though the Dirichlet function had better predictive performance in this stage, it has more parameters that need to be tuned and is thus much slower. In the aggregation stage, the results show that including information about multiple modules (summation in our case), provides better results.

The predictive performance results of the alternative settings, similarly to the prior studies that tackle this problem, were inconclusive of whether it is beneficial to consider the link weights. On the other hand, we provided an extensive assessment over multiple datasets, where the best AUCs on every dataset were achieved by also including the link weights. The framework we propose in this chapter is by no means the only way that the bigraph weight information can be utilized for classification. Other possible settings can be explored in future research.

B

Appendix

Table B.1.: Kemeny-Young ranking for all the combinations of techniques.

Link weight	Top node weight	Component weight	Aggregated weight	Classifier	Rank
Beta	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	**8.6**
Gaussian function	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	**12.4**
Beta	Tanh	Alpha param	Sum of Shared Nodes	WvRN	**14.1**
Beta	Tanh	Times	Sum of Shared Nodes	WvRN	**14.2**
Dirichlet similarity	Tanh	Alpha param	Sum of Shared Nodes	WvRN	*15.7*
Gaussian function	Tanh	Times	Sum of Shared Nodes	WvRN	**16.4**
Inverse distance	Tanh	Alpha param	Sum of Shared Nodes	WvRN	*18.1*
Sq. inverse distance	Tanh	Alpha param	Sum of Shared Nodes	WvRN	*18.6*
Gaussian function	Tanh	Alpha param	Sum of Shared Nodes	WvRN	*18.6*
Inverse distance	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	**19.1**
Beta	Tanh	Dirichlet	Max	WvRN	19.3
Exp distance	Tanh	Alpha param	Sum of Shared Nodes	WvRN	*20.3*
Gaussian function	Tanh	Dirichlet	Max	WvRN	*21.8*
Exp distance	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	**21.9**
Sq. inverse distance	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	**23.3**
Beta	Tanh	Times	Max	WvRN	*24.1*
Gaussian function	Tanh	Times	Max	WvRN	*25.6*
Inverse distance	Tanh	Alpha param	Max	WvRN	*25.8*
Gaussian function	Tanh	Alpha param	Max	WvRN	*25.8*
Sq. inverse distance	Tanh	Alpha param	Max	WvRN	*26.4*
Beta	Tanh	Alpha param	Max	WvRN	*26.6*
Inverse distance	Tanh	Dirichlet	Max	WvRN	*27.6*
Exp distance	Tanh	Times	Sum of Shared Nodes	WvRN	**27.8**
Exp distance	Tanh	Alpha param	Max	WvRN	*27.8*
Gaussian function	Tanh	Sum	Sum of Shared Nodes	WvRN	28.0
Inverse distance	Tanh	Times	Sum of Shared Nodes	WvRN	**28.8**
Exp distance	Tanh	Dirichlet	Max	WvRN	*28.9*
Sq. inverse distance	Tanh	Dirichlet	Max	WvRN	*30.1*
Gaussian function	-	–	Sum of Shared Nodes	WvRN	30.6
Dirichlet similarity	Tanh	Sum	Sum of Shared Nodes	WvRN	*31.1*
Sq. inverse distance	Tanh	Times	Sum of Shared Nodes	WvRN	**31.6**
Dirichlet similarity	Tanh	Times	Sum of Shared Nodes	WvRN	*31.6*
Inverse distance	Tanh	Sum	Sum of Shared Nodes	WvRN	32.3
Sq. inverse distance	Tanh	Sum	Sum of Shared Nodes	WvRN	32.9
Beta	Tanh	Sum	Sum of Shared Nodes	WvRN	33.5
Dirichlet similarity	Tanh	Alpha param	Max	WvRN	*33.8*
Beta	-	–	Sum of Shared Nodes	WvRN	*33.9*
Inverse distance	Tanh	Times	Max	WvRN	*35.3*
Exp distance	Tanh	Times	Max	WvRN	*35.3*
Dirichlet similarity	-	–	Sum of Shared Nodes	WvRN	35.4

Continued on next page

Table B.1 – *Continued from previous page*

Link weight	Top node weight	Component weight	Aggregated weight	Classifier	Rank
Exp distance	Tanh	Sum	Sum of Shared Nodes	WvRN	35.6
	Tanh		Sum of Shared Nodes	WvRN	*36.9***
				SVM unweighted	37.9
Inverse distance	-	–	Sum of Shared Nodes	WvRN	37.9
Gaussian function	Tanh	Sum	Max	WvRN	38.4
				SVM weighted	*38.6*
Exp distance	-	–	Sum of Shared Nodes	WvRN	38.8
Sq. inverse distance	Tanh	Times	Max	WvRN	*39.1*
Sq. inverse distance	-	–	Sum of Shared Nodes	WvRN	41.2
Dirichlet similarity	Tanh	Sum	Max	WvRN	*41.5*
Gaussian function	-	–	Max	WvRN	41.6
Inverse distance	Tanh	Sum	Max	WvRN	42.0
Dirichlet similarity	Tanh	Times	Max	WvRN	42.6
Sq. inverse distance	Tanh	Sum	Max	WvRN	43.4
			Cosine similarity weighted	kNN	43.5
Beta	-	–	Max	WvRN	44.7
Exp distance	Tanh	Sum	Max	WvRN	46.3
			Cosine similarity	kNN	47.1
Exp distance	-	–	Max	WvRN	49.3
Inverse distance	-	–	Max	WvRN	49.4
Beta	Tanh	Sum	Max	WvRN	50.1
Dirichlet similarity	-	–	Max	WvRN	50.1
Sq. inverse distance	-	–	Max	WvRN	53.4
			Cosine similarity weighted	WvRN	53.5
			Cosine similarity	WvRN	54.4
			Jaccard similarity weighted	WvRN	57.2
			Jaccard similarity	WvRN	58.0
				Unweighted pro.	58.1
			Jaccard similarity weighted	kNN	63.9
			Jaccard similarity	kNN	65.1

Dataset	Target Label	l_0	l_1	$n_\top$	$n_\perp$	m	$k_\top$	$k_\perp$	k	δ
Loans	default	632	152	51,069	677	57,546	1.13	85.00	2.22	0.0017
MovieLens	gender	273	670	1682	943	100,000	59.45	106.04	76.19	0.0630
Yahoo Movies	gender	2,206	5,436	11,915	7,642	221,330	18.57	28.96	22.63	0.0024
PAKDD	gender	3,297	11,703	383	15,000	20,323	53.06	1.35	2.64	0.0035
Fraud	fraud	30,479	62	7,495	30,541	73,753	9.84	2.41	3.88	3.22e-04
TaFeng	age	17,330	14,310	23,719	31,640	723,449	30.5	22.86	26.14	9.64e-04
Book-Crossing	age	21,709	24,542	145,457	46,251	301,470	2.07	6.52	3.14	4.48e-05
LibimSeTi	gender	43,510	57,606	135,359	101,116	13,594,717	100.43	134.45	114.98	9.93e-04

Table B.2.: Descriptive statistics of the weighted bipartite datasets: class distribution (l_0, l_1), number of top ($n_\top$) and bottom ($n_\perp$) nodes, number of edges (m), average degree for top ($k_\top$) and bottom ($k_\perp$) nodes, average combined degree (k) and density ($\delta(G)$).

Rank	Method
1	**SVM weighted**
2	**SVM unweighted**

Table B.3.: Ranking of the predictive performance when using a linear SVM on the weighted and unweighed adjacency matrix of the bigraph.

Rank	Vector Similarity Metrics	Classifier
1	**Cosine similarity combined weights**	**kNN**
2	**Cosine similarity combined weights**	**WvRN**
3	**Cosine similarity link weights**	**WvRN**
4	**Jaccard similarity combined weights**	**WvRN**
5	*Cosine similarity link weights*	*kNN*
6	**Jaccard similarity**	**kNN**
7	**Cosine similarity**	**kNN**
8	**Cosine similarity**	**WvRN**
9	**Jaccard similarity link weights**	**WvRN**
10	**Jaccard similarity**	**WvRN**
11	*Jaccard similarity combined weights*	*kNN*
12	*Jaccard similarity link weights*	*kNN*

Table B.4.: Average rankings for the vector similarity techniques in a combination with a weighted k-nearest neighbour classifier or WvRN based on their predictive performance.

Rank	Method
1	**WvRN**
2	**kNN**

Table B.5.: Average ranking of the predictive performance for the combinations of methods that include weighted k-nearest neighbour classifier or WvRN.

Method	Avg Ranking
1	**Cosine similarity combined weights**
2	**Cosine similarity link weights**
3	**Cosine similarity**
4	**Jaccard similarity**
5	**Jaccard similarity combined weights**
6	*Jaccard similarity link weights*

Table B.6.: Average ranking of different vector similarity techniques based on the predictive performance.

Dataset	Link weight	Top node weight	Component weight	Aggregated weight	Classifier	AUC
Loans (Default)	Beta	-	–	Sum of Shared Nodes	WvRN	0.6337
	Beta	Tanh	Alpha param	Sum of Shared Nodes	WvRN	0.6337
Fraud (Incoming)	Beta	Tanh	Times	Sum of Shared Nodes	WvRN	0.8151
	Beta	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	0.8151
Ta Feng (Amount of products)	Beta	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	0.6853
PAKDD (Number of views)	Dirichlet similarity	Tanh	Alpha param	Sum of Shared Nodes	WvRN	0.8160

Table B.7.: Best combinations of methods per dataset with measurable weights.

Dataset	Link weight	Top node weight	Component weight	Aggregated weight	Classifier	AUC
MovieLens (Gender)					SVM weighted	0.7833
Yahoo Movies (Gender)	Exp distance	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	0.8218
Book Crossing (Age)	Gaussian function	Tanh	Dirichlet	Sum of Shared Nodes	WvRN	0.6528
LibimSeTI (Gender)					SVM weighted	0.8726

Table B.8.: Best combinations of methods per dataset with preferential weights.

B.1 Run-time performance

In Figures B.1 and B.2 we plot the average duration of the techniques over the datasets with less than 100,000 bottom nodes. Note that for the large LibMISeTi dataset, the Jaccard similarity measure is not able to run and it takes very long time to tune the parameters of the Beta and Dirichlet distributions. In Figure B.1, we plot all the combinations of functions that include the sum of the shared nodes as an aggregation function, and compare the run-time performance to the techniques from the second and third setting. For each of the combining function (x-axis) we plot the time performance of the link similarity functions. The setting in Figure B.2 is similar, with the difference that the considered aggregation function is the maximum of the shared nodes.

The run-time performance of the techniques in both Figures is similar. As we saw earlier, although the performance of the parametrized functions in general is better, fine tuning the parameters is computationally expensive. This is valid for the combining techniques, where the Alpha and the Dirichlet functions are much slower than the other, as well as on a link weight level, where the Gaussian, Beta and the Dirichlet are much slower than the non-parametrized metrics. Intuitively, the number of parameters that need to be tuned influences the time performance of the technique. In order to make the parameter estimation procedure less computationally intensive, one can consider a more coarse grid search with only one or two levels, which would result in reduced predictive performance. Another time improvement that can be considered is incorporating knowledge about the parameters, so that the the range of values that need to be evaluated is narrowed.

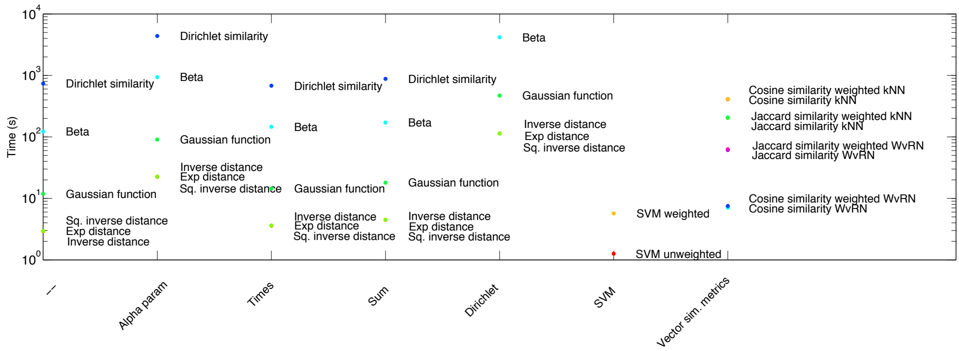


Figure B.1.: Comparison of the time performance for different methods. The aggregation function used in the framework is the sum of shared nodes.

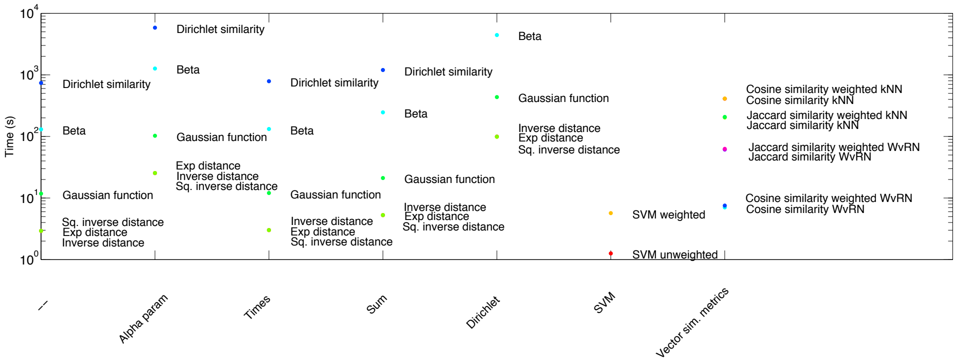


Figure B.2.: Comparison of the time performance for different methods. The aggregation function used in the framework is the maximum of shared nodes.

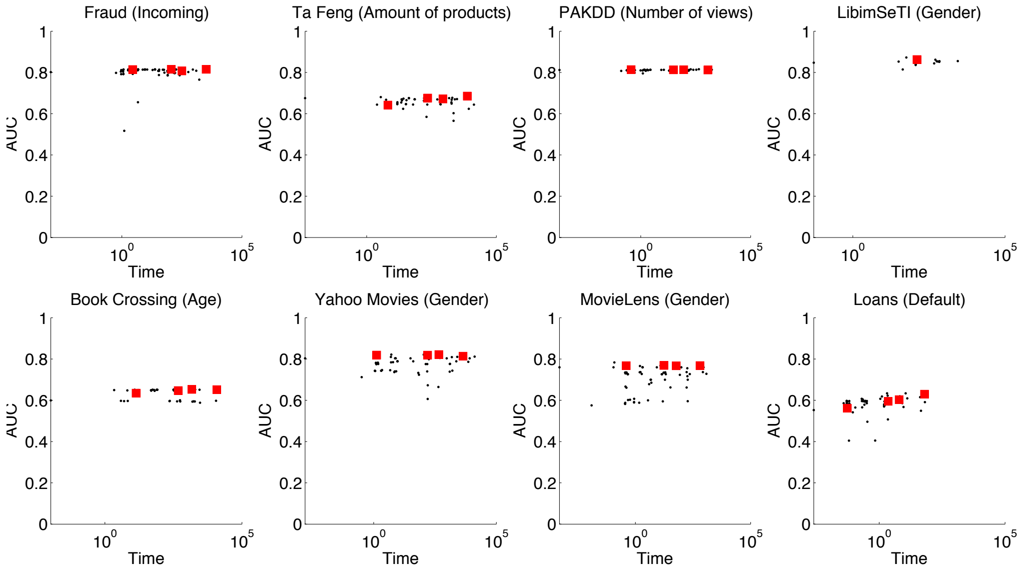


Figure B.3.: Predictive and run-time performance for all the techniques per datasets. The performance of the recommended combinations is highlighted in red.

Part III

APPLICATIONS

5

Corporate residence fraud detection

"Fraud is as Belgian as beer and fries."

former State Secretary for Fraud in Belgium, John Crombez (2013)

With the globalisation of the world's economies and ever-evolving financial structures, fraud has become one of the main dissipaters of government wealth and perhaps even a major contributor in the slowing down of economies in general. Although corporate residence fraud is known to be a major factor, data availability and high sensitivity have caused this domain to be largely untouched by academia. The current Belgian government has pledged to tackle this issue at large by using a variety of in-house approaches and cooperations with institutions such as academia, the ultimate goal being a fair and efficient taxation system. This is the first data mining application specifically aimed at finding corporate residence fraud, where we show the predictive value of using both structured and fine-grained invoicing data. We further describe the problems involved in building such a fraud detection system, which are mainly data-related (e.g. data asymmetry, quality, volume, variety and velocity) and deployment-related (e.g. the need for explanations of the predictions made).

5.1 Introduction

The social contract (Rousseau 1762) between governments, citizens and corporations comprises the mutual agreement between these parties on how to allocate resources for common expenses such as road construction, hospitals and the environment. Most democratic societies have implemented this social contract in the form of a taxation system in which each party agrees to contribute to the total expenditure of the country. Needless to say, the success of such a system depends on the fairness and efficiency and thus the compliance of all actors to the system in place. Falsifying or withholding information in order to limit the amount of tax liability is therefore against the law and constitutes fiscal (or tax) fraud. This is a large-scale problem that affects a multitude of entities: the public sector, the private sector and citizens (National Fraud Authority 2013). Fiscal fraud exists in several forms, which can broadly be categorized as evasion of direct (income and corporate tax) and indirect (VAT) taxes. Governments are a frequent target of fraudsters, who undermine the system and abuse its benefits, grants and tax programs.

In Belgium, fiscal fraud is acknowledged as a significant problem. The former State Secretary for Fraud in Belgium even stated that *"Fraud is as Belgian as beer and fries"* (Crombez 2013). Estimations by the European Commission show that the Belgian government loses about €30 billion annually due to fiscal fraud, which corresponds to 6% of its GDP (Crombez 2013), placing Belgium's among the highest fraud rates in Western Europe. On a larger level the overall European losses due to tax evasion and avoidance are estimated to be an astonishing €1 trillion (European Commission 2013). These numbers show that the fight against fraud is a crucial aspect of any fiscal system. Not only does fraud cause serious damage to society, it also has a direct financial impact on individuals. The relevance of fraud detection in the current climate of severe fiscal consolidation and social hardship is motivated in the declaration of the G20 leaders of September 2013. In this statement, they emphasize the importance of ensuring that all taxpayers pay their fair share of taxes as well as the need to tackle tax avoidance, harmful practices and aggressive tax planning (Organisation for Economic Co-operation and Development 2013).

Since most tax systems use audits to ensure compliance with tax laws, an accurate selection of the most likely fraudulent cases is crucial to maintain an efficient tax inspection. Given this urgent need to identify specifically the most suspicious cases, the Belgian government joined forces with academia to work on automated data mining systems that look for fraud patterns in large amounts of data to detect corporate residence fraud. This type of fraud occurs when companies deceitfully attempt to place their residency in a low-tax country in order to avoid paying the higher taxes of their real location. The data consists of two types of records: on the one hand we have structured data on the Belgian companies (sector, city, etc), on the other hand we have transactional data (invoicing logs) between Belgian and foreign companies [1]. Although using this fine-grained transactional data can be tricky, the information that could be retrieved from it is very valuable in order to detect residence fraud. Consider the following (fictitious) example: let's say we see that a foreign company receives invoices from a golf club in Brussels. This could be an indication that the company and its owner(s) likely reside in Belgium. If this is indeed so, other foreign companies that also receive invoices from this specific golf club make for interesting suspects. Working at such a fine-grained identifier level makes available very informative data (Perlich and Provost 2006).

The potential of data mining techniques has also been acknowledged by governmental entities, including the Belgian government. In their action plan to strengthen the fight against tax fraud, the European Commission articulates it as follows: *"For tax administrations, the development and full use of automated tools and risk management techniques would release human and budgetary resources to concentrate on achieving targeted objectives."* (EUR-LEX 2012).

The rest of the chapter is structured as follows: In the next section, a literature overview is given on the importance of fraud detection, the different types of fraud, and the main domain challenges. Section 5.3 looks deeper into the type and size of the data and Section 5.4 describes the specific methods that were used. Section 5.5 shows the results and the deployment, with concluding remarks in Section 5.6.

5.2 Literature overview

5.2.1 *The Importance of Fraud Detection*

As discussed above, the Belgian government is a frequent target of fraudsters. Abuse of the tax system is a very costly fraud type (National Fraud Authority 2013), with estimates of losses going into the billions of euros (dollars, pounds) for the EU, US and UK governments. Translating these numbers to impact on members of society is an easy exercise. For instance, Belgian estimates reveal that fraud against the public sector is estimated to be €30 billion per year and thus directly costs every adult in the country about €2,700 annually.

As mentioned before, the elementary form of damage from fraud in government-allotted resources is an immediate financial loss and thus the unfair redistribution of wealth. Note, however, that the consequences can be much broader. Fraud losses could result in cuts to thinly spread government-budgets, tax increases, less investment in the public sector (such as new roads, hospitals, schools, etc.) and

[1] In other words, our aim is to discover which of the companies that are registered abroad (marked as foreign companies), are actually falsifying their country of residence for tax benefits. For illustration, these transactional data can be represented as bigraphs, where the top nodes denote the domestic companies and the bottom nodes represent the foreign companies for which we need to decide whether they are fraudulent or not. We will come back to this with more details in Section 5.4.

eventually a slower economy altogether. Effective fraud detection, on the other hand, can lead to many benefits. Not only is there the direct impact of recovering parts of the loss of capital, increased effectiveness can also lead to enhanced deterrence (Baer 2008). That is, the increased likelihood of being captured, causes the net expected benefit from the fraudulent activities to be outweighed by their (proportionally increased) expected costs, thus decreasing the appeal of such fraud. Needless to say, governments try hard to cope with ever-more creative fraud-schemes such as the ones addressed in this project.

5.2.2 *Data Mining for Fraud Detection*

In the literature, data mining has been applied to many domains for fraud detection. Some of them include the banking sector for discovering fraudulent credit card transactions or card applications (Bhattacharyya et al. 2011; Brause, Langsdorf, and Hepp 1999; Juszczak et al. 2008; Sánchez et al. 2009; Whitrow et al. 2009), identifying fraudulent service subscriptions or calls in the telecommunications domain (Cortes, Pregibon, and Volinsky 2001; Fawcett and Provost 1996, 1997; Hilas and Mastorocostas 2008), detecting false insurance claims (Phua et al. 2010), revealing websites with high level of non-intentional traffic for online advertising (Stitelman et al. 2013) or uncovering tax evasions in the public sector (Basta et al. 2009; Gonzlez and Velsquez 2013; R.-S. Wu et al. 2012) and etc. A comprehensive overview of the complete fraud detection literature is beyond the scope of this chapter and for more details we refer to the following studies (Bolton and Hand 2002; Ngai et al. 2011; Phua et al. 2010).

Many of the fraud detection studies need to deal, similarly to our work, with heterogeneous types of data and especially large amounts of transactional data. The applications are mainly in the banking and the telecommunications sectors, where the companies keep logs of card transactions and calls. Due to the high dimensionality of the transactional data, a very common approach in the literature is to perform some type of aggregation over the transactional data. One way to do so is to create transaction aggregates for each user account that characterize the typical legitimate behaviour of the user (Bolton and Hand 2001; Fawcett and Provost 1997). Any new transaction that deviates from the typical behaviour of that user would be suspected as fraudulent. Other studies (Bhattacharyya et al. 2011; Juszczak et al. 2008; Whitrow et al. 2009), take the approach of deriving RFM (Recency, Frequency, Monetary Value) attributes from the original features over a period of time. The RFM attributes are then used as inputs for a classification technique. Aggregating the transactions creates new structured data and loses the fine-grained information that is included in the transactions (cf., the golf club example from Section 5.1).

To our knowledge, there have been only a few prior studies that take into account information from the very high-dimensional categorical attributes, especially the so called identifier attributes as described by Perlich and Provost (Perlich and

Provost 2006). These identifier attributes can represent particular features such as company accounts, names of locations or persons and etc. The work of Fawcett and Provost (Fawcett and Provost 1996, 1997) incorporates such attributes by first searching for individual classification rules based on the transaction-level data (such as location in cell phone calls), and then building higher-level features based on these rules. The studies by Brause et al. (Brause, Langsdorf, and Hepp 1999) and Sanchez et al. (Sánchez et al. 2009) include these attributes by using classification based on association rules, which is only applicable to smaller datasets. On the other hand, Cortes et al. (Cortes, Pregibon, and Volinsky 2001) and Stitelman et al. (Stitelman et al. 2013) apply relational inference on the networks defined among persons that call each other (Cortes, Pregibon, and Volinsky 2001) and among the browsers connected if they visit the same website (Stitelman et al. 2013). Our work explores and combines fraud data on both levels: we apply scalable algorithms to extract fine-grained knowledge from large amount of transactional data and we also consider the structured data. By doing so, we are able to harness the predictive power of both types of data, as well as the added value of combining them.

For the purpose of tax evasion, data mining has been applied to the problem of corporate fraud (Cecchini et al. 2010; Kirkos, Spathis, and Manolopoulos 2007), where companies falsify their financial statements, as well as Value Added Tax (VAT) evasion (Basta et al. 2009; Gonzlez and Velsquez 2013; R.-S. Wu et al. 2012), solely on structured data.

5.2.3 *Domain Challenges*

Typical challenges encountered when applying data mining techniques in the domain of corporate residence fraud detection relate to positive label scarcity and quality. Additionally, due to the way in which the data is generated nowadays, we also encounter problems related to Big Data with respect to size (volume), type (variety) and speed of data generation and stationarity-violation (velocity). Furthermore, the acceptance by stakeholders of the resulting models is highly dependent on their comprehensibility, which needs to be taken into account both during and after the modelling phase.

Data scarcity: Fraud data are usually *highly unbalanced*, as there are many more non-fraudulent instances than the number of fraudulent ones. Furthermore, limited resources and the very expensive labeling procedure (auditing) further bias the class balance. Moreover, one often encounters pollution of the data labels: data instances can have wrong labels if a fraudulent instance has not yet been discovered and therefore is marked as a legitimate one. Additionally, very little structured data is available on the foreign companies (except for the country where they are located).

Volume, variety and velocity: Every quarter, the government receives millions of tax data entries containing hundreds or even thousands of transactions as well as structured data on each of the companies involved in these entries. As such, not only do the datasets have very large *volume*, the size also continues to increase. Even so, this is not the only issue related to *velocity*. Fraudsters are known to change the way in which they commit fraud in progressively more creative and covert ways to evade the detection systems in place. This adversary effect requires continuous back-testing and updating of the models because stationarity assumptions might be violated. Needless to say, when taken as a whole, the datasets coming from our domain need fast algorithms that can cope with these challenges.

As mentioned before, the government receives both tax declarations as well as transactional data. Furthermore, the government has a database with additional information on each of these companies. Ideally, one wants to connect all these various bits of information in order to obtain the best predictions. Unfortunately, it is not trivial to do so in a sensible way. For instance, how could one combine transactional logs (e.g., foreign company FC_1 transferred money to a golf club) with a geographical location? Possible answers include hierarchical modelling, ensemble methods and stacking; clearly, this situation opens up many possible paths of model combination and design. To the best of our knowledge, we are the first to propose a solution for this corporate governance problem.

Comprehensibility: The success of a tax fraud detection system depends on more than accurately flagging suspected cases. Each suspected case is sent to an investigator who determines whether it is indeed fraudulent and collects evidence. As each investigator develops his/her own expertise on tax fraud, this expertise can conflict with the predictions. If investigators receive many cases they see as clearly non-suspect, they might reject the prediction system altogether. When the system however explains why a case is flagged as suspect, investigators can quickly determine whether this is in line with their experience or not. Further, in a confirmed match situation, the explanation provided by the system can serve as a starting point for the actual investigation. Thus a model that is comprehensible at the instance/decision level is critical both to get user acceptance and to speed up the manual investigation.

5.3 Data

Before we can dig into the modelling approaches for this domain, we must first discuss the exact data available to us. Although we received data from various sources, we can discern two main types of records. First we have invoicing records between 2,745,478 Belgian companies and 873,640 foreign companies (*transactional data, T*). Second, we also have structured information on each of the Belgian companies (*structured data, S*).

Transactional data: In terms of transactional invoicing data, we can distinguish between two types of invoices: incoming invoices from foreign companies to Belgian

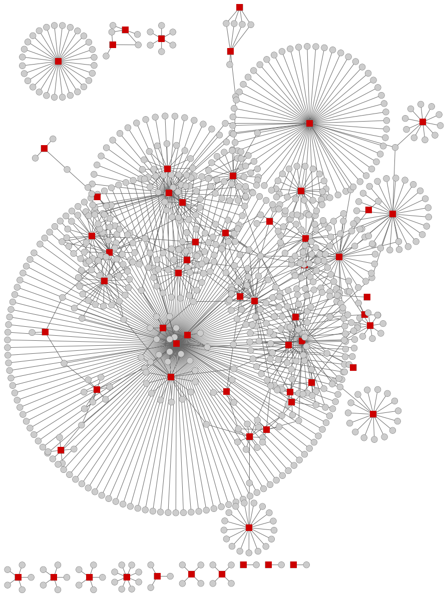


Figure 5.1.: The structure of the invoicing network based on the incoming and outgoing transactions between the fraudulent foreign companies (red squares) and the Belgian companies they interact with (grey nodes). As can be seen from the big cluster, many of the fraudulent foreign companies are connected to the same Belgian companies.

Table 5.1.: Statistics for the three invoice datasets: Incoming (Inc.), Outgoing (Out.) and Incoming and Outgoing (Inc. and Out.)

	Inc.	Out.	Inc. and Out.
Number of transactions	251,198	6,551,512	6,802,710
Number of unique transactions	73,753	1,955,912	2,029,641
Number of Belgian accounts	7,495	107,345	108,753
Number of Foreign accounts	30,541	858,131	858,703
Average number of transactions per Belgian account	9.84	18.22	18.66
Average number of transactions per foreign account	2.41	2.28	2.36

companies, and outgoing invoices from Belgian to foreign companies. We engineered three different datasets from these invoice logs: a dataset of incoming invoices, a dataset from outgoing invoices and a third dataset where we merged both the incoming and outgoing invoices. Additional statistics for the datasets are shown in Table 5.1. There can be multiple transactions between two companies, on different dates or with different amounts of money. Hence in Table 5.1, both the total number of transactions and the number of unique transactions between the companies are shown. The latter counts only the transactions where the sender/recipient pair is

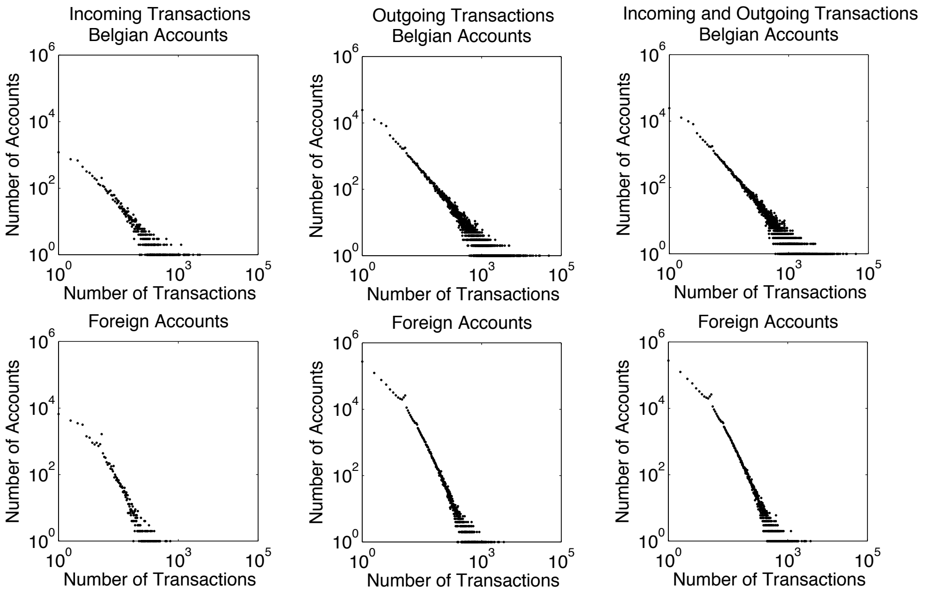


Figure 5.2.: The number of unique transactions per account for the invoicing datasets, when considering the Belgian (top) and foreign (bottom) companies. Most of the Belgian companies typically send or receive invoices to only few foreign companies and, vice versa, most of the foreign companies interact with only few Belgian companies.

unique. Note that this transactional data can be represented both as a matrix and as a bipartite graph. In the *matrix* representation each row i corresponds to a foreign company; column-values indicate whether the foreign company made a connection to resident company j, with entry $x_{i,j}$ equal to 1 and 0 otherwise. A subset of the bigraph containing all of the fraudulent nodes is visualized in Figure 5.1 with red squares representing the fraudulent foreign nodes and the grey nodes representing Belgian companies they interact with. Figure 5.2 shows the degree distributions (number of transactions) of the Belgian-foreign bipartite company networks.

These graphs help us to understand the power of the fine-grained data in the modeling results presented below. Although keeping the full fine-grained data instead of aggregate values can be tricky to work with, previous studies have shown fine-grained transaction data to enhance the predictive power of models (Junqué de Fortuny, Martens, and Provost 2013; Martens and Provost 2011; Perlich and Provost 2006). This is partly due to the fat tail in the degree distribution we see in Figure 5.2: many companies appear related to only very few other companies, but these low-connectivity companies make up the vast majority of the companies. Thus, it is relatively difficult to compress the company-related information into a small number of simple aggregate variables (that do not obscure the fine-grained connectivity information). Figure 5.1 in turn shows that the fraudulent foreign companies indeed do seem to interact with the same Belgian companies, as illustrated by the big cluster in which most fraudulent companies are found. Thus, it makes sense to intelligently—i.e., in a supervised fashion—examine the specific companies in the predictive modeling (note that it is informative both to be connected to one or more suspicion-inducing companies as well as to be connected to one or many suspicion-reducing companies; cf., (Perlich and Provost 2006)).

Structured data: Most of the available structured information is on residential companies because, to date, there is still no sharing of information between governments. This asymmetry in data is one of the challenges to overcome on the level of policy making. For each of the Belgian companies we have information on their geographical location, industry type, start-up date, etc. For foreign companies, we only know in which country they are located as well as the target label. As shown in Figure 5.3, we can infer certain aggregate characteristics for the foreign companies, based on what the average Belgian company that connects to it looks like.[2] For the particular foreign company shown in the figure, we can deduce that its average transaction value is a certain amount and that its usual geographical correspondence location is located in Brussels (median region in Belgium). These characteristics can be added into the input vector in order to augment the prediction information. This set-up leads to a total of 31 features per foreign company.

An important problem that arises in our scenario, due to limited resources and the very expensive labeling procedure, is skewness in the distribution of the target variable. Out of the total 873,640 available foreign companies, only 62 are marked as positive cases. Because of this skewness, we make use of AUC and lift curves and we repeat each of the experiments 10 times on different out-of-sample selections to ensure robustness and the external validity of the results.

5.4 Methods

Given the variety and volume of the data, different feature engineering and modeling techniques are first applied and subsequently combined. In this section we first describe each of the different methods briefly after which we discuss their combination, displayed in Figure 5.4.

5.4.1 *Structured Data*

In the structured learning scenario, we are interested in predicting whether or not a foreign company is fraudulent, based on the aggregate, structured information of the associated resident companies. This turns out to be a classical predictive modeling

[2] Due to the sensitivity of the data, all of the examples given in the figures are only illustrative; aggregate results and statistics are of course computed on the true data.

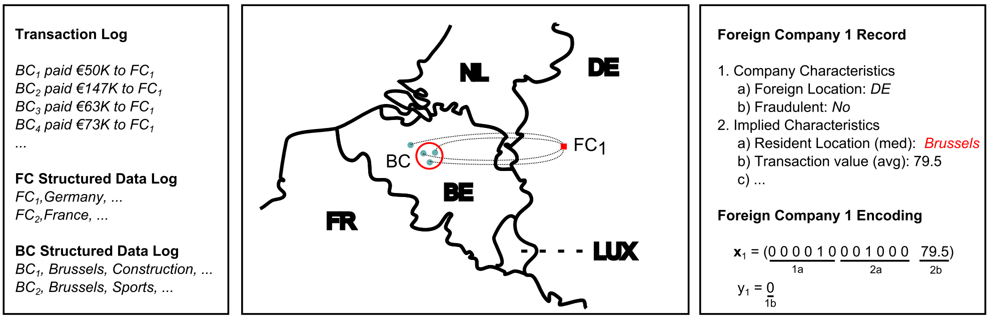


Figure 5.3.: Example of the feature engineering for structured data. The foreign company (FC_1) has many associated Belgian companies (BC_i). Original company characteristics such as the location are combined with implied characteristics such as the average transaction values and the median resident location.

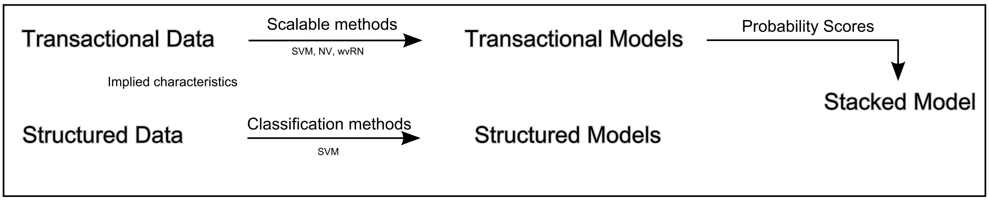


Figure 5.4.: Overview of the system design for fraud detection. In a first step, the transactional data and structured data are engineered into features. Afterwards different modeling techniques are applied to generate both transactional and structured models. These models are evaluated separately at first, but combined in a final step as well into a stacked model.

set-up in which we predict target variable y based on vectors of structured data $\mathbf{x}$, one for each foreign company. To deal with the many-to-one variables, such as location, which appear in the transaction data, we encode them in the structured data via a "weight-of-evidence" encoding; this is a logarithmic transformation that allows one to transform a categorical variable into a variable that is monotonically related to the target variable (L. C. Thomas 2009; L. C. Thomas, Edelman, and J. N. Crook 2002). For example, if most of the Belgian companies connected to a foreign company are located in Brussels, this will be encoded as a one in the position of the dummy-encoded "Brussels" location variable. Examples of structured variables include the location of the linked company (up to town level), the main activity code of the linked company, and the legal construct type of the linked company (with a total of 31 such variables). Once the features have been engineered into a structured input vector, we train an SVM with a linear kernel. SVMs are known to work well with these kinds of data (Sahin and Duman 2011) and the choice of kernel is motivated by the need for comprehensibility of the model (more on this later).

5.4.2 *Transactional Data*

The transactional data can also be represented by vectors as follows: for each of the n foreign companies we look up its previous associations with companies in Belgium. Each of the m Belgian companies is represented by a feature and the value of this feature in the foreign company's m-dimensional vector $\mathbf{x}$ will be equal to one if such a connection was made and zero otherwise. By aggregating all of these vectors we end up with a very high-dimensional, but highly sparse, matrix. There are two main approaches of handling this kind of data: (a) applying propositional learners (such as SVMs and Naive Bayes) on the huge, sparse matrix representation and (b) using relational learning/inference on the graph representation.

PROPOSITIONAL LEARNERS A first approach is to gather all of the data in a big matrix and apply *SVM* (using the LibLinear package (Fan et al. 2008a)). Clearly, due to the size of the data, this will take quite a while on a standard desktop computing set-up. Further, it likely will not perform very well due to class imbalance, as explained by Wallace et al. (Wallace et al. 2011). Indeed, poor performance is revealed in the very low AUC and lift values of this approach (SVM_T, Table 5.1). As a first improvement, we train the SVM on a balanced subset of the data. By equally weighing the number of positive and negative examples, the SVM learns to put equal importance on each of the classes and performs much better (SVM_T(50-50)). Other improvements toward this end, could be to directly optimize for a different loss function (Rudin 2009). An in-depth discussion on this matter is beyond the scope of this application-focused chapter.

In a similar vein, we also apply the *binary Bernoulli Naive Bayes* (NB) which is specifically tailored for massive, sparse, binary data (Junqué de Fortuny, Martens, and Provost 2013). This classifier uses the same input vectors $\mathbf{x}$, but makes an estimate based on the MAP likelihood estimation of a probability parameter for each of the features. These are gathered in a vector with elements $\theta_j = P(X_j = x_{i,j}|C = c)$ and used in a 'naive' model, where all features are assumed to be conditionally independent of each other, given the class, resulting in the following probability estimate for each class (i.e., fraudulent or not):

$$P(C = c|\mathbf{x_i}) \propto \prod_{\mathbf{j=1}}^{\mathbf{m}} (\mathbf{\Theta_j})^{\mathbf{x_{i,j}}} (\mathbf{1} - \mathbf{\Theta_j})^{(\mathbf{1}-\mathbf{x_{i,j}})} \tag{5.1}$$

In this formulation, fraud is encoded by class label C, and the $x_{i,j}$ indicate whether a transaction was made from foreign company j to resident company i. A decision is made by comparing the probability estimate for the fraudulence of the company ($C = 1$) and the non-fraudulence ($C = 0$). The NB modelling procedure does not suffer from the class skew problems of the SVM and the method does not need any further modifications to be run efficiently on the fine-grained data.

RELATIONAL LEARNERS Intuitively, it makes sense to apply a learner that is specifically tailored for the networks resulting from transactions like the ones described in Section 5.3. In order to do so, we must first realize that such transactional logs between two entities (foreign and resident companies) can be represented as a bipartite graph. The visualization already suggests ("two-hop") assortativity in fraud status in the network of foreign companies.

Numerous relational learners exist for graphs with only one type of nodes. In order to make use of them, we can apply the three-step framework for classification within bigraphs proposed in Chapter 3. The idea is to project the bigraph into a unigraph in which foreign companies are connected, based on shared Belgian company connections and then apply a relational learner. Based on the results from Chapter 3, we use a combination of the weighted-voted Relational Neighbor (wvRN) inference method (Macskassy and Provost 2003), together with the sum over the shared nodes (see Table 3.2) and the hyperbolic tangent (*tanh*) top node function (see Table 3.1). This type of top node weighting means that, if say a Belgian company has connections with all of the foreign companies, this company will define a relatively low weight in the total probability calculation. If it does not, it will likely be more informative and thus should be weighted accordingly.

5.4.3 *Stacked model*

Ideally, we want to build a model that incorporates all of the available information. As one can see from the previous sections, it is not trivial to combine these heterogeneous types of data because they require different sorts of models. One way to cope with this problem, while still preserving the variety of modeling approaches is to combine the models in a *stacked model*. The expected efficacy of such a model is explained by the fact that we are incorporating more information into one model than we did before, which should result in a lower modeling bias (Wolpert 1992).

In our scenario, as the stacked model we use a linear SVM to produce a final model that is a linear combination of the output scores of the transactional classifiers and the structured model. An important reason for this particular design is that we do want to keep a maximum level of comprehensibility without sacrificing too much predictive performance. Specifically, the 31 variables of structured data are manageable to a human observer, but the millions of transactions clearly are not. It is much more informative to have the scores of these models encoded as variables—a human inspector can assess the contribution of the network-data component. Should this be high enough, specialized techniques can be used to inspect the underlying reasons for the predictions of the network-data component (as discussed in the next section).

Figure 5.4 summarizes all of the steps required to generate the complete, stacked model. First, the data is converted to (a) transactional (graph) data and (b) structured data. Next, predictive models are built on top of these data, each specifically tailored to cope with the particular aspects of the corresponding data (as explained in the previous section). Lastly, the scores of the graph models are combined with the structured data as input to the final stacked model.

5.5 Results and discussion

5.5.1 *Results*

The results of all of the previously explained methods in terms of predictive power (AUC) are shown in Table 5.2. The best performance for each dataset is denoted in boldface and underlined. Performances that are not significantly different at a 5% confidence level (according to a Wilcoxon signed rank test (Demšar 2006)) are tabulated in bold face. Significant differences at the 1% level are emphasized in italics, and differences at the 5% but not at the 1% level are reported in normal script. A first observation that one can make from this table is that our best models achieve very high AUC values (up to 96.22%). The somewhat high standard deviations on these percentages can be explained by the class imbalance (detecting one more or one fewer example can result in a percentage change of about 10%). Nevertheless, our results do show that our best model (the stacked combination of structured and relational models; SVM_{S+T}) performs significantly better than all of the transactional methods for the incoming and the outgoing data. Although it is still the best performing model for the combination of both data types, the variance is too large to conclude statistical significance at the 5% level using the Wilcoxon test.

Although these results are certainly interesting in terms of global predictive power and ranking ability, we should note that in the specific context of detecting fraudulent companies we are more interested in the lift (how much better than random) when targeting the highest ranking members of the dataset. This is because the fraud analysts investigate the companies deemed to be most suspicious. The lift curves (Figure 5.5) show the clear superiority at the highest percentiles of the models built on transactional data, where they are able to perform up to a few hundred times better as opposed to random targeting. The traditional, structured-data model and the stacked model deliver clear improvements as well, but at the highest percentiles the lifts are not nearly as strong as those resulting from using the fine-grained transaction-based models.

As we motivated previously, we can now observe empirically that the fine-grained information included in the transactional data provides substantial gains for de-

Table 5.2.: Results of different techniques in terms of AUC. Subscript S refers to models based on structured data. Subscript T refers to models based on fine-grained transaction data. Subscript $S+T$ refers to models based on both structured data and transaction data.

	Incoming data		Outgoing data		Combined data	
Technique used	AUC	std.dev.	AUC	std.dev.	AUC	std.dev.
wvRN$_T$	76.74	(±5.87)	77.32	(±6.21)	**94.55**	(±5.26)
Naive Bayes$_T$	76.64	(±5.94)	77.6	(±6.37)	**94.74**	(±5.45)
SVM$_T$	*46.88*	(±12.76)	*56*	(±9.27)	*70.26*	(±12.46)
SVM$_T$ (50-50)	*62.23*	(±21.03)	57.95	(±33.66)	74.85	(±19.97)
SVM$_S$	**82.71**	(±10.52)	**86.34**	(±7.74)	**91.77**	(±8.16)
SVM$_{S+T}$	**85.92**	(±7.48)	**86.44**	(±10.23)	**96.22**	(±4.8)

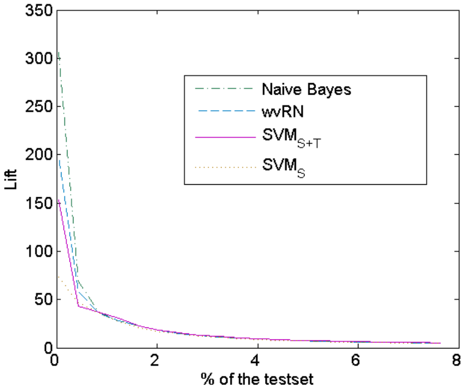


Figure 5.5.: Lift curves of the combined dataset

tecting fraudulent companies. Referring back to our example, the other fraudulent companies that transact with the Brussels golf club receive high transactional fraud scores, and rightly so apparently—as demonstrated by the very high lifts. Once these other foreign companies that transact with these suspicion-conferring Belgian companies[3] are investigated, structured data may still help to find other suspects.

In conclusion, we can say that if one is interested in a global ranking method, the stacked model would be the best design choice in our scenario, whereas the models based on transactional data are better suited for detecting the most likely frauds. The latter result highlights the importance of keeping the fine-grained information as a whole as opposed to only aggregating it into summary variables.

[3] The Belgian companies themselves are not suspicious per se, but the foreign companies that transact with them are.

5.5.2 *Comprehensibility*

In the actual deployment of our model, we have been made aware of the tremendous importance of being able to explain the decisions made. Specifically, the auditors need to understand the exact reasons why classification models make particular decisions. Cases (even if they be few) where the model makes an obvious wrong decision can create disillusionment with the system and reluctance to use it, unless the reasons behind the decision appear to be sound. Therefore, it is essential that the decisions made by the predictive model can be explained; the auditor can decide to over-rule a specific suggestion and confidently move on to the next one. Going back to our running example of a company that has received an invoice from a golf club in Brussels: Although it might be the case that most foreign companies that receive invoices from that entity are indeed fraudsters, a foreign company such as Rolex that has sponsored a golf tournament at this specific golf club (and therefore has also received an invoice) is likely not fraudulently located abroad. So if the explanation for the classification is given (i.e., receiving an invoice from the specific golf club), an auditor can quickly see why it is or is not valid in the context of the particular focal company.

To our knowledge, the distinction between different types of comprehensibility has received relatively little attention in the data mining literature, even though it often is a crucial criterion for final acceptance and increased use of the predictive models. At least two types of explanations exist. Global explanations provide improved understanding of the complete model, and its performance over the entire space of possible instances. Instance-level explanations on the other hand provide explanations for the model's prediction regarding a particular instance. When using transactional data, the total number of variables and/or data values considered by the model (in our case, millions) is much larger than for the typical structured data. Global explanation methods, such as examining the coefficients of a linear model or using a rule-based model, are simply not applicable in such a high-dimensional context. However, an instance-level approach used for document classification (Martens and Provost 2014a), which faces a similar challenge with a large vocabulary, can also be used in this transactional setting: an explanation is defined as the minimal set of entities one received/sent an invoice to, such that removing all the invoices to/from this set changes the predicted class from the class of interest. For our running example, an explanation could be: *'if this company did not receive an invoice from golf club XYZ in Brussels, the predicted class would change to non-fraudulent'*. As such, instance based explanations provide an excellent tool for models that use the fine-grained invoicing data. For more on how explanations can be used both to improve acceptance and also to improve the model itself, as well as further references to related work, we refer to (Martens and Provost 2014a).

Global explanations do still have value, but in a different way. Decision makers need insight into the general methods used by fraudsters and their evolution. One way to do so is to list all variables of the stacked model in order, ranked according to the size of the coefficients in the linear model. Then we could see for example that the country dummies for certain countries are very high on the list, as well as the scores from the transactional models, and certain activity codes. A rule-based model could provide similar insights. These insights may then lead to different sorts of cases being discovered, which then would prime the network models to find similar instances. We are not able to show the actual global explanations, as they involve confidential information.

5.5.3 *Deployment*

In reference to this project, former State Secretary for Fraud John Crombez reported: *"The interaction between the two worlds [academia and government] has proven very valuable. Other countries are now visiting Belgium to see how the Social Intelligence and Investigation Service and the Special Tax Inspection service apply this technique. That is why we need to continue to invest in this technology."* Not only is the predictive performance of our models appreciated, but also considered to be important to success is the fact that in general use this data mining technique can operate on anonymized data, whereby each company is encoded as a "random" number. A company's identity only then needs to be revealed in the context of a particular investigation of a top-suspicion instance. Further, the emphasis on the comprehensibility of the results is deemed essential.

During deployment, the system has to deal with large volumes of heterogeneous data and with new data arriving every quarter, where the underlying data generating process is non-stationary due to the problem being adversarial. The stacked model approach specifically deals with the variety of the data by combining the transactional data from invoices with structural data from tax declarations. The need to retrain the model frequently is facilitated by the scalability of the underlying (naive Bayes and wvRN) methods. They can be run (on a desktop) on the complete data and produce results in a matter of minutes.

5.6 **Conclusion**

In this chapter we have described what to our knowledge is the first data-mining-based method for building a system for detecting corporate residence fraud. The system is based on transactional and structured data, which is gathered by the Belgian government. The success of such a detection system in practice depends on a combination of factors, including efficiency, efficacy and comprehensibility. As such, an important part of our research was to evaluate how one can cope with these conflicting requirements. When used for targeting new fraudsters, a combination of the fine-grained transactional data model and instance-based explanations results in a good trade-off between the needs of an auditor. On the other hand, combining both structured data and fine-grained data in a stacked model is more suited when the main goal is to gain macro-level insights and policy guidance. Given the success of this pilot study, we believe further research into this application to be a logical next step. There are still many opportunities for improvement. Besides simply improving the modeling methods, one particular aspect that we did not touch upon yet is the pro-active gathering of data with active data-acquisition techniques (see e.g., (Macskassy and Provost 2005) for a suspicion-scoring application). It is important to continue to stress the importance of deploying counter-fraud measures for the social good of countries. Although our experiments focus on data from the Belgian government, we hope that researchers from other countries are motivated by our results to apply such methods to or to find better methods for their own countries' data, and/or to convince their governments to do so. It is important for us to understand whether and how data mining indeed can improve government fraud detection efficacy and perhaps even policy making. Once we are convinced, then we can work to remove any lingering doubt or scepticism among decision makers.

This chapter is part of the following publication: Marija Stankova*, Enric Junque De Fortuny*, Julie Moeyersoms, Bart Minnaert, Foster Provost, David Martens, "Corporate Residence Fraud Detection", 2014, KDD 2014 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, New York, USA. My contribution is focused on the bigraph setting of the problem, the literature study and part of the data analysis.

6

Credit scoring in microfinance using alternative data

“The first thing [in credit] is character. Before money or property or anything else.”’

J.P. Morgan

Microfinance has known a large increase in popularity, yet the scoring of such credit still remains a difficult challenge. In general, retail credit scoring uses socio-demographic and credit data. We complement such data with social network data in an innovative manner, i.e. with fine-grained interest and social network data from Facebook. Using a unique dataset of 4,985 microfinance loans from the Philippines, we show how the different data types can predict creditworthiness. A distinction is made between the relationships that the available data imply: (1) look-a-likes are persons who resemble one another in some manner, be it liking the same pages, having the same education, etc. (2) friends have a clearly articulated friendship relationship on Facebook, and finally (3) the “Best Friends Forever” (BFFs) are friends that interact with one another. Our analyses show two interesting conclusions for this emerging application. Firstly, applying relational learners on BFF data yields better results than considering only the friends data. Secondly, the interest-based data that defines look-a-likes, is more predictive than the friendship or BFF data. Moreover, the model built on interest data is not significantly worse than the model that uses all available data, including the friendship data. Hence demonstrating the potential of Facebook data in a microfinance setting.

6.1 Introduction

In microfinance, where credit history data is often lacking, character is considered an important predictor for loan repayment (Schreiner 2003). Manual screening of the applicants by the loan officer is used to gather information about their trustworthiness. Though effective, this is a timely and costly process. Attempts to replace the credit screening process with automated credit scoring have shown that the use of traditional socio-demographic and credit data is insufficient (Schreiner 2000; Van Gool et al. 2012). These types of data are unable to capture the unwillingness to repay the loan, one of the main causes of low repayment rates. Microfinance comes with a social mission of alleviating poverty, enhancing economic development and achieving social impact in the community (Copestake 2007). The creditworthiness decisions should be in line with this social mission. Investing in improved credit scoring models helps microfinance lenders to distinguish the risky population from the target population.

We obtained data from Lenddo, a company specialized in social authentication and scoring technology [1]. Lenddo uses alternative data to provide credit scoring and verification for the emerging middle class in developing markets. The company has developed patented technology to collect and process billions of data points, and uses advanced machine learning techniques to build predictive algorithms. Lenddo has multiple algorithms which draw upon a wide array of data, with user consent, from Facebook, Twitter, LinkedIn, Gmail, Yahoo, Android, IOS, machine fingerprinting, etc. Its LenddoScore product is currently being used by banks and lending institutions worldwide to reduce risk, reach new customers and improve customer service. Lenddo's technology is designed to service thin-file and new-to-credit consumers, such as the upcoming middle class who is "underbanked" and in need of small loans and other financial services. The borrowers often lack an established credit history, making commercial banks reluctant to grant them credit but are often active users of social networks, enabling Lenddo to provide unique insights about their creditworthiness. For the purposes of this study, only a small anonymised subset of Lenddo's data was shared and analysed. The analysis and methodology presented in this article are similar in concept to the approaches used by Lenddo, however they do not describe any of the algorithms and scoring solutions currently or previously used by Lenddo in its business.

The predictive modelling task that we consider is identifying the risky loan applicants that would not fully repay their loan. For the analysis, we use data from Facebook and categorise it as follows: socio-demographic data, interest data and social network data. The socio-demographic data includes traditional features such as age, place of residence and education level. The interest data captures fine-grained data on for example the pages a user likes or the companies he worked for. Finally, the social network data consists of friendship connections between borrowers on Facebook. We

[1] http://partners.lenddo.com

use and combine this data in an innovative manner for credit scoring purposes as these define different relationships: look-a-likes, friends and BFFs (see Figure 6.1).

Look-a-likes (LAL) refer to people that are similar to one another. In this case this can be interpreted as persons either demonstrating similarities regarding certain socio-demographic characteristics, liking the same pages on Facebook, having a Facebook-friend in common or commenting on the same status. Clearly, this does not say anything about any real connections between those persons. That is, these individuals are not necessarily connected in real life, in fact they most likely have never met at all. However, the information included in these similarities can be an important predictor for default behavior since similar behavior in one domain (e.g. preferences) might imply similarities in other domains (e.g. default) as well (Martens and Provost 2011; Moeyersoms and Martens 2015; Provost, Martens, and Murray 2015a; Raeder et al. 2012). Additional Facebook data is available as explicitly stated *Friends*. The last category of data implies relationships of the form *"Best Friends Forever" (BFFs)*. These are Facebook friends that interact with one another, be it being tagged together in a picture, commenting on each others status, etc. Note that we do not distinguish in strength regarding the BFF relation, i.e. two persons are considered BFFs both in the case where one interaction occurs and in the case where multiple interactions are recorded.

The contributions of this chapter are three-fold, as illustrated in Figure 6.1. To our knowledge, we are the first to investigate the use of Facebook data for credit scoring for microfinance. The potential of such an automated credit scoring process is innovative and has large implications for the widespread use of microfinance and the potential economic growth of developing countries. Secondly, whereas previous studies that use Facebook data for predictive modeling focus on either the social network data or the interest data, we explicitly assess the combination of both. Finally, within the area of social network Facebook data, we further investigate the difference in predictive power of different levels of closeness, i.e. friends versus BFFs.

6.2 Related Work

6.2.1 *Credit scoring for microfinance*

Up to now, the use of interest-based and social network Facebook data to predict creditworthiness has not been investigated. Research on credit scoring mainly focuses on the use of structured data, such as sociodemographic factors (Banasik, J. Crook, and L. Thomas 2003; Hand, Sohn, and Kim 2005) and balance sheets (Emel et al. 2003; Min and Jeong 2009), thereby ignoring the high-quality information available in other data formats. In microfinance, the applicant's selection is often judgmental, i.e. the loan officer assesses the risk based on its own prior experience, his opinion on the applicant and the loan conditions (Schreiner 2003). In many cases the loan

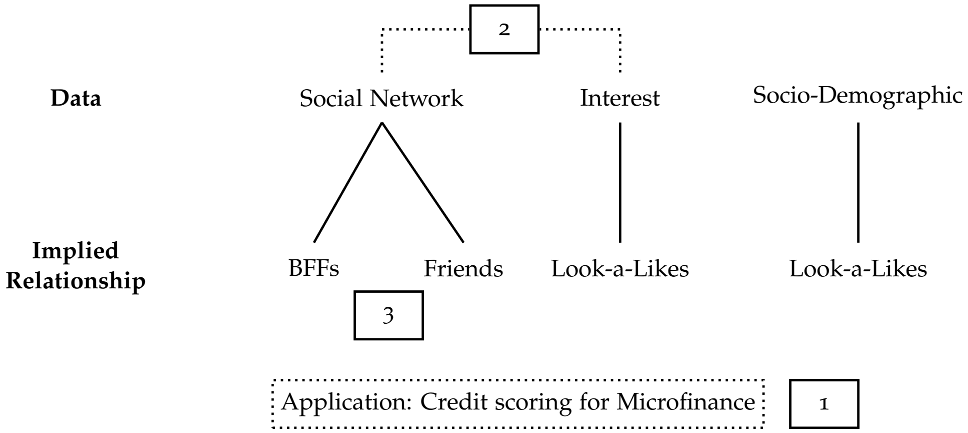


Figure 6.1.: Contributions.

officer communicates with the local community of the client to get an idea about the client's trustworthiness (Morduch 1999). In literature, this type of lending is called relationship-based lending where the lender gains information about the borrower during the course of their relationship. A second type of microfinance lending is group-based lending, in which social capital is created and used to alleviate the problem of asymmetric information and moral hazard (Hermes and Lensink 2007). Social capital - defined by Putman (Putnam 1995) as "features of social organization such as networks, norms, and social trust that facilitate cooperation and coordination" - operates under the form of peer-pressure in these joint liability groups.

Research on microfinance credit scoring is limited. Zeller (Zeller 1998) and Sharma and Zeller (Sharma and Zeller 1997) used group, community and lender or program characteristics to describe credit risk of joint liability groups. Schreiner (Schreiner 2003) remarked that statistical scoring will probably not work well for group-based lending, since there is no data on individual risk. Group risk appears to be much less strongly linked to group characteristics than individual risk to individual characteristics. Van Gool et al (Van Gool et al. 2012) investigated whether traditional credit scoring is applicable to microfinance lending. Using borrower, loan and lender characteristics they built a credit scoring model for a Bosnian microlender. They found that their credit scoring models are not able to fully replace the traditional credit process of manual screening. These findings confirm the conclusion of Schreiner (Schreiner 2000) whose study revealed that automated credit scoring complements, but does not replace the judgment of a loan officer based on qualitative, informal knowledge about the character of the applicant.

What the above mentioned studies have in common, is that they only use structured data in their credit scoring models. This data includes loan characteristics (purpose of the loan, duration of the loan), borrower characteristics (age, gender, education) and credit history (repayment of previous loans) and therefore does not differ much from the credit scoring models used in traditional lending. The complex nature of microfinance necessitates an assessment of character. Schreiner (Schreiner 2003) advises microlenders to search for personal character traits that are predictive of repayment behavior. Recently, Wei et al (Y. Wei et al. 2014) showed in a theoretical framework how network data can improve the accuracy of customer credit scores. Their framework is based upon the assumption of homophily, the notion that linked entities are more likely to have the same characteristics.

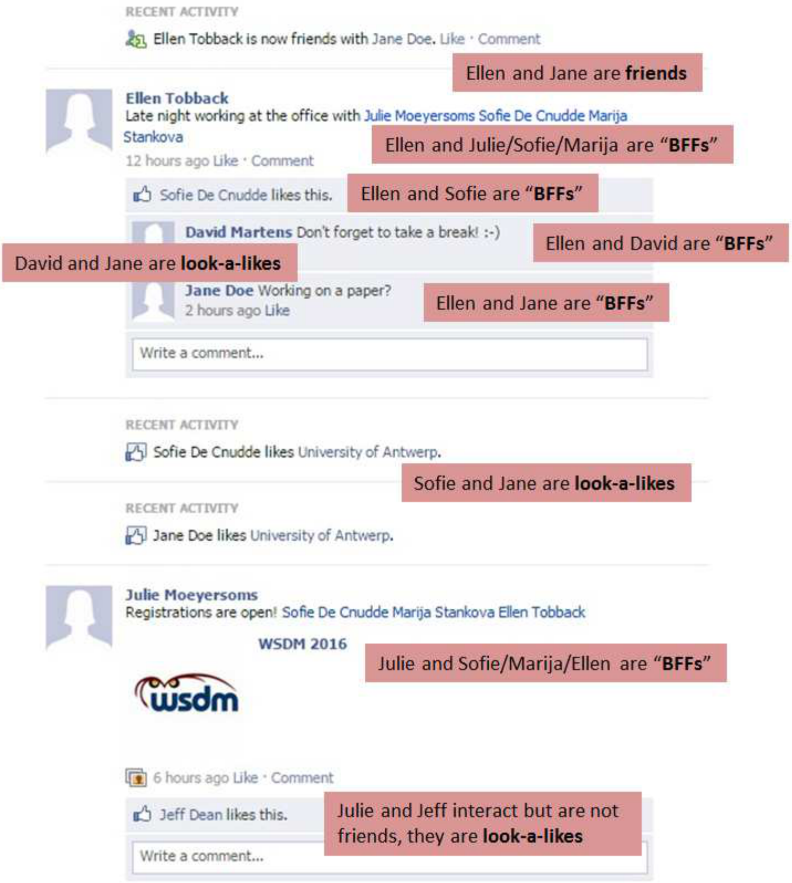
Moreover, Facebook has patented technology to assess creditworthiness of users based on credit ratings of people present in the users' social network (Facebook Inc 2014). Although not deployed yet, Facebook's interest in this data corroborates the possible value that lies in the use of alternative data for credit scoring purposes.

6.2.2 *Interest-based vs social network data*

Different types of data are commonly used for predictive modeling in a retail setting (Van Gestel, Baesens, and Martens 2015). Except for the conventional socio-demographic data, social network and interest data can be considered as well. Social network data represents real relationships between customers, while interest data refers to the often fine-grained observed interests and preferences of persons.

A seminal paper that uses social network data is that of Hill et al. (Hill, Provost, and Volinsky 2006), which uses the social relationships observed in calling behavior to predict product/service adoption in a telecommunications setting. Other studies have looked at call behavior as well to predict churn (Verbeke, Martens, and Baesens 2014) and social network data for viral marketing (Domingos 2005). However, often no real network data is available and other characteristics which are beyond the traditional socio-demographics data, can be used to detect similarities between people. For instance, Kosinski et al. (Kosinski, Stillwell, and Graepel 2013) and Junque de Fortuny et al. (Junqué de Fortuny, Martens, and Provost 2013) looked at predicting different personality traits from a dataset of users liking Facebook pages. The studies of Goel et al. (Goel, Hofman, and Sirer 2012) and Hu et al. (J. Hu et al. 2007) predict demographic attributes and Raeder et al. (Raeder et al. 2012) predict brand interest from people's browsing history. Weber et al. (I. Weber, Garimella, and Borra 2013) reveal political views from history of videos watched on YouTube. For financial applications, Martens and Provost (Martens and Provost 2011) predict interest in financial products from transactional datasets of consumers making payments to merchants and Provost et al. (Provost, Martens, and Murray 2015a) consider geo-location data to connect people if they visited the same places with the goal of predicting brand interest.

To the best of our knowledge, no study has included both social network and fine-grained interest-based data in order to predict default in microfinance settings. In



RECENT ACTIVITY

Ellen Tobback is now friends with Jane Doe. Like · Comment

Ellen and Jane are **friends**

Ellen Tobback
Late night working at the office with Julie Moeyersoms Sofie De Cnudde Marija Stankova
12 hours ago Like · Comment

Ellen and Julie/Sofie/Marija are "**BFFs**"

Sofie De Cnudde likes this.

Ellen and Sofie are "**BFFs**"

David Martens Don't forget to take a break! :-)

Ellen and David are "**BFFs**"

David and Jane are **look-a-likes**

Jane Doe Working on a paper?
2 hours ago Like

Ellen and Jane are "**BFFs**"

Write a comment...

RECENT ACTIVITY

Sofie De Cnudde likes University of Antwerp.

Sofie and Jane are **look-a-likes**

RECENT ACTIVITY

Jane Doe likes University of Antwerp.

Julie Moeyersoms
Registrations are open! Sofie De Cnudde Marija Stankova Ellen Tobback

WSDM 2016

Julie and Sofie/Marija/Ellen are "**BFFs**"

6 hours ago Like · Comment

Jeff Dean likes this.

Julie and Jeff interact but are not friends, they are **look-a-likes**

Write a comment...

Figure 6.2.: Illustration of look-a-likes, friends and BFFs.

this work, both data types are combined so that potential differences in predictive power between the data sources can be observed.

6.3 Data

A balanced sample is made available to us of 4,985 loan applications made by 4,512 users. As stated previously and visualized in Figure 6.1, we dispose of three data categories which we use to distinguish three levels of relations in terms of look-a-likes, friends and BFFs. We use Figure 6.2 to illustrate these. Table 6.1 shows a list of the constructed data structures along with some relevant data characteristics. Note that any names or personally identifiable information shown in this chapter are fictitious and do not relate to names or information of actual Lenddo members.

Name	Represented data	Category	m	N	ρ
Sociodemo	Socio-demographic attributes of a person	Socio-demographic data	29	111,989	83 %
LAL_Likes_Item	Persons liking a page on Facebook	Interest-based LAL	48,701	127,241	0.052%
LAL_Likescat_Item	Persons liking a category of a page on Facebook	Interest-based LAL	238	53,441	4.504%
LAL_Groups_Item	Persons joined in a group on Facebook	Interest-based LAL	38,037	55,399	0.029%
LAL_Education_Item	Persons going to specific educational institutions	Interest-based LAL	4,620	11,015	0.048%
LAL_Employers_Item	Persons working for employers	Interest-based LAL	5,190	13,173	0.051%
LAL_Position_Item	Persons holding employment positions or business titles	Interest-based LAL	3,393	9,983	0.059%
LAL_Comments_Items	Persons commenting on a status	Interest-based LAL	2,141,630	1,763,453	0.017%
LAL_Photos_Items	Persons mentioned in a picture	Interest-based LAL	293,155	404,896	0.028%
LAL_Links_Items	Persons mentioned in a link	Interest-based LAL	297,410	407,358	0.028%
LAL_Status_Items	Persons mentioned in a status	Interest-based LAL	667,298	806,411	0.024%
LAL_Videos_Items	Persons mentioned in a video	Interest-based LAL	27,442	33,602	0.024%
LAL_Likes_Items	Persons liking an item (video/status/photo/comment)	Interest-based LAL	4,122,418	2,846,613	0.014%
LAL_Comments_All	Persons giving/receiving comments to/from each other	Interest-based LAL	896,164	1,217,744	0.027%
LAL_Photos_All	Persons mentioning one another in one of their photos	Interest-based LAL	731,574	235,645	0.007%
LAL_Links_All	Persons mentioning one another in one of their links	Interest-based LAL	2,627,614	1,051,770	0.008%
LAL_Status_All	Persons mentioning one another in one of their statuses	Interest-based LAL	630,749	490,942	0.016%
LAL_Videos_All	Persons mentioning one another in one of their videos	Interest-based LAL	46,078	30,899	0.014%
LAL_Likes_All	Persons liking each other's video/status/photo/comment	Interest-based LAL	1,817,619	2,692,752	0.030%
LAL_Comments_Borrowers	Borrowers giving/receiving comments to/from each other	Relational LAL	4,985	20,301	0.081%
LAL_Photos_Borrowers	Borrowers mentioning one another in one of their photos	Relational LAL	4,985	9,199	0.037%
LAL_Links_Borrowers	Borrowers mentioning one another in one of their links	Relational LAL	4,985	14,318	0.057%
LAL_Status_Borrowers	Borrowers mentioning one another in one of their statuses	Relational LAL	4,985	9,949	0.040%
LAL_Videos_Borrowers	Borrowers mentioning one another in one of their videos	Relational LAL	4,985	1,496	0.006%
LAL_Likes_Borrowers	Borrowers liking each other's video/status/photo/comment	Relational LAL	4,985	29,814	0.120%
FRI_FBFriends	Borrowers befriending one another	Friends	4,985	30,347	0.122%
BFF_Comments	Friends giving/receiving comments to/from one another	BFF	4,985	18,391	0.074%
BFF_Photos	Friends mentioning one another in one of their photos	BFF	4,985	8,609	0.035%
BFF_Links	Friends mentioning one another in one of their links	BFF	4,985	13,072	0.053%
BFF_Status	Friends mentioning one another in one of their statuses	BFF	4,985	9,469	0.038%
BFF_Videos	Friends mentioning one another in one of their videos	BFF	4,985	1,438	0.006%
BFF_Likes	Friends liking each other's video/status/photo/comment	BFF	4,985	22,606	0.091%
BFF_All	Friends having any kind of interaction	BFF	4,985	24,243	0.098%

Table 6.1.: Overview of the constructed data matrices indicating when people are connected in the network, the category of the resulting relation, the number of features (m), the number of active elements (N) and the sparsity (ρ) defined as $\rho = N/(m \times n)$, with n the number of data instances ($n = 4,985$).

6.3.1 *Socio-demographic data*

The socio-demographic data originate from mandatory and optional information the user provides both Lenddo and Facebook. Such variables include date of birth, hometown, religion and school level. A total of 29 socio-demographic characteristics are used in the constructed Sociodemo matrix. The number of missing values is approximately 16.65%. Note that a missing value might denote data intentionally left blank by users, which is also modeled in the input data.

6.3.2 *Interest data*

In addition to traditionally available socio-demographic characteristics, we also dispose of fine-grained interest characteristics, which let us determine look-a-likes. First, there are interests which manifest themselves immediately. Liking a Facebook page or joining a Facebook group are direct testimonies of an interest. We also use schools visited, employers worked for and employment positions held to define an interest. Note that borrowers are not required to provide this information. In Figure 6.2, both Sofie and Jane like the page of University of Antwerp and therefore are look-a-likes. The constructed LAL_*_Item matrices model in a binary manner persons (rows) with a common interest (page or category of that page), group, school, employer or employment position (columns). Figure 6.3 shows a network of users and four page categories. Two of them are discriminative for defaulters, the other two for non-defaulters. Already this shows the potential of using such data for default prediction. Figure 6.4 shows the degree distributions for look-a-likes based on similar interests (pages and the categories of these pages) and groups. The distributions illustrate that a Facebook page, a category of a Facebook page and a group all have low probability of having many likes or memberships respectively, which is in line with previous research (Ugander et al. 2011).

Secondly, interests can also become clear by looking at interactions between users. In order to delimit the space of interactions considered in this study, we refer to interactions on Facebook belonging to one of these: (1) Interacting with a person using plain text, links, photos or videos (here, both *sharing* of the text, link, photo or video on someone's wall and *tagging* are included), (2) Commenting on text, links, photos or videos, and (3) Liking text, links, photos or videos. If two users comment on a status or like a status of the same person, this may imply a common interest. In Figure 6.2, David and Jane are look-a-likes as both of them comment on Ellen's status. Julie and Jeff are not friends, but both might be member of the WSDM group on Facebook which implies a common interest, making them look-a-likes.

Three types of data matrices are constructed to model look-a-likes in the network. First, the LAL_*_Borrowers matrix of size 4,985 x 4,985 represents borrowers directly interacting with one another through comments, photos, links, statuses, videos or likes. Since these interactions do not imply the users being friends, this matrix clearly

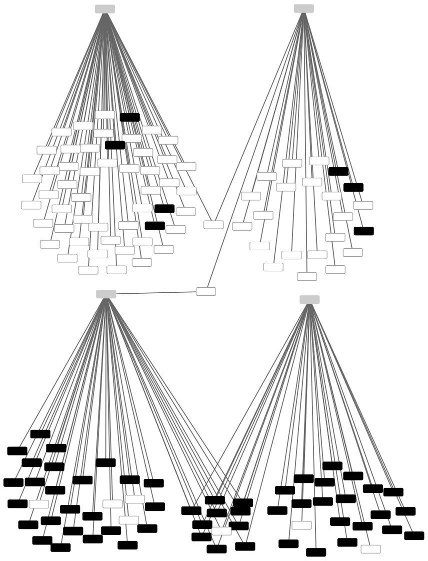


Figure 6.3.: Network of users liking four page categories (gray), two of them discriminative for defaulters (black) and two of them discriminative for non-defaulters (white).

represents look-a-likes. The second matrix, LAL_*_All, extends the previous one by also including interactions with Facebook users that are non-borrowers. Lastly, LAL_*_Items attempts to add even more information by representing an interaction between users (rows) and items (columns). Including the specific item commented on for example may add more detailed information with respect to the look-a-like relation.

6.3.3 *Social network data*

Social network data is used to distinguish plain friends from BFFs. Two users are referred to as friends if they befriended one another on Facebook. In the first interaction in Figure 6.2, Ellen and Jane become friends. This information is modeled

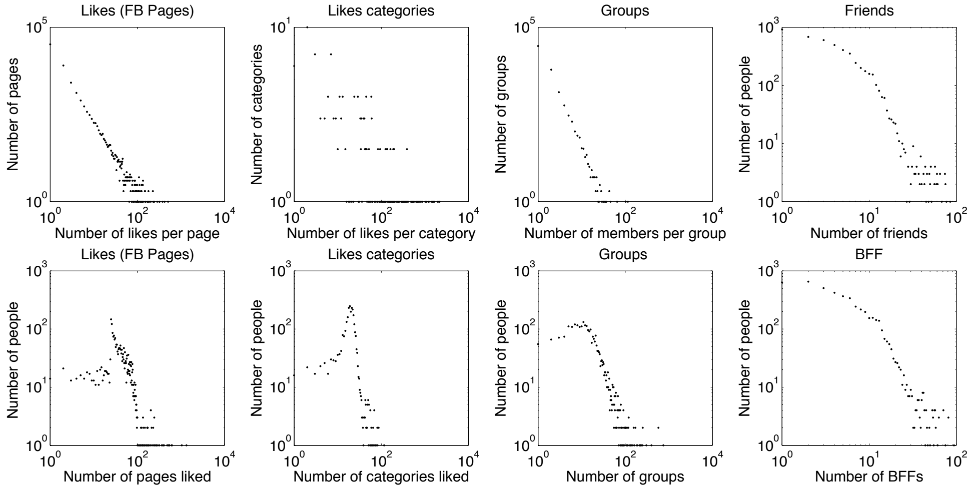
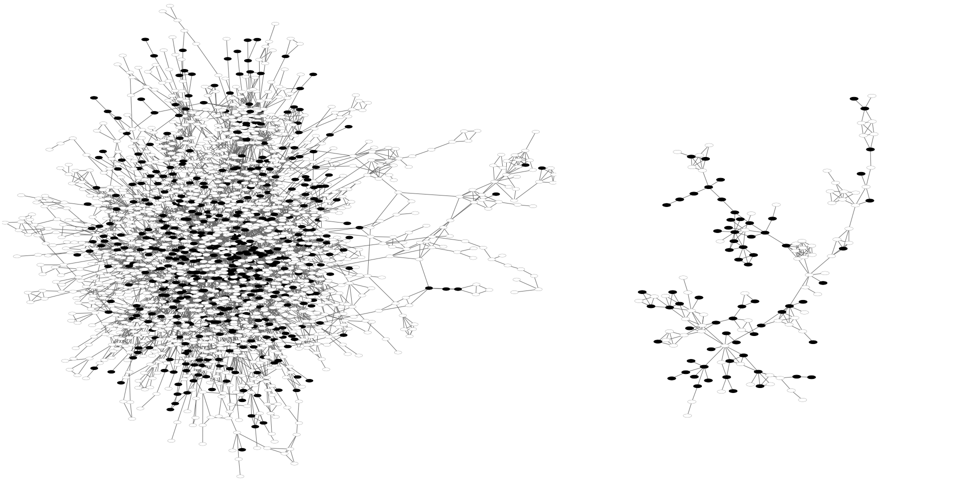


Figure 6.4.: Degree distributions for the pages, the categories of the pages, the groups, the friends and the BFFs.



(a) Network of friends.

(b) Network of BFFs interacting by photos (subset of network).

Figure 6.5.: The black dots in the graphs represent defaulters and the white dots represent non-defaulters.

in the FRI_FBFriends matrix. Figure 6.5(a) shows the friends connections between the borrowers. White nodes represent good borrowers, black nodes represent bad borrowers. The network is a large cluster in which no apparent pattern can immediately be distinguished.

Two Facebook friends that actually interact with one another by e.g. liking one another's status, makes them BFFs. When Jane comments on Ellen's status in the second interaction of Figure 6.2, Jane and Ellen change from being just friends to being BFFs. Supposing Julie, Sofie, Marija and Ellen befriended one another in the past, Julie tagging them in her status update, makes all of them BFFs. This data is modeled by combining the direct interactions in LAL_*_Borrowers with the friends in FRI_FBFriends. Figure 6.5(b) shows a portion of a BFF network based on interactions with photos. Two users in this network are connected if they are friends and if one of them has shared a photo on the other person's wall or mentioned the other person in a photo. The entire BFF photos network consists of separate, smaller networks like the one presented in Figure 6.5(b). The network clearly contains clusters of good and clusters of bad borrowers. Figure 6.4 shows the degree distribution of the friends and the BFFs on a log-log scale. Both distributions are monotonically decreasing and very similar to that of the entire Facebook community (Ugander et al. 2011), where most users have a moderate number of friends and only a few users have an unusually high degree.

6.4 **Methodology and Results**

6.4.1 *Methodology*

For each of the data categories we use specifically tailored techniques that we describe in more detail in the following section.

The interest-based data can be modelled as bipartite graphs (bigraphs), where one set of the nodes represents the loan applicants and the other set refers to their items of interest. We use the proposed three-step framework for node classification within bigraphs from Chapter 3 to create a weighted projection of the bigraph and then apply a unigraph relational learner. The projection is created by connecting the persons that have at least one shared interest and weighted in the following manner. Based on the empirical results from the study, we apply the hyperbolic tangent function to weight the items of interest, by assigning a lower score to the very popular items as being less informative for the target variable. In the following step, we calculate the strength w_{ij} between two persons i and j in the projection by summing the weights of their shared items of interest. To this weighted unigraph representation of the bigraph, we apply the network-only Link-Based classifier (nLB) (Lu and Getoor 2003), which is a powerful relational learner that is able to capture complex network patterns. As discussed in Chapter 3, the nLB classifier builds a class vector $CV(i)$ for every training instance (i.e. node) i in the network which contains the probability estimates (scores) that the node under study has a class label default or non-default (see Equation 3.7). From the formula, one can see the probability estimate of a node i belonging to a certain class (c), is calculated as a weighted average of the scores of its neighbouring nodes ($j \in N(i)$).

Subsequently, nLB creates a logistic regression model based on these class vectors (see Equation 3.8).

As an alternative to this network based approach, we also look at this from a standard classification perspective, where we apply a state-of-the-art discriminative learner on the matrix representation of the data (X. Wu et al. 2008). More specifically, we employ a linear SVM from the package LibLinear (Fan et al. 2008b) to the sparse, high-dimensional feature data. In a similar manner, the social network data can be modelled as graphs with only one type of nodes (unigraphs), where the persons are connected to their Facebook friends or BFFs. For this type of data, we again apply the linear SVM to the adjacency matrix and the nLB classifier directly on the unweighed unigraphs. Additionally, we also build a baseline SVM model with the 29 socio-demographic variables available for each loan applicant. The categorical variables are included in the model by dummy encoding them.

Finally, we incorporate all the pieces of information into an ensemble model, where the socio-demographic data are combined with the scores from the different techniques applied over the interest-based and the social network data. As a classification technique for the ensemble we use a linear SVM, since we need to be able to understand the decisions made by the classifier. Comprehensibility is an important issue in credit scoring and we further elaborate on it later. For the experimental setting we use a 10 fold cross-validation procedure where (i) 40% of the data is used for training and validation of the classifiers used with the interest based and the social network data, (ii) 40% is for training, 10% for validation and 10% for testing the ensemble model. As explained by Moeyersoms and Martens (Moeyersoms and Martens 2015), it is paramount that we carefully calculate the scores for the interest based and the social network data on a separate subset of the data that is not used for building the ensemble in order to avoid overfitting.

6.4.2 *Results*

The results for all different data sources are given in Figure 6.6 and 6.7 which present the performance for the SVM and nLB in terms of AUC. AUC is a widely-used performance evaluation metric in the machine learning community and represents the probability that a randomly chosen positive instance is ranked higher by the classification technique than a randomly chosen negative instance (Fawcett 2006). The Y-axis shows the AUC whereas the X-axis denotes the different data categories. The first observation is that the look-a-likes data, especially Likes and Likes categories have the most predictive value as compared to other data sources. Interestingly, for both methods the look-a-likes data performs better as compared to BFFs and friends data. That is, it appears from the results that similarities in interests or behavior includes more information than the real social network of a person with respect to default prediction.

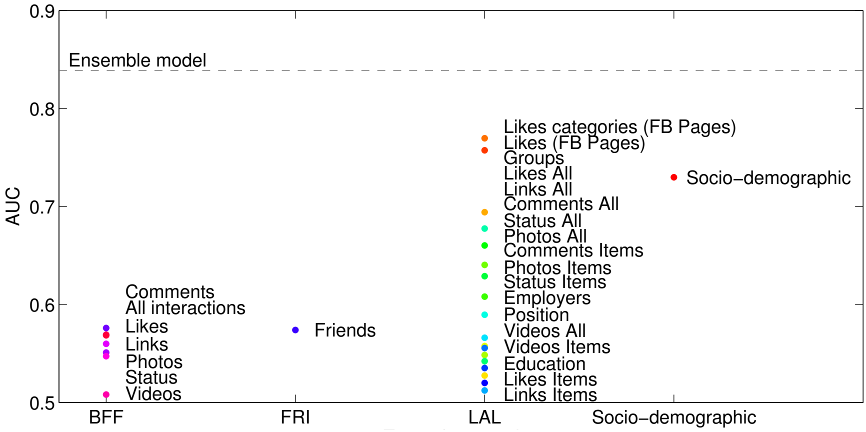


Figure 6.6.: AUC results for the different data categories when using a linear SVM.

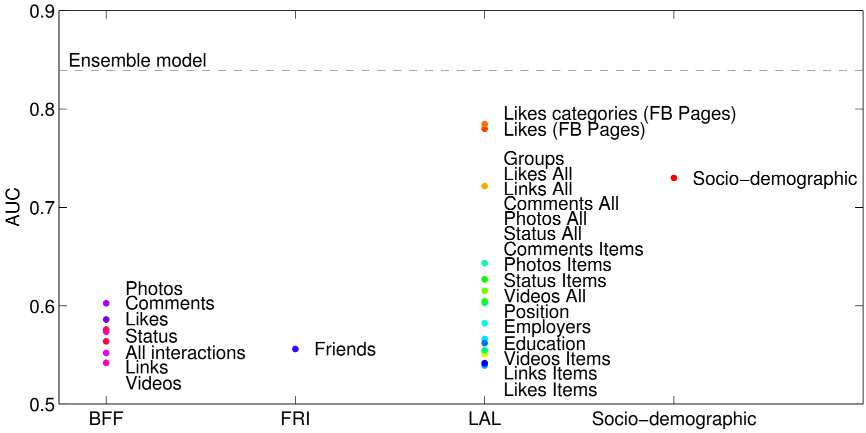


Figure 6.7.: AUC results for the different data categories when using the network-only Link-Based classifier (nLB).

The baseline socio-demographic model appears to have a large predictive performance as well, thereby performing better than BFFs, friends and even most of the look-a-likes data. When comparing BFFs with friends data, it can be seen that there is no major difference between BFFs and friends when applying the SVM. The nLB on the other hand, shows that most of the BFF data has higher predictive value as compared to friends. This could indicate that real, active friendships are more predictive than merely being connected on Facebook.

Lastly, the ensemble model, which includes all data sources, seems to outperform the individual data sources. The latter result can also be seen from Table 6.2, where the results, in terms of p-value, of the different models are tested as compared to the ensemble model by using a Wilcoxon signed rank test. The diagonal elements show the model where all data sources of the respective data type are included. The rest of the matrix indicate the results of the combinations of the corresponding data categories. The ensemble model, that uses all the data, is shown in the last row. Performances that are not significantly different at the 5% level from the top performance (ensemble model) with respect to a Wilcoxon signed rank test are tabulated in bold face. Statistically significant underperformances at the 1% level are emphasized in italics. From this table, one can conclude that although the ensemble model is performing best, the performance of the model which includes the look-a-likes data is not significantly worse as compared to the ensemble model. The same can be seen for other combinations of data which include the look-a-likes data. Again, this confirms our previous finding that interest data gives more information than the social network data. Moreover, this implies that in this case, using one source of data (look-a-likes) is sufficient to build the predictive model and assess creditworthiness.

Table 6.2.: Results (in terms of p-value of Wilcoxon signed rank test) of the different models. Performances that are not significantly different at the 5% level from the top performance (ensemble model) with respect to a Wilcoxon signed rank test are tabulated in bold face. Statistically significant underperformances at the 1% level are emphasized in italics.

	SD	look-a-likes	friends	BFFs	ensemble
SD	*0.002*	**0.232**	*0.002*	*0.002*	-
Look-a-likes	-	**0.193**	**0.275**	**0.193**	-
friends	-	-	*0.002*	*0.002*	-
BFFs	-	-	-	*0.002*	-
ensemble	-	-	-	-	**1.000**

Using these models, the credit scoring process becomes an automated process. It can complement the manual screening that is traditionally applied in microfinance. It is nevertheless also important for the credit lender to understand the predictions of the model (Martens, Baesens, et al. 2007). In credit scoring one is likely to be interested in knowing why a particular applicant was predicted to be a potential defaulter. An instance-level explanation method, that was developed to explain document classification, could be used to explain the predicted class (Martens and Provost 2014b). In this case an explanation would be defined as the minimal set of likes/interactions such that removing this set changes the class. A possible explanation could be: *If the user would NOT have liked ("Who cares about data science?" "Credit scoring is boring") then the class would change from default to non-default.* For further information regarding the implementation of this method, we refer to (Martens and Provost 2014b).

6.5 Conclusion

In this chapter, we investigated the potential of Facebook data for microfinance credit scoring. The good predictive performance of the generated models allows to automate the credit scoring process for microfinance to massive settings, mainly thanks to the ability to include the difficult concept of character. The splitup in different data categories shows that there is a significant difference in the predictive power of each, with interest-based data being the most valuable. It should be noted however that our methodology is limited to the setting where Facebook data is available, which is not always the case in microfinance lending. Also, the validity of our results is limited to this specific application on a dataset from the Philippines. It would be interesting to see to what extent these findings on BFFs and friends, as well as the superiority of interest-based data translate to other applications .

This chapter is part of the following publication: Marija Stankova*, De Cnudde Sofie*, Julie Moeyersoms*, Ellen Tobback*, Vinayak Javaly, David Martens, ″Who cares about your Facebook friends? Credit scoring for microfinance″, 2015, Working paper, University of Antwerp, Belgium. My contribution is focused on the network analysis and part of the literature study.

Part IV

NONLINEAR CLASSIFICATION WITHIN DATA WITH A BIPARTITE STRUCTURE

7

Non-linear classification within bipartite data

The general conclusions in prior literature have been that for dense datasets, non-linear classification tend to lead to better predictive performance than linear classification at the cost of longer training and testing time. In this thesis, we study a different type of large, sparse datasets with a bipartite structure, where every feature contains small and relevant information about the target variable. Motivated by the success of non-linear classification with dense datasets, in this chapter we move the discussion forward and look at whether this non-linear approach can provide better results in our context. To be more specific, we investigate the predictive performance of applying non-linear techniques to the datasets or adding higher order interaction effects to the original features. Since we deal with big datasets, this means that more complex models can be applied to the data. The predictive results are compared to a benchmark linear technique and show, similarly to previous literature, that for the considered approaches there are only limited improvements in using non-linear classification with this kind of datasets.

7.1 Introduction

In the case of "dense" datasets, where the instances have non-trivial values for most features, prior literature has shown that linear methods generally have low predictive performance compared to highly non-linear methods (Baesens et al. 2003; Chang, C.-J. Hsieh, et al. 2010). Although the non-linear methods achieve better accuracy, this usually comes at the cost of slower training and lower scalability (Fan et al. 2008a). Given the success of the linear approach that we introduced before, in this chapter we investigate the potential improvement of adding non-linearities. Due to the large dimensionality and sparseness of the specific type of datasets that we study, non-linear techniques are often not computationally feasible. Therefore, how to perform non-linear classification for this type of large data is still an open question.

First of all, let us consider what the intuition would be for adding non-linearities. If we reconsider the running example from Chapter 2, where a bigraph of people visiting locations is used to predict brand interest, let us assume we know that two persons both visited the campus of the University of Antwerp and the same tennis club in Berchem (see Figure 7.1). In a linear model, the coefficients of each location would simply be added and there is no additional gain/reduction in the score because of these joint visits. If we would introduce interaction effects, such a combined visit could be used to indicate the much stronger signal for brand interest (for example for a sports reduction card of the University of Antwerp). In terms of the bigraph, this would mean adding new top nodes that are combinations of the existing nodes.

Unlike the dense datasets, we deal with sparse data where many different features (and combinations of features) carry very small amounts of information about the target variable. The most accurate models will integrate all that information, as also motivated by Zaidi et al.: *"The amount of information present in big data is typically much greater than that present in small quantities of data. As a result, big data can support the creation of very detailed models that encode complex higher-order multivariate distributions, whereas, for small data, very detailed models will tend to overfit and should be avoided"* (Zaidi et al. 2015). Prior literature has looked at this problem mainly in the context of document classification, where the datasets are also characterised by high-dimensionality and sparseness, though at a much smaller scale (e.g. thousands of dimensions versus millions of dimensions). In this domain, the bigraphs are defined between documents and the terms they contain (*bag-of-words* representation). The general observations have been that linear classifiers provide comparable results to non-linear data for documents classification and that the additional complexity of non-linear classification does not tend to pay for itself in terms of significantly better predictions (Aggarwal and Zhai 2012; Rennie and Rifkin 2001; Yang and Liu 1999; Zhang, Yoshida, and Tang 2008). In this chapter, we extend the analysis to other types of bigraphs and examine whether the conclusions generalise to our kind of datasets too.

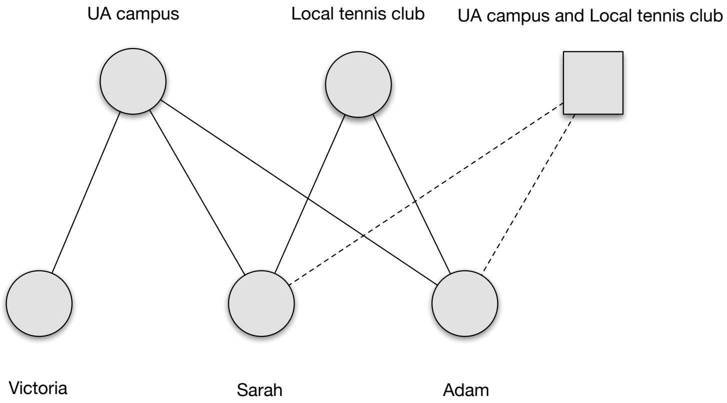


Figure 7.1.: Adding higher order interaction effects to the bigraph.

In general, we could group the commonly used approaches for non-linear classification in the following manner:

1. **Applying non-linear techniques:** such as Neural Networks (Hornik, Stinchcombe, and White 1989), Random Forests (Breiman 2001) or SVMs with a non-linear kernel (Vapnik 2013) to the original input space or some mapped feature space. By using intelligent feature engineering (with techniques like Singular Value Decomposition (Clark and Provost 2015)), one can map the input space to a linear space of features with smaller dimensions that should capture most of the information.
2. **Adding non-linear features:** on top of the original input space, to get the advantage of linear classifiers' fast training with explicit non-linear data mappings to capture the non-linear patterns in the data. These feature combinations are shown to be useful for dense datasets (Cheng et al. 2007; Van Gestel et al. 2007).

7.2 Datasets

As we mentioned above, the fine-grained data that we look at are characterised by large dimensions and high sparseness. Since we already introduced the datasets in the previous chapters, here we only give a brief overview and a summary of the data statistics in Table 7.1. To be more specific, we use the following binary data with a two-class target variable. The MovieLens [1] and Yahoo (Koenigstein, Dror, and Koren 2011) datasets entail information about movie ratings and our goal is to predict the gender of the users. In a similar manner, we use data about book ratings from the website Bookcrossing.com (Ziegler et al. 2005) to predict the age of the users. Furthermore, data about which products were bought in a supermarket (TaFeng) or browsed online (PAKDD) are used to predict the gender and the age of the consumers, respectively (H.-S. Huang et al. 2005). The Loans dataset includes mobile phone usage data from a microlender company and our aim is to predict default (see Chapter 6). Lastly, we use transactional data between companies situated in Belgium and abroad to detect corporate residence fraud (see Chapter 5).

In Figure 7.2, we additionally illustrate the sparseness of the data (percentage of elements with a value zero in the matrix) in combination with the size of the datasets. As one can see, the sparseness of the datasets is very high, with all of them except for the MovieLens dataset having more than 99% of zero elements.

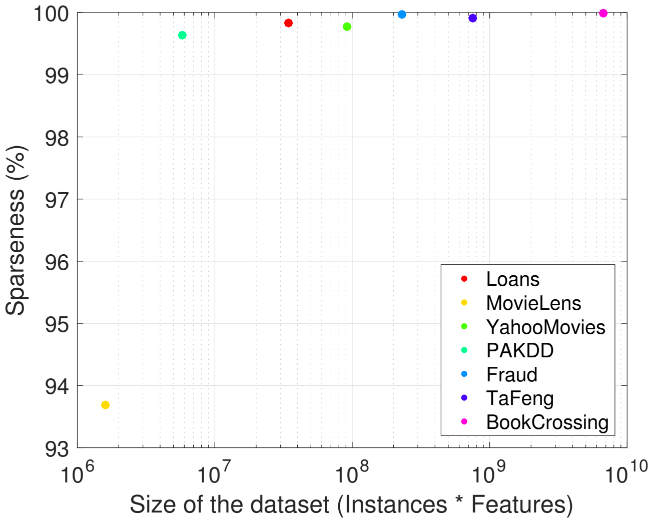


Figure 7.2.: Size and sparseness (% of elements with value zero) of the datasets under study.

7.3 Methods and feature engineering

As discussed in Section 7.1, we summarize the previous approaches for non-linear classification into two categories: applying non-linear techniques and adding non-linear features to the original input space. More specifically, we examine the first approach of applying non-linear techniques to the original input set by using an SVM

[1] http://www.grouplens.org

Dataset	Target Variable	l_0	l_1	Features	Instances	Active elements	Active pairs	Active pairs that appear more than once
Loans	default	632	152	51,069	784	75,829	0.1503%	0.0008%
MovieLens	gender	273	670	1682	943	100,000	69.5474%	48.8453%
Yahoo Movies	gender	2,206	5,436	11,916	7,642	221,330	8.2595%	2.8554%
PAKDD	gender	3,297	11,703	441	15,000	20,323	2.6263%	1.0874%
Fraud	fraud	30,479	62	7,495	30,541	73,753	0.6677%	0.1384%
TaFeng	age	17,330	14,310	23,719	31,640	723,449	4.4042 %	1.0932%
Book-Crossing	age	21,709	24,542	145,457	46,251	301,470	323,699,711	0.3060%

Table 7.1.: Descriptive statistics of the datasets: class distribution (l_0 and l_1), number of features (top nodes), number of instances (bottom nodes), the number of elements with a value one in the data (number of links), percent of all pairs that appear at least once in the dataset (active pairs), percent of all pairs that appear more than once in the dataset.

with a non-linear RBF kernel (Bishop 2006). The function of the kernel is to implicitly map the inputs into high-dimensional features space, where the data can be linearly separable (see Section 1.1). Although this has shown accuracy improvements over linear classification for dense datasets, the time needed for training and testing larger datasets is usually very long.

For the second approach, we create new features that are combinations of the existing single features (singles) and add them to the original input space. Explicitly enumerating all feature combinations would be unnecessary since the data are highly sparse and most of them will not appear in the data. Therefore, we only consider the combinations that are active pairs, meaning that they have at least one non-zero value in the matrix. In the context of our running example, this means that we take into account only the pairs of locations that have both been visited by at least one person. We only consider combinations of two features (pairs), because it makes little sense to create higher order features given the sparseness of the data. Even these combinations of size two do not generalise well, as most pairs appear only once in the data (see Table 7.1). We further analyse if selecting only high-quality pairs would strengthen the performance, instead of using all the active pairs. Feature selection is mainly employed for dense datasets to either avoid overfitting or select the few features that are relevant in some applications (Joachims 1998). To select the most informative features, we use the following strategies/measures: (i) L1 regularization (Bishop 2006), (ii) Information Gain (IG) (Forman 2003; Joachims 1998) and (iii) using a simple heuristics that in every cycle selects the feature with the highest support among the instances that are not yet covered by the previously selected features (Cheng et al. 2007).

In summary, given the prior literature, we decided to test the following approaches:

1. A linear model on the original input space: *'SVM - Singles'*.
2. A linear model on the original input space, with L1 regularization: *'L1 - Singles'*.
3. A linear model on the original input space, with IG feature selection: *'IG - Singles'*.
4. A RBF SVM model on the original input space: *'SVM (RBF) - Singles'*.
5. A linear model on the space with all active interactions of two inputs. (RBF SVM can not be trained due to the dimensions.): *'SVM (only pairs)'*.
6. A linear model on the space with both the input space and all active interactions of two inputs: *'SVM (all)'*.
7. A linear model on the space with both the input space and selected interactions of two inputs using L1 regularization: *'L1 - Singles and Pairs'* .
8. A linear model on the space with both the input space and selected interactions of two inputs using IG: *'IG - Singles and Pairs'*.

9. A linear model on the space with both the input space and selected interactions of two inputs Support Heuristics: *'Support Heur. - Singles and Pairs'*.

7.3.1 *Experimental setting*

The basic technique that is used with all the aforementioned approaches is a Support Vector Machine (see Section 1.1) with a linear or an RBF kernel from the LibSVM toolbox (Chang and Lin 2011). In order to asses the predictive performance of the methods, we use a 10 fold cross-validation procedure and we report the average AUC result over all folds (Fawcett 2006). Since we need to select a proper value for several parameters on a separate validation set, we additionally use nested cross-validation that adds an inner 3 fold loop to each of the training sets. Therefore, the inner loop is used to select the parameter values that yield the best AUC results and the outer loop is used to train a new model with the previously chosen parameter values. In case when multiple parameters need to be tuned in the cross-validation procedure, we do a full grid search where all combinations of possible parameter values are tested. In our study, the value of the C parameter is chosen from {0.01, 0.1, 1, 10}, for gamma among {0.001,0.01,0.1,1,10} and for the Information Gain threshold we look at the top ranked {0.01,0.1,1,10} percent of the features.

7.4 **Results**

The results from our experiments are shown in Table 7.2, with the best methods for each dataset emphasised in bold. As one can see, the approach of using a non-linear technique over the data (*'SVM (RBF) - Singles'*), provides comparable results to our benchmark linear technique (*'SVM - Singles'*). Indeed, this corresponds to the observations from the text mining literature, that the gains from using a non-linear technique are rather small (if existent) for this type of datasets, especially when one takes into account the time needed for training the two models [2]. The only notable example of much better predictive performance is on the highly-imbalanced Fraud dataset, where the results from the RBF SVM are in the same range as the binary Bernoulli Naive Bayes and the SW transformation from Chapter 5 [3]. Moreover, selecting only a subset of features to be used with the linear model (*'L1 - Singles'* and *'IG - Singles'* [4]), as expected, degrades performance. Since the results are worse than considering all the singles, this shows that the features carry small amount of additional and relevant information that should be included for better predictions.

[2] For the larger BookCrossing and TaFeng datasets, it took hours to finish training the non-linear models, compared to a couple of minutes for the linear ones.

[3] It remains an interesting question for future research, what is the effect for extremely skewed datasets, as well as oversampling techniques for such data.

[4] As expected, for all datasets the best selected threshold for Information Gain was the one that includes most features (in our case 10% of all features).

The results of the third approach for performing non-linear classification, by adding new features to the original input space, are labeled as "Singles and Pairs" in Table 7.2. By looking at the results of applying a linear technique over the pairs (*'SVM (only pairs)'*), we can see that there is a signal in the new features, since the AUC values are larger than 0.5. However, adding the pairs unselectively to the singles (*'SVM (all)'*), generally dilutes the performance. L1 (*'L1 - Singles and Pairs'*), on the other hand, seems to select good pairs and adds a bit to the performance compared to only considering the singles, although again the gain is not so large. By using L1, the independent or near-independent features get a coefficient of zero, instead of some really small value. Interestingly, in our experiments L1 gives zeros to all pairs for the largest TaFeng and BookCrossing datasets. This means that the pairs for these datasets carry very small, in this case neglectable, amount of information. The last two approaches for selecting informative pairs by using Information Gain (*'IG - Singles and Pairs'*) or Support Heuristics (*'Support Heur. - Singles and Pairs'*), perform generally worse than feature selection with L1 (*'L1 - Singles and Pairs'*).

Finally, in Figure 7.3 we visualise the predictive performance, by plotting the AUC difference between our benchmark model (linear SVM) and all the other methods. As discussed previously, the non-linear approaches provide comparable results to the linear technique.

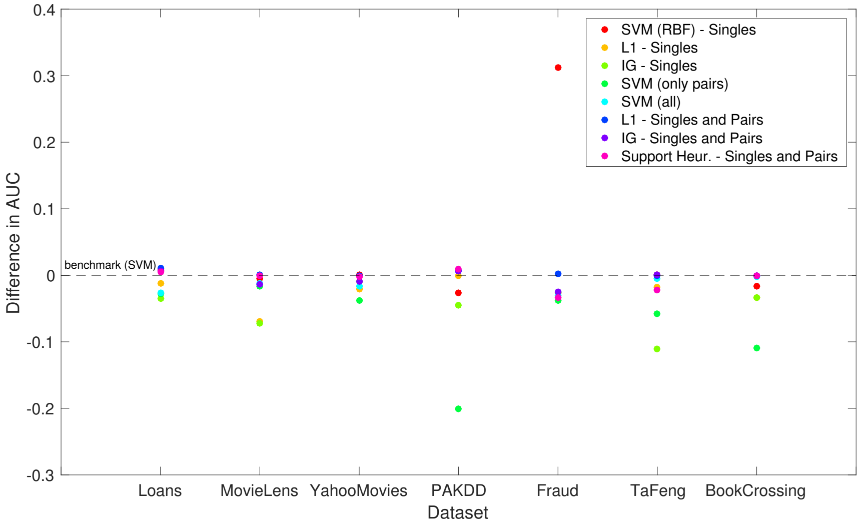


Figure 7.3.: Results.

7.5 Conclusion

In this chapter, we examine the added value of using non-linear classification with high-dimensional, sparse data given the prior success of this approach for dense data.

Dataset	Singles				Singles and Pairs				
	SVM	SVM (RBF)	L1	IG	SVM (only pairs)	SVM (all)	L1	IG	Support Heur.
Loans	0.5335 (±0.0629)	0.5420 (±0.0560)	0.5220 (±0.0688)	0.4978 (± 0.0581)	0.5054 (±0.0608)	0.5064 (±0.0684)	**0.5449** (**±0.0559**)	0.5399 (± 0.0580)	0.5390 (±0.0651)
MovieLens	0.7672 (±0.0433)	0.7621 (±0.0253)	0.6976 (±0.0433)	0.6948 (± 0.0339)	0.7502 (±0.0380)	0.7551 (±0.0401)	**0.7675** (**±0.0432**)	0.7537 (± 0.0374)	0.7663 (±0.0422)
Yahoo Movies	0.8053 (±0.0150)	**0.8066** (**±0.0091**)	0.7847 (±0.0134)	0.7967 (± 0.0148)	0.7674 (±0.0146)	0.7894 (±0.0139)	0.8048 (±0.0149)	0.7955 (± 0.0169)	0.8034 (±0.0149)
PAKDD	0.8011 (±0.0121)	0.7740 (±0.0156)	0.8001 (±0.0118)	0.7555 (± 0.0126)	0.5998 (±0.0119)	0.8087 (±0.0094)	0.8076 (±0.0101)	0.8073 (± 0.0106)	**0.8100** (**±0.0102**)
Fraud	0.4859 (±0.1651)	**0.7976** (**±0.1586**)	0.4591 (±0.1048)	0.4590 (±0.0953)	0.4483 (±0.1404)	0.4557 (±0.1517)	0.4887 (±0.1623)	0.4605 (± 0.1597)	0.4521 (±0.1610)
TaFeng	0.7033 (±0.0112)	**0.7040** (**±0.0114**)	0.6853 (±0.0108)	0.5928 (± 0.0175)	0.6448 (±0.0096)	0.6979 (±0.0092)	0.7033 (±0.0112)	0.7034 (± 0.0113)	0.6811 (±0.0087)
BookCrossing	**0.6453** (**±0.0084**)	0.6295 (±0.107)	0.6112 (±0.0096)	0.6112 (± 0.0062)	0.5366 (±0.0056)	0.6429 (±0.0104)	**0.6453** (**±0.0084**)	0.6452 (± 0.0082)	0.6443 (±0.0075)

Table 7.2.: Results.

More specifically, we focused on applying a non-linear technique over the data and engineering new features which are added to the original input space. The results seem to be in line with the prior text mining literature, that reports rather small predictive performance improvement in some cases using non-linear classification. Moreover, the results of using feature selection over the original feature space show that this will likely hurt performance due to the loss of information. Finally, we do not claim that non-linear classification does not work overall for this type of data, we rather argue that the aforementioned approaches have limited success in this context.

Ethical reflection

The collection of human behaviour data used to be a difficult and time consuming process, where the data were collected through a set of traditional methods such as surveys, case studies and interviews (Boyd and Crawford 2012). Nowadays, our online activities leave traces of personal information (also known as digital footprints (Shmueli 2016)), that are readily accessed and processed by companies, governments and even other individuals. Different parties automatically collect vast amount of data from social network interactions, search queries, payment transactions, clickstreams, logs, health records, mobile phone usage, etc in order to analyse them and gain useful insights and knowledge. With such an automation of the data collection process and reduction of the costs for storing information, it becomes increasingly easy for many parties to get involved with Big Data. However, although the field of data science is evolving fast and the number of practitioners is increasing rapidly, the privacy protection laws surrounding the field do not develop at such pace. This lack of solid legislation increases the chances of immoral behaviour and potential misuse of the data and can lead to decreased civil rights and higher government or corporate control (Boyd and Crawford 2012). In many cases, it is likely that the individuals that create the data are not aware that their data are being collected or they do not fully understand the risks associated with analysing them. Therefore, an ethical conduct and an adequate privacy protection should remain an imperative for all data science practitioners and policymakers. We must note that although the main focus of this thesis is not on the privacy aspects of data science, we believe that this is an important issue that deserves an ethical reflection.

The International Telecommunications Union defines *privacy* as "the right of individuals to control or influence what information related to them may be disclosed" (International Telecommunications Union 2003). Moreover, the European Commission (EC) considers *personal data* as "any information relating to an identified or identifiable natural person", where an identifiable person is "one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity" (European Commission 1995). The EC recognises the right of the individuals to have an effective control over their personal data as a fundamental right for every European citizen (European Commission 2016). However, a recent report (European Commission 2016) has shown that the individuals are concerned about their online privacy and how their data are being handled. Only 15% of the survey respondents feel like they have complete control over the information they provide online and 71% feel that they are obliged to disclose personal information in order to get an online service. Companies consider Big Data as a valuable asset that can position them better in the market. They collect and analyse human behaviour data to provide personalised services or products to their customers (Koren, Bell, and Volinsky 2009), identify the likely churners (Moeyersoms and Martens 2015; Verbeke, Martens, and Baesens 2014) or adopters of the service (Martens and Provost 2011) for marketing purposes, detect fraudsters (Bhattacharyya et al. 2011), hire better employees (Ranjan, Goyal, and Ahson 2008), manage the risks associated with investments or giving loans (Huang, Chen, and Wang 2007), etc. Nevertheless, the use of predictive modelling with sensitive data can have negative consequences for the individuals, including exclusion and discrimination which are harder to detect and prove due to the automation of the process (Boyd and Crawford 2012). Predictive modelling can infer unknown information about the individuals, including potentially sensitive attributes like medical conditions or sexual orientation (Kosinski, Stillwell, and Graepel 2013). Furthermore, these personal data could be exposed through data breaches (Fhom 2015) or even through the company's marketing campaigns, such as the famous case when Target promoted baby products to a pregnant girl, even before she told her family [1]. Another worrisome aspect that shows how little control the individuals have over the process is the fact that many of the companies actively trade with their consumers' data (wrongfully or not) [2]. As acknowledged by Junqué de Fortuny et al. (Junqué de Fortuny, Martens, and Provost 2013), more data lead to more accurate predictions, which in turn gives the large data-centric companies like Google, Facebook, the large banks, telecommunication companies and media a high concentration of power and advantage over their smaller competitors (also known as information asymmetry (Fhom 2015)). On the other hand, the companies are not the only potential threat to the individuals' rights and freedoms. Governments also, especially authoritarian regimes, could potentially misuse personal data for increased control and manipulation of the society.

The aforementioned examples clearly illustrate the need for adequate control mechanisms. In the past, companies have mainly relied on informed consent and anonymization to protect the privacy of their customers (Barocas and Nissenbaum 2014). As explained by Barocas and Nissenbaum (Barocas and Nissenbaum 2014), these methods face several challenges in the context of predictive modelling. With informed consent for instance, the companies expect their consumers to understand and give consent to often very complicated privacy disclosures. Also, in a predictive modelling setting, the decisions of what other consumers disclose about themselves, have an impact on what the company can infer about the individual. Anonymization, on the other hand, has its own pitfalls. When companies claim to use anonymized data, this usually means that they do not collect or they exclude identifying attributes such as name, address, national identity number, etc (Barocas and Nissenbaum 2014). This, however, does not mean that they do not hold attributes that can distinguish a specific browser, computer or phone from others (such as AdID by Google or

[1] http://www.forbes.com/sites/kashmirhill/2012/02/16/how-target-figured-out-a-teen-girl-was-pregnant-before-her-father-did

[2] http://www.gartner.com/newsroom/id/2299315

AppleID by Apple) and to which they associate the individual's behaviour data. Previously, this would mean that the companies could not be able to get external attributes for an individual by matching different databases. As discussed in this work, with predictive modelling the companies can infer unknown attributes about the consumers, based on other consumers data.

This calls for additional measures that would strengthen privacy protection. In both the United States and the European Union, the privacy legislation is framed around the OECD guidelines on privacy protection [3], with a various degree of implementation. In what follows, we discuss a part of the privacy principles in the European legislation [4] and how they can help overcome some of the privacy issues. These principals are not binding in many countries around the world, but we believe that they could serve as useful ethical guidelines for the data science practitioners. Firstly, companies and governments should be *transparent* about their Big Data practices and inform the individuals in a clear and simple manner about the purpose of the data collection and the logic behind the decision making process, without compromising their internal confidential information. As discussed previously, they should ask for explicit *consent* for gathering the data [5] [6], but also for making them available to others (e.g. data trading). This would allow the individuals to better understand the implications of their actions and would also give them an opportunity to challenge the decisions made by the models. Moreover, the data collectors should focus on gathering the *minimal* set of data necessary to reach their legitimate goals and not further intrude the individuals' privacy. They should also be responsible and provide a *secure* infrastructure for storing the data and notify the individuals in case of any data breaches. Additionally, the latest reforms in the EU legislation require *accountability* from all larger or data-centric companies and public authorities, meaning they need to provide internal regulators (Data Privacy Officers) that can ensure all personal data are handled in compliance with the law and the ethical principles of the organisation.

Lastly, we would like to discuss the privacy protection within academia. Most of the universities, especially in the US and in EU, have ethical committees [7] whose task is to oversee the risks associated with research that includes human behaviour data (Shmueli 2016). An approval of such committees, is often required by many funding agencies or journals for publishing. However, the recent cases of Danish researchers publishing sensitive data from the dating website OkCupid [8] and the study on emotional behaviour with Facebook data from Cornell researchers [9] show

[3] https://www.oecd.org/sti/ieconomy/privacy-guidelines.htm

[4] http://ec.europa.eu/justice/data-protection

[5] Within the EU legislation, the authorities are not a subject to this rule when it comes to matters of public interest.

[6] http://ec.europa.eu/justice/data-protection/data-collection/legal/index_en.htm

[7] In the US, these committees are known as Institutional Review Boards (IRB).

[8] http://fortune.com/2016/05/18/okcupid-data-research

[9] http://mediarelations.cornell.edu/2014/06/30/media-statement-on-cornell-universitys-role-in-facebook-emotional-contagion-research

that these ethical approvals can be circumvented. In this type of circumstances, where the rules are not well defined, the ethical conduct of the data science researchers is of high importance. Therefore, the universities should invest more in ethical courses for data scientists, something which is now mainly aimed for social science researchers (Shmueli 2016).

Conclusions

The focus of this dissertation is on the task of node classification within a special type of datasets that can be modelled as bigraphs. As discussed throughout this work, the bigraphs are an intuitive representation for many relational, behavioural and transactional datasets. Most of the previous studies that tackled this problem, have looked at it from a classical classification perspective with massive, sparse feature data. We, on the other hand, proposed an alternative network-based formulation that effectively deals with this specific type of bigraph data. Moreover, we validated our designs with two real-world applications, namely for fraud detection and credit scoring. In this section, we summarize the main features of the chapters and highlight their contributions. Furthermore, we also discuss potential avenues for future research.

1 Thesis conclusions

In **Chapter** 3 we proposed a general three-stage framework for doing classification in bipartite data via projection, which was informed by our survey of prior work in the area. For each of the stages, we applied the existing methods from literature and we also introduced some new alternatives. By mixing-and-matching the techniques from the different stages, we had the flexibility to explore the design space more systematically than prior work has done. In fact, the best performing combination of techniques and several other well ranked classifiers were new methods discovered by the framework. The results of our comparative analysis over a large benchmark collection of bipartite datasets are encouraging and also show the superiority of this network-oriented formulation over a classical approach, where a linear SVM is applied on the adjacency matrix of the bigraph. Lastly, we introduced a fast and comprehensible technique (SW-transformation) that scales easily to very big datasets with up to millions of nodes.

In **Chapter** 4 we builded upon our work from the previous chapter and proposed a multiple-stage framework for node classification within weighted bigraphs. The framework is build systematically, by framing the intuition behind every stage in a set of preferred properties that guide the design of the proposed methods. The experiments showed that the predictive performance of the alternative settings, similarly to the prior studies that tackled this problem, were inconclusive of whether it is beneficial to consider the link weights. We, on the other hand, provided an extensive assessment of our approach over multiple datasets, where the best AUCs on every dataset were achieved by also including the link weights. Therefore, to best of our knowledge, this is the first study which proved that by including information about the bigraph link weights, we can create more representative projections and thus improve the prediction results.

In **Chapter 5** we collaborated with the Belgian government to detect companies that fraudulently reside outside of Belgium for tax benefits. This entails what we believe to be the first published data-mining-based approach to detecting corporate residence fraud. More specifically, we applied our framework from Chapter 3, among other methods, on bigraphs of transactions between companies situated in Belgium and abroad. The results again showed the predictive performance advantage of our framework in terms of both AUC and lift over a classical setting, where a linear SVM is applied on the adjacency matrix of the bigraph. There was however, no significant difference when compared to the binary Bernoulli Naive Bayes (Junqué de Fortuny, Martens, and Provost 2013). The SW-transformation proved to be very suitable for this application as it can scale to the large dimensions of the transactional datasets. More importantly, the comprehensibility of the model can give the necessary confidence to the auditors to use the system.

In **Chapter 6** we worked together with a micro-lender company called Lenddo, in order to assess the creditworthiness of the loan applicants. To the best of our knowledge, we were the first to investigate the use of alternative Facebook data for credit scoring for microfinance. The potential of such an automated credit scoring process is innovative and has large implications for the widespread use of microfinance and the potential economic growth of developing countries. For the study, we applied our framework from Chapter 3 on bigraphs of Facebook behavioural data such as applicants liking pages, being members of groups, being tagged in a photo, commenting on an item, etc. Our results revealed interesting insights as these behavioural bigraphs that demonstrate the interests and the preferences of a person proved to be more valuable for default prediction than the online friendship network of a person. It would be an interesting avenue for future research to see how these findings translate to other domains.

Lastly, in **Chapter 7**, we examined the added value of using non-linear classification with high-dimensional, sparse (bipartite) data given the prior success of this approach for dense data. More specifically, we focused on applying a non-linear technique over the data and adding higher order interaction effects to the original input space. The results seem to be in line with prior literature, and show the limited predictive performance improvement that can be achieved using non-linear classification with this type of datasets.

2 Future research

The graph mining literature so far has mainly focused on designing metrics and techniques for the more simple case of graphs with homogeneous nodes (unigraphs). However, bigraphs have a specific structure that should be exploited for better predictive performance. In this work, we concentrated on a projecting approach that would allow the practitioners to make use of the wealth of unigraph techniques already available. The modular frameworks open the design space to new methods and thus to new classifiers that could possibly achieve better performances. That being said, this work can directly be extended by considering other methods for the different stages in the two frameworks, for both unweighted and weighted bigraphs.

Moreover, the approach that we presented largely simplifies the problems under study in many domains, where additional information about the nodes and the edges might be available. The amount of information available from both sources, i.e. the network structure and the local information, varies greatly for each dataset and so does the predictive power that comes from the two of them. The local attributes about the bottom nodes can be included in the analysis by for instance applying a traditional model over such structured data (local information) and then using the scores as priors in the relational methods (Macskassy and Provost 2007). On the other hand, the scores from the relational classifiers can also be considered as new features that capture the information encoded in the relations between the nodes and be used to complement the structured data in a traditional propositional model. The later approach was considered in Chapter 5 and Chapter 6, where we investigated the value of both data sources and concluded that combining them results in better predictions. Nevertheless, a problem arises with including the top node and the edge attributes in the setting, since both frameworks try to model this information directly in the weights of the projection. Including more information in the projections certainly presents an interesting avenue for future research and as the interest for this type of data increases, we expect significant advances in the form of alternative settings and techniques for node classification, some of which could possibly be directly applicable to the bigraphs. Hopefully the frameworks introduced in this dissertation represent a stepping stone to such advancements.

Furthermore, in this dissertation we also discussed how these frameworks can be applied for two real-world applications, that for instance could be extended by considering data from more sources. As for the credit scoring application, we could create new bigraphs from other platforms such as bigraphs of loan applicants as one set of nodes and the tweets they have shared or favoured, the emails they have sent or received, the people they have called to or received calls from or even the phone applications that they use as the other set of nodes. Some of these data are already being collected by the microlender company Lenddo. In the case of the fraud detection application, a potential collaboration between governments could possibly provide more data and define bigraphs between the foreign companies and their board of directors or shareholders. Moreover, it would be interesting to consider other domains where the frameworks could be useful, such as bioinformatics to define bigraphs of proteins and metabolic processes with the aim of annotating the protein function, in banking to discover potential adopters of financial products from bigraphs of payers and payment receivers, in the telecommunications industry to identify likely churners from bigraphs of people and the locations they visit, for the police to detect deviant behaviour from bigraphs of citizens browsing webpages and many more.

Bibliography

Adamic, Lada A and Eytan Adar (2003). "Friends and neighbors on the Web." In: *Social Networks* 25.3, pp. 211–230 (cit. on pp. 23, 25, 28, 30).

Adomavicius, Gediminas and Alexander Tuzhilin (2005). "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions." In: *Knowledge and Data Engineering, IEEE Transactions on* 17.6, pp. 734–749 (cit. on p. 86).

Aggarwal, Charu C and ChengXiang Zhai (2012). "A survey of text classification algorithms." In: *Mining text data*. Springer, pp. 163–222 (cit. on p. 136).

Allali, Oussama, Clémence Magnien, and Matthieu Latapy (2011). "Link prediction in bipartite graphs using internal links and weighted projection." In: *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*, pp. 936–941 (cit. on pp. 23, 25, 26, 28, 30, 42).

Antonie, M-L and Osmar R Zaiane (2002). "Text document categorization by term association." In: *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on*. IEEE, pp. 19–26.

Arya, Sunil, David M Mount, Nathan S Netanyahu, Ruth Silverman, and Angela Y Wu (1998). "An optimal algorithm for approximate nearest neighbor searching fixed dimensions." In: *Journal of the ACM (JACM)* 45.6, pp. 891–923 (cit. on p. 20).

Baer, Miriam H (2008). "Linkage and the Deterrence of Corporate Fraud." In: *Virginia Law Review*, pp. 1295–1365 (cit. on p. 102).

Baesens, Bart, Tony Van Gestel, Stijn Viaene, Maria Stepanova, Johan Suykens, and Jan Vanthienen (2003). "Benchmarking state-of-the-art classification algorithms for credit scoring." In: *Journal of the operational research society* 54.6, pp. 627–635 (cit. on p. 136).

Banasik, Jonathan, John Crook, and Lyn Thomas (2003). "Sample selection bias in credit scoring models." In: *Journal of the Operational Research Society* 54.8, pp. 822–832 (cit. on p. 119).

Barber, Michael J (2007). "Modularity and community detection in bipartite networks." In: *Physical Review E* 76.6, p. 066102 (cit. on p. 43).

Barocas, Solon and Helen Nissenbaum (2014). "Big data's end run around procedural privacy protections." In: *Communications of the ACM* 57.11, pp. 31–33 (cit. on p. 146).

Basta, Stefano, Fabio Fassetti, Massimo Guarascio, Giuseppe Manco, Fosca Giannotti, Dino Pedreschi, Laura Spinsanti, Gianfilippo Papi, and Stefano Pisani (2009). "High Quality True-Positive Prediction for Fiscal Fraud Detection." In: *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*. IEEE, pp. 7–12 (cit. on pp. 102, 103).

Benchettara, Nesserine, Rushed Kanawati, and Celine Rouveirol (2010). "Supervised machine learning applied to link prediction in bipartite social networks." In: *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*. IEEE, pp. 326–330 (cit. on pp. 42, 43).

Bennett, James and Stan Lanning (2007). "The netflix prize." In: *Proceedings of KDD cup and workshop*. Vol. 2007, p. 35 (cit. on pp. 66, 69).

Bhattacharyya, Siddhartha, Sanjeev Jha, Kurian Tharakunnel, and J Christopher Westland (2011). "Data mining for credit card fraud: A comparative study." In: *Decision Support Systems* 50.3, pp. 602–613 (cit. on pp. 4, 102, 146).

Bishop, Christopher M (2006). "Pattern recognition." In: *Machine Learning* 128 (cit. on pp. 6, 28, 140).

Bolton, Richard J and David J Hand (2002). "Statistical fraud detection: A review." In: *Statistical Science*, pp. 235–249 (cit. on p. 102).

Bolton, Richard J, David J Hand, et al. (2001). "Unsupervised profiling methods for fraud detection." In: *Credit Scoring and Credit Control VII*, pp. 235–255 (cit. on p. 102).

Borgatti, Stephen P and Martin G Everett (1997). "Network analysis of 2-mode data." In: *Social networks* 19.3, pp. 243–269 (cit. on p. 42).

Borgatti, Stephen P and Daniel S Halgin (2011). "Analyzing affiliation networks." In: *The Sage handbook of social network analysis*, pp. 417–433 (cit. on pp. 12, 42, 43).

Boyd, Danah and Kate Crawford (2012). "Critical questions for big data: Provocations for a cultural, technological, and scholarly phenomenon." In: *Information, communication & society* 15.5, pp. 662–679 (cit. on pp. 145, 146).

Brandes, Ulrik (2008). "On variants of shortest-path betweenness centrality and their generic computation." In: *Social Networks* 30.2, pp. 136–145 (cit. on p. 70).

Brause, R, T Langsdorf, and Michael Hepp (1999). "Neural data mining for credit card fraud detection." In: *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*. IEEE, pp. 103–106 (cit. on pp. 102, 103).

Breese, John S, David Heckerman, and Carl Kadie (1998). "Empirical analysis of predictive algorithms for collaborative filtering." In: *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., pp. 43–52 (cit. on p. 86).

Breiman, Leo (2001). "Random forests." In: *Machine learning* 45.1, pp. 5–32 (cit. on p. 137).

Brozovsky, Lukas and Vaclav Petricek (2007). "Recommender System for Online Dating Service." In: *Proceedings of Conference Znalosti 2007*. Ostrava: VSB (cit. on pp. 34, 66, 80).

Cancho, Ramon Ferrer i and Richard V Solé (2001). "The small world of human language." In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 268.1482, pp. 2261–2265 (cit. on p. 11).

Cecchini, Mark, Haldun Aytug, Gary J Koehler, and Praveen Pathak (2010). "Detecting management fraud in public companies." In: *Management Science* 56.7, pp. 1146–1160 (cit. on p. 103).

Cha, Sung-Hyuk (2007). "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions." In: (cit. on p. 79).

Chang, Cho-Jui Hsieh, Kai-Wei Chang, Michael Ringgaard, and Chih-Jen Lin (2010). "Training and testing low-degree polynomial data mappings via linear SVM." In: *Journal of Machine Learning Research* 11.Apr, pp. 1471–1490 (cit. on p. 136).

Chang and Lin (2011). "LIBSVM: a library for support vector machines." In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3, p. 27 (cit. on p. 141).

Cheng, Hong, Xifeng Yan, Jiawei Han, and Chih-Wei Hsu (2007). "Discriminative frequent pattern analysis for effective classification." In: *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, pp. 716–725 (cit. on pp. 137, 140).

Cho, Eunjoon, Seth A. Myers, and Jure Leskovec (2011). "Friendship and mobility: user movement in location-based social networks." In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '11. San Diego, California, USA: ACM, pp. 1082–1090 (cit. on p. 13).

Clark, Jessica and Foster Provost (2015). "Dimensionality Reduction via Matrix Factorization for Predictive Modeling from Large, Sparse Behavioral Data." In: (cit. on p. 137).

Claypool, Mark, Phong Le, Makoto Wased, and David Brown (2001). "Implicit interest indicators." In: *Proceedings of the 6th international conference on Intelligent user interfaces*. ACM, pp. 33–40 (cit. on pp. 66, 69).

Conitzer, Vincent, Andrew Davenport, and Jayant Kalagnanam (2006). "Improved bounds for computing Kemeny rankings." In: *AAAI*. Vol. 6, pp. 620–626 (cit. on pp. 36, 82).

Copestake, James (2007). "Mainstreaming microfinance: social performance management or mission drift?" In: *World Development* 35.10, pp. 1721–1738 (cit. on p. 118).

Cortes, Corinna, Daryl Pregibon, and Chris Volinsky (2001). *Communities of interest*. Springer (cit. on pp. 102, 103).

Crombez, John (2013). *Zwart en wit*. De Bezige Bij (cit. on pp. 99, 100).

De Sá, Hially Rodrigues and Ricardo Bastos Cavalcante Prudêncio (2011). "Supervised link prediction in weighted networks." In: *Neural Networks (IJCNN), The 2011 International Joint Conference on*. IEEE, pp. 2281–2288 (cit. on p. 87).

Demšar, Janez (2006). "Statistical comparisons of classifiers over multiple data sets." In: *The Journal of Machine Learning Research* 7, pp. 1–30 (cit. on pp. 36, 82, 111).

Devroye, Luc (1996). *A probabilistic theory of pattern recognition*. Vol. 31. Springer (cit. on pp. 20, 21).

Domingos, Pedro (2005). "Mining social networks for viral marketing." In: *IEEE Intelligent Systems* 20.1, pp. 80–82 (cit. on p. 121).

Doreian, Patrick, Vladimir Batagelj, and Anuška Ferligoj (2004). "Generalized blockmodeling of two-mode network data." In: *Social Networks* 26.1, pp. 29–53 (cit. on p. 43).

Duan, Ran and Hsin-Hao Su (2012). "A scaling algorithm for maximum weight matching in bipartite graphs." In: *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM, pp. 1413–1424 (cit. on p. 67).

Eagle, Nathan and Alex Pentland (2006). "Reality mining: sensing complex social systems." In: *Personal and Ubiquitous Computing* 10.4, pp. 255–268 (cit. on pp. 11, 34).

Easley, David and Jon Kleinberg (2010). *Networks, crowds, and markets*. Vol. 8. Cambridge Univ Press (cit. on pp. 11, 18, 19).

Emel, Ahmet Burak, Muhittin Oral, Arnold Reisman, and Reha Yolalan (2003). "A credit scoring approach for the commercial banking sector." In: *Socio-Economic Planning Sciences* 37.2, pp. 103–123 (cit. on p. 119).

EUR-LEX (2012). *Communication from the Commission to the European Parliament and the Council* (cit. on p. 101).

European Commission (2013). *Fight against tax fraud and tax evasion: A huge problem*. Taxation and Customs Union (cit. on p. 100).

European Commission, EU (1995). "The Data Protection Directive, EU Directive 95/46/EC." In: URL: `https://www.dataprotection.ie/docs/EU-Directive-95-46-EC-Chapter-1/92.htm` (cit. on p. 145).

European Commission, EU (2016). "The EU Data Protection Reform and Big Data factsheet." In: URL: `http://ec.europa.eu/justice/data-protection/files/data-protection-big-data_factsheet_web_en.pdf` (cit. on p. 145).

Facebook Inc (2014). *Authorization and authentication based on an individual's social network*. Patent (cit. on p. 121).

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin (2008a). "LIBLINEAR: A Library for Large Linear Classification." In: *Journal of Machine Learning Research* 9, pp. 1871–1874 (cit. on pp. 35, 109, 136).

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin (2008b). "LIBLINEAR: A library for large linear classification." In: *The Journal of Machine Learning Research* 9, pp. 1871–1874 (cit. on pp. 78, 128).

Faust, Katherine (1997). "Centrality in affiliation networks." In: *Social networks* 19.2, pp. 157–191 (cit. on p. 42).

Fawcett, Tom (2006). "An introduction to ROC analysis." In: *Pattern recognition letters* 27.8, pp. 861–874 (cit. on pp. 7–9, 35, 82, 128, 141).

Fawcett, Tom and Foster Provost (1996). "Combining Data Mining and Machine Learning for Effective User Profiling." In: *Proceedings of the Third KDD International Conference on Knowledge Discovery and Data Mining*, pp. 8–13 (cit. on pp. 102, 103).

Fawcett, Tom and Foster Provost (1997). "Adaptive fraud detection." In: *Data mining and knowledge discovery* 1.3, pp. 291–316 (cit. on pp. 102, 103).

Fhom, Hervais Simo (2015). "Big Data: Opportunities and privacy challenges." In: *arXiv preprint arXiv:1502.00823* (cit. on p. 146).

Forbes, Catherine, Merran Evans, Nicholas Hastings, and Brian Peacock (2011). *Statistical distributions*. John Wiley & Sons (cit. on pp. 24, 74).

Forman, George (2003). "An extensive empirical study of feature selection metrics for text classification." In: *Journal of machine learning research* 3.Mar, pp. 1289–1305 (cit. on p. 140).

Fortunato, Santo (2010). "Community detection in graphs." In: *Physics Reports* 486.3, pp. 75–174 (cit. on p. 43).

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani (2001). *The elements of statistical learning*. Vol. 1. Springer series in statistics Springer, Berlin (cit. on pp. 4, 5).

Gallagher, Brian, Hanghang Tong, Tina Eliassi-Rad, and Christos Faloutsos (2008). "Using ghost edges for classification in sparsely labeled networks." In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 256–264 (cit. on p. 42).

Goel, Sharad, Jake M Hofman, and M Irmak Sirer (2012). "Who Does What on the Web: A Large-Scale Study of Browsing Behavior." In: *ICWSM* (cit. on pp. 14, 18, 121).

Gonzlez, Pamela Castelln and Juan D. Velsquez (2013). "Characterization and detection of taxpayers with false invoices using data mining techniques." In: *Expert Systems with Applications* 40.5, pp. 1427–1436. URL: `http://dblp.uni-trier.de/db/journals/eswa/eswa40.html#GonzalezV13` (cit. on pp. 102, 103).

Granovetter, Mark S (1973). "The strength of weak ties." In: *American journal of sociology*, pp. 1360–1380 (cit. on p. 87).

Gregor, Shirley and Izak Benbasat (1999). "Explanations from intelligent systems: Theoretical foundations and implications for practice." In: *MIS quarterly*, pp. 497–530 (cit. on pp. 5, 40).

Guillaume, Jean-Loup and Matthieu Latapy (2006). "Bipartite graphs as models of complex networks." In: *Physica A: Statistical Mechanics and its Applications* 371.2, pp. 795–813 (cit. on pp. 11–13, 19, 28, 30).

Gupte, Mangesh and Tina Eliassi-Rad (2012). "Measuring tie strength in implicit social networks." In: *Proceedings of the 4th Annual ACM Web Science Conference*. ACM, pp. 109–118 (cit. on pp. 23–26, 28, 30, 44, 88).

Hand, David J, So Young Sohn, and Yoonseong Kim (2005). "Optimal bipartite scorecards." In: *Expert Systems with Applications* 29.3, pp. 684–690 (cit. on p. 119).

Hermes, Niels and Robert Lensink (2007). "The empirics of microfinance: what do we know?" In: *The Economic Journal* 117.517, F1–F10 (cit. on p. 120).

Hilas, Constantinos S and Paris As Mastorocostas (2008). "An application of supervised and unsupervised learning approaches to telecommunications fraud detection." In: *Knowledge-Based Systems* 21.7, pp. 721–726 (cit. on p. 102).

Hill, Shawndra, Foster Provost, and Chris Volinsky (2006). "Network-Based Marketing: Identifying Likely Adopters via Consumer Networks." In: *Statist. Sci.* 21.2, pp. 256–276. URL: `http://dx.doi.org/10.1214/088342306000000222` (cit. on p. 121).

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White (1989). "Multilayer feedforward networks are universal approximators." In: *Neural networks* 2.5, pp. 359–366 (cit. on pp. 5, 137).

Hsu, Chun-Nan, Hao-Hsiang Chung, and Han-Shen Huang (2004). "Mining skewed and sparse transaction data for personalized shopping recommendation." In: *Machine Learning* 57.1-2, pp. 35–59 (cit. on pp. 66, 69).

Hu, Jian, Hua-Jun Zeng, Hua Li, Cheng Niu, and Zheng Chen (2007). "Demographic prediction based on user's browsing behavior." In: *Proceedings of the 16th international conference on World Wide Web*. ACM, pp. 151–160 (cit. on pp. 14, 18, 121).

Hu, Yifan, Yehuda Koren, and Chris Volinsky (2008). "Collaborative filtering for implicit feedback datasets." In: *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, pp. 263–272 (cit. on pp. 66, 85).

Huang, Han-Shen, Koung-Lung Lin, Jane Yung-jen Hsu, and Chun-Nan Hsu (2005). "Item-triggered recommendation for identifying potential customers of cold sellers in supermarkets." In: *Proceedings of Beyond Personalization*, pp. 37–42 (cit. on pp. 34, 81, 85, 138).

Huang, Zan, Xin Li, and Hsinchun Chen (2005). "Link prediction approach to collaborative filtering." In: *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*. ACM, pp. 141–142 (cit. on pp. 42, 43).

Huang, Chen, and Chieh Wang (2007). "Credit scoring with a data mining approach based on support vector machines." In: *Expert systems with applications* 33.4, pp. 847–856 (cit. on pp. 4, 146).

International Telecommunications Union, (ITU) (2003). "Security in Telecommunications and Information Technology: An overview of issues and the deployment of existing ITU-T Recommendations for secure telecommunications." In: p. 2. URL: `http://www.itu.int/ITU-T/edh/files/security-manual.pdf` (cit. on p. 145).

Jacobs, Adam (2009). "The pathologies of big data." In: *Communications of the ACM* 52.8, pp. 36–44 (cit. on p. 3).

Jensen, David and Jennifer Neville (2002). "Linkage and autocorrelation cause feature selection bias in relational learning." In: *ICML*. Vol. 2. Citeseer, pp. 259–266 (cit. on p. 38).

Jensen, David, Jennifer Neville, and Brian Gallagher (2004). "Why collective inference improves relational classification." In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 593–598 (cit. on p. 11).

Joachims, Thorsten (1998). "Text categorization with support vector machines: Learning with many relevant features." In: *European conference on machine learning*. Springer, pp. 137–142 (cit. on p. 140).

Jones, Karen Sparck (1972). "A Statistical Interpretation of Term Specificity and its Application in Retrieval." In: *Journal of Documentation* 28, pp. 11–21 (cit. on p. 23).

Junqué de Fortuny, Enric, David Martens, and Foster Provost (2013). "Predictive Modeling With Big Data: Is Bigger Really Better?" In: *Big Data* 1.4, pp. 215–226 (cit. on pp. 21, 106, 109, 146, 150).

Junqué de Fortuny, Enric, David Martens, and Foster Provost (2013). *Wallenius Naive Bayes*. Tech. rep. 2451/33545. New York University (cit. on p. 121).

Juszczak, Piotr, Niall M Adams, David J Hand, Christopher Whitrow, and David J Weston (2008). "Off-the-peg and bespoke classifiers for fraud detection." In: *Computational Statistics & Data Analysis* 52.9, pp. 4521–4532 (cit. on p. 102).

Kirkos, Efstathios, Charalambos Spathis, and Yannis Manolopoulos (2007). "Data mining techniques for the detection of fraudulent financial statements." In: *Expert Systems with Applications* 32.4, pp. 995–1003 (cit. on p. 103).

Koenigstein, Noam, Gideon Dror, and Yehuda Koren (2011). "Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy."

In: *Proceedings of the fifth ACM conference on Recommender systems*. ACM, pp. 165–172 (cit. on pp. 66, 80, 85, 138).

Koren, Yehuda, Robert Bell, and Chris Volinsky (2009). "Matrix factorization techniques for recommender systems." In: *Computer* 42.8, pp. 30–37 (cit. on pp. 14, 146).

Kosinski, Michal, David Stillwell, and Thore Graepel (2013). "Private traits and attributes are predictable from digital records of human behavior." In: *Proceedings of the National Academy of Sciences* 110.15, pp. 5802–5805 (cit. on pp. 14, 18, 121, 146).

Kunegis, Jérôme, Ernesto W De Luca, and Sahin Albayrak (2010). "The link prediction problem in bipartite networks." In: *Computational intelligence for knowledge-based systems design*. Springer, pp. 380–389 (cit. on p. 43).

Lambiotte, Renaud and Marcel Ausloos (2005). "Uncovering collective listening habits and music genres in bipartite networks." In: *Physical Review E* 72.6, p. 066107 (cit. on p. 43).

Laney, Doug (2001). "3D data management: Controlling data volume, velocity and variety." In: *META Group Research Note* 6, p. 70 (cit. on p. 3).

Latapy, Matthieu, Clémence Magnien, and Nathalie Del Vecchio (2008). "Basic notions for the analysis of large two-mode networks." In: *Social Networks* 30.1, pp. 31–48 (cit. on pp. 11, 66).

Latapy, Matthieu, Clémence Magnien, and Nathalie Del Vecchio (2008). "Basic Notations for the Analysis of Large Two - mode Networks." In: *Social Networks* 30, pp. 31–48 (cit. on pp. 12, 13, 18, 42, 43).

Liben-Nowell, David and Jon Kleinberg (2007). "The link-prediction problem for social networks." In: *Journal of the American society for information science and technology* 58.7, pp. 1019–1031 (cit. on p. 42).

Lind, Pedro G, Marta C Gonzalez, and Hans J Herrmann (2005). "Cycles and clustering in bipartite networks." In: *Physical review E* 72.5, p. 056127 (cit. on p. 42).

Linden, Greg, Brent Smith, and Jeremy York (2003). "Amazon. com recommendations: Item-to-item collaborative filtering." In: *Internet Computing, IEEE* 7.1, pp. 76–80 (cit. on pp. 66, 85, 86).

Lü, Linyuan and Tao Zhou (2010). "Link prediction in weighted networks: The role of weak ties." In: *EPL (Europhysics Letters)* 89.1, p. 18001 (cit. on p. 87).

Lu, Qing and Lise Getoor (2003). "Link-based classification." In: *ICML*. Vol. 3, pp. 496–503 (cit. on pp. 28, 42, 127).

Macskassy, Sofus and Foster Provost (2003). "A Simple Relational Classifier." In: *Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM-2003) at KDD-2003* (cit. on pp. 27, 42, 110).

Macskassy, Sofus and Foster Provost (2005). "Suspicion scoring based on guilt-by-association, collective inference, and focused data access." In: *International conference on intelligence analysis* (cit. on p. 115).

Macskassy, Sofus and Foster Provost (2007). "Classification in networked data: A toolkit and a univariate case study." In: *The Journal of Machine Learning Research* 8, pp. 935–983 (cit. on pp. 13, 18, 21, 25–27, 30, 42, 44, 78, 151).

Marr, Bernard (2015). "Why only one of the 5 Vs of big data really matters." In: URL: `http://www.ibmbigdatahub.com/blog/why-only-one-5-vs-big-data-really-matters` (cit. on p. 3).

Martens, David and Bart Baesens (2010). "Building acceptable classification models." In: *Data Mining*. Springer, pp. 53–74 (cit. on p. 5).

Martens, David, Bart Baesens, Tony Van Gestel, and Jan Vanthienen (2007). "Comprehensible credit scoring models using rule extraction from support vector machines." In: *European journal of operational research* 183.3, pp. 1466–1476 (cit. on pp. 5, 40, 130).

Martens, David and Foster Provost (2011). "Pseudo-social network targeting from consumer transaction data." In: *Faculty of Applied Economics, University of Antwerp, Belgium* (cit. on pp. 11, 19, 20, 23–25, 30, 44, 66, 69, 106, 119, 121, 146).

Martens, David and Foster Provost (2014a). "Explaining Data-Driven Document Classifications." In: *MIS Quarterly* 38.4 (cit. on pp. 40, 113).

Martens, David and Foster Provost (2014b). "Explaining data-driven document classifications." In: *MIS Quarterly* 38.1, pp. 73–100 (cit. on pp. 5, 40, 130).

Martens, David, Foster Provost, Jessica Clark, and Enric Junqué de Fortuny (2013). *Mining fine-grained consumer payment data to improve targeted marketing*. Tech. rep. Technical report, Stern School of Business, New York University (cit. on p. 79).

McFee, Brian, Thierry Bertin-Mahieux, Daniel PW Ellis, and Gert RG Lanckriet (2012). "The million song dataset challenge." In: *Proceedings of the 21st international conference companion on World Wide Web*. ACM, pp. 909–916 (cit. on pp. 66, 85).

McPherson, Miller, Lynn Smith-Lovin, and James M Cook (2001). "Birds of a feather: Homophily in social networks." In: *Annual review of sociology*, pp. 415–444 (cit. on pp. 11, 19).

Min, Jae H and Chulwoo Jeong (2009). "A binary classification method for bankruptcy prediction." In: *Expert Systems with Applications* 36.3, pp. 5256–5263 (cit. on p. 119).

Moeyersoms, Julie and David Martens (2015). "Including high-cardinality attributes in predictive models: A case study in churn prediction in the energy sector." In: *Decision Support Systems* 72, pp. 72–81 (cit. on pp. 4, 119, 128, 146).

Morduch, Jonathan (1999). "The microfinance promise." In: *Journal of economic literature*, pp. 1569–1614 (cit. on p. 120).

Murata, Tsuyoshi and Sakiko Moriyasu (2007). "Link prediction of social networks based on weighted proximity measures." In: *Web Intelligence, IEEE/WIC/ACM international conference on*. IEEE, pp. 85–88 (cit. on p. 87).

Murata, Tsuyoshi and Sakiko Moriyasu (2008). "Link prediction based on structural properties of online social networks." In: *New Generation Computing* 26.3, pp. 245–257 (cit. on p. 87).

National Fraud Authority (2013). "Annual Fraud Indicator 2013." In: (cit. on pp. 100, 101).

Newman, Mark (2010). *Networks: an introduction*. Oxford University Press (cit. on pp. 66, 85).

Newman, Mark EJ (2001a). "Scientific collaboration networks. I. Network construction and fundamental results." In: *Physical review E* 64.1, p. 016131 (cit. on pp. 28, 30, 42).

Newman, Mark EJ (2001b). “Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality.” In: *Physical review E* 64.1, p. 016132 (cit. on pp. 11, 25, 28, 30, 42).

Newman, Mark EJ (2001c). “Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality.” In: *Physical review E* 64.1, p. 016132 (cit. on p. 70).

Newman, Mark EJ (2003). “The structure and function of complex networks.” In: *SIAM review* 45.2, pp. 167–256 (cit. on p. 69).

Ngai, EWT, Yong Hu, YH Wong, Yijun Chen, and Xin Sun (2011). “The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature.” In: *Decision Support Systems* 50.3, pp. 559–569 (cit. on p. 102).

Opsahl, Tore (2011). “Triadic closure in two-mode networks: Redefining the global and local clustering coefficients.” In: *Social Networks* (cit. on pp. 11, 42).

Organisation for Economic Co-operation and Development (2013). *Tax and development themes in recent G20 discussion*. Communique. OECD. URL: `http://search.oecd.org/officialdocuments/publicdisplaydocumentpdf/?cote=DCD/DAC/RD(2013)13/RD2&docLanguage=En` (cit. on p. 100).

Padrón, Benigno, Manuel Nogales, and Anna Traveset (2011). “Alternative approaches of transforming bimodal into unimodal mutualistic networks. The usefulness of preserving weighted information.” In: *Basic and Applied Ecology* 12.8, pp. 713–721 (cit. on pp. 19, 66).

Perlich, Claudia and Foster Provost (2006). “Distribution-based aggregation for relational learning with identifier attributes.” In: *Machine Learning* 62.1-2, pp. 65–105 (cit. on pp. 44, 45, 86, 101, 102, 106, 107).

Phua, Clifton, Vincent Lee, Kate Smith, and Ross Gayler (2010). “A comprehensive survey of data mining-based fraud detection research.” In: *arXiv preprint arXiv:1009.6119* (cit. on p. 102).

Provost, Foster, Brian Dalessandro, Rod Hook, Xiaohan Zhang, and Alan Murray (2009). “Audience selection for on-line brand advertising: privacy-friendly social network targeting.” In: *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 707–716 (cit. on pp. 19, 44, 86).

Provost, Foster and Tom Fawcett (2013). “Data Science for Business: What you need to know about data mining and data-analytic thinking.” In: (cit. on pp. 3, 5, 7–9, 23, 79).

Provost, Foster and Venkateswarlu Kolluri (1999). “A survey of methods for scaling up inductive algorithms.” In: *Data mining and knowledge discovery* 3.2, pp. 131–169 (cit. on p. 29).

Provost, Foster, David Martens, and Alan Murray (2012). “Geo-social network advertising.” In: *2012 Winter Conference on Business Intelligence*. Snowbird, Utah (cit. on pp. 11, 13, 19, 25, 26, 30, 44).

Provost, Foster, David Martens, and Alan Murray (2015a). “Finding Similar Mobile Consumers with a Privacy-Friendly Geo-Social Design.” In: *Information Systems Research* In Press (cit. on pp. 119, 121).

Provost, Foster, David Martens, and Alan Murray (2015b). "Finding Similar Mobile Consumers with a Privacy-Friendly Geosocial Design." In: *Information Systems Research* 26.2, pp. 243–265 (cit. on pp. 69, 79, 85, 86).

Putnam, Robert D (1995). "Bowling alone: America's declining social capital." In: *Journal of democracy* 6.1, pp. 65–78 (cit. on p. 120).

Raeder, Troy, Ori Stitelman, Brian Dalessandro, Claudia Perlich, and Foster Provost (2012). "Design principles of massive, robust prediction systems." In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 1357–1365 (cit. on pp. 14, 18, 119, 121).

Ranjan, Jayanthi, DP Goyal, and SI Ahson (2008). "Data mining techniques for better decisions in human resource management systems." In: *International Journal of Business Information Systems* 3.5, pp. 464–481 (cit. on p. 146).

Rennie, Jason DM and Ryan Rifkin (2001). "Improving multiclass text classification with the support vector machine." In: (cit. on p. 136).

Robins, Garry and Malcolm Alexander (2004). "Small worlds among interlocking directors: Network structure and distance in bipartite graphs." In: *Computational & Mathematical Organization Theory* 10.1, pp. 69–94 (cit. on p. 42).

Rousseau, Jean-Jacques (1762). *The Social Contract, Or Principles of Political Right (Du contrat social ou Principes du droit politique)* (cit. on p. 100).

Rrnyi, Alfred (1961). "On measures of entropy and information." In: *Fourth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 547–561 (cit. on p. 39).

Rudin, Cynthia (2009). "The P-Norm Push: A simple convex ranking algorithm that concentrates at the top of the list." In: *The Journal of Machine Learning Research* 10, pp. 2233–2271 (cit. on p. 109).

Sahin, Y and E Duman (2011). "Detecting credit card fraud by decision trees and support vector machines." In: *Proceedings of the International MultiConference of Engineers and Computer Scientists*. Vol. 1 (cit. on p. 108).

Sánchez, Daniel, MA Vila, L Cerda, and José-Marıa Serrano (2009). "Association rules applied to credit card fraud detection." In: *Expert Systems with Applications* 36.2, pp. 3630–3640 (cit. on pp. 102, 103).

Scholkopf, Bernhard and Alexander J Smola (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press (cit. on pp. 4, 6).

Schreiner, Mark (2000). "Credit scoring for microfinance: Can it work?" In: *Journal of Microfinance/ESR Review* 2.2, pp. 105–118 (cit. on pp. 118, 120).

Schreiner, Mark (2003). *Scoring: the next breakthrough in microcredit*. Consultative group to assist the poorest (CGAP) (cit. on pp. 118–121).

Seierstad, Cathrine and Tore Opsahl (2011). "For the few not the many? The effects of affirmative action on presence, prominence, and social capital of women directors in Norway." In: *Scandinavian Journal of Management* 27.1, pp. 44–54 (cit. on pp. 11, 34, 39).

Sharma, Manohar and Manfred Zeller (1997). "Repayment performance in group-based credit programs in Bangladesh: An empirical analysis." In: *World development* 25.10, pp. 1731–1742 (cit. on p. 120).

Shaw, Michael J, Chandrasekar Subramaniam, Gek Woo Tan, and Michael E Welge (2001). "Knowledge management and data mining for marketing." In: *Decision support systems* 31.1, pp. 127–137 (cit. on p. 4).

Shmueli, Galit (2016). "Analyzing Behavioral Big Data: Methodological, Practical, Ethical, and Moral Issues." In: *Practical, Ethical, and Moral Issues (February 9, 2016)* (cit. on pp. 145, 147, 148).

Soni, Jyoti, Ujma Ansari, Dipesh Sharma, and Sunita Soni (2011). "Predictive data mining for medical diagnosis: An overview of heart disease prediction." In: *International Journal of Computer Applications* 17.8, pp. 43–48 (cit. on p. 4).

Stitelman, Ori, Claudia Perlich, Brian Dalessandro, Rod Hook, Troy Raeder, and Foster Provost (2013). "Using Co-Visitation Networks For Classifying Non-Intentional Traffic." In: (cit. on pp. 102, 103).

Su, Xiaoyuan and Taghi M Khoshgoftaar (2009). "A survey of collaborative filtering techniques." In: *Advances in artificial intelligence* 2009, p. 4 (cit. on pp. 85, 86).

Sun, Jimeng, Huiming Qu, Deepayan Chakrabarti, and Christos Faloutsos (2005). "Neighborhood formation and anomaly detection in bipartite graphs." In: *Data Mining, Fifth IEEE International Conference on*. IEEE, 8–pp (cit. on p. 43).

Thomas, Lyn C (2009). *Consumer Credit Models: Pricing, Profit and Portfolios: Pricing, Profit and Portfolios*. Oxford University Press (cit. on p. 108).

Thomas, Lyn C, David B Edelman, and Jonathan N Crook (2002). *Credit scoring and its applications*. Siam (cit. on p. 108).

Ugander, Johan, Brian Karrer, Lars Backstrom, and Cameron Marlow (2011). "The anatomy of the facebook social graph." In: *arXiv preprint arXiv:1111.4503* (cit. on pp. 124, 127).

Van Gestel, Tony, Bart Baesens, and David Martens (2015). *Predictive Analytics: Techniques and Applications in Credit Risk Modelling*. Oxford University Press, p. 691 (cit. on p. 121).

Van Gestel, Tony, David Martens, Bart Baesens, Daniel Feremans, Johan Huysmans, and Jan Vanthienen (2007). "Forecasting and analyzing insurance companies' ratings." In: *International Journal of Forecasting* 23.3, pp. 513–529 (cit. on p. 137).

Van Gool, Joris, Wouter Verbeke, Piet Sercu, and Bart Baesens (2012). "Credit scoring for microfinance: is it worth it?" In: *International Journal of Finance & Economics* 17.2, pp. 103–123 (cit. on pp. 118, 120).

Vapnik, Vladimir (2013). *The nature of statistical learning theory*. Springer Science & Business Media (cit. on p. 137).

Verbeke, Wouter, David Martens, and Bart Baesens (2014). "Social network analysis for customer churn prediction." In: *Applied Soft Computing* 14, Part C, pp. 431–446. URL: `http://www.sciencedirect.com/science/article/pii/S1568494613003116` (cit. on pp. 4, 121, 146).

Vert, Jean-Philippe, Koji Tsuda, and Bernhard Schölkopf (2004). "A primer on kernel methods." In: *Kernel Methods in Computational Biology*, pp. 35–70 (cit. on p. 74).

Wallace, Byron C, Kevin Small, Carla E Brodley, and Thomas A Trikalinos (2011). "Class imbalance, redux." In: *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, pp. 754–763 (cit. on p. 109).

Wasserman, Stanley (1994). *Social network analysis: Methods and applications*. Vol. 8. Cambridge University Press (cit. on pp. 66, 70, 85).

Weber, Ingmar, Venkata Rama Kiran Garimella, and Erik Borra (2013). "Inferring audience partisanship for youtube videos." In: *Proceedings of the 22nd international conference on World Wide Web companion*. International World Wide Web Conferences Steering Committee, pp. 43–44 (cit. on pp. 14, 18, 121).

Weber, Roger, Hans-Jörg Schek, and Stephen Blott (1998). "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces." In: *VLDB*. Vol. 98, pp. 194–205 (cit. on p. 20).

Wei, Yanhao, Pinar Yildirim, Christophe Van den Bulte, and Chrysanthos Dellarocas (2014). "Credit Scoring with Social Network Data." In: *Available at SSRN 2475265* (cit. on p. 121).

Whitrow, Christopher, David J Hand, Piotr Juszczak, D Weston, and Niall M Adams (2009). "Transaction aggregation as a strategy for credit card fraud detection." In: *Data Mining and Knowledge Discovery* 18.1, pp. 30–55 (cit. on p. 102).

Witten, Ian H and Eibe Frank (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann (cit. on p. 5).

Wolpert, DH (1992). "Stacked generalization." In: *Neural networks*. URL: `http://www.sciencedirect.com/science/article/pii/S0893608005800231` (cit. on p. 110).

Wu, Roung-Shiunn, Chin-Shyh Ou, Hui-ying Lin, She-I Chang, and David C Yen (2012). "Using data mining technique to enhance tax evasion detection performance." In: *Expert Systems with Applications* 39.10, pp. 8769–8777 (cit. on pp. 102, 103).

Wu, Xindong, Vipin Kumar, J Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J McLachlan, Angus Ng, Bing Liu, S Yu Philip, et al. (2008). "Top 10 algorithms in data mining." In: *Knowledge and Information Systems* 14.1, pp. 1–37 (cit. on p. 128).

Yang and Liu (1999). "A re-examination of text categorization methods." In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 42–49 (cit. on p. 136).

Young, H Peyton and Arthur Levenglick (1978). "A consistent extension of Condorcet's election principle." In: *SIAM Journal on applied Mathematics* 35.2, pp. 285–300 (cit. on pp. 36, 82).

Yu, Hsiang-Fu, Hung-Yi Lo, Hsun-Ping Hsieh, Jing-Kai Lou, Todd G McKenzie, Jung-Wei Chou, Po-Han Chung, Chia-Hua Ho, Chun-Fu Chang, Yin-Hsuan Wei, et al. (2010). "Feature engineering and classifier ensemble for KDD cup 2010." In: *Proceedings of the KDD Cup 2010 Workshop*, pp. 1–16 (cit. on p. 35).

Zaidi, Nayyar A, Geoffrey I Webb, Mark J Carman, and Francois Petitjean (2015). "Deep broad learning-Big models for Big data." In: *arXiv preprint arXiv:1509.01346* (cit. on p. 136).

Zeller, Manfred (1998). "Determinants of repayment performance in credit groups: The role of program design, intragroup risk pooling, and social cohesion." In: *Economic development and cultural change* 46.3, pp. 599–620 (cit. on p. 120).

Zha, Hongyuan, Xiaofeng He, Chris Ding, Horst Simon, and Ming Gu (2001). "Bipartite graph partitioning and data clustering." In: *Proceedings of the tenth international conference on Information and knowledge management*. ACM, pp. 25–32 (cit. on p. 43).

Zhang, Wen, Taketoshi Yoshida, and Xijin Tang (2008). "Text classification based on multi-word with support vector machine." In: *Knowledge-Based Systems* 21.8, pp. 879–886 (cit. on p. 136).

Zhou, Tao, Jie Ren, Matúš Medo, and Yi-Cheng Zhang (2007). "Bipartite network projection and personal recommendation." In: *Physical Review E* 76.4, p. 046115 (cit. on pp. 43, 44).

Ziegler, Cai-Nicolas, Sean M McNee, Joseph A Konstan, and Georg Lausen (2005). "Improving recommendation lists through topic diversification." In: *Proceedings of the 14th international conference on World Wide Web*. ACM, pp. 22–32 (cit. on pp. 11, 34, 66, 80, 85, 138).

Zweig, Katharina Anna and Michael Kaufmann (2011). "A systematic approach to the one-mode projection of bipartite graphs." In: *Social Network Analysis and Mining* 1.3, pp. 187–218 (cit. on p. 43).