Restricted Boltzmann Machine as Recommendation Model for Venture Capital

GUSTAV FREDRIKSSON

ANTON HELLSTRÖM
Restricted Boltzmann Machine as Recommendation Model for Venture Capital

GUSTAV FREDRIKSSON
ANTON HELLSTRÖM
Restricted Boltzmann Machine as Recommendation Model for Venture Capital

Anton Hellström
Gustav Fredriksson
Supervisor at KTH: Henrik Hult
Supervisor at SEB: Salla Franzén
Abstract

In this thesis, we introduce restricted Boltzmann machines (RBMs) as a recommendation model in the context of venture capital. A network of connections is used as a proxy for investors’ preferences of companies. The main focus of the thesis is to investigate how RBMs can be implemented on a network of connections and investigate if conditional information can be used to boost RBMs.

The network of connections is created by using board composition data of Swedish companies. For the network, RBMs are implemented with and without companies’ place of origin as conditional data, respectively. The RBMs are evaluated by their learning abilities and their ability to recreate withheld connections.

The findings show that RBMs perform poorly when used to recreate withheld connections but can be tuned to acquire good learning abilities. Adding place of origin as conditional information improves the model significantly and show potential as a recommendation model, both with respect to learning abilities and the ability to recreate withheld connections.
Restricted Boltzmann Machine som Rekommendationsmodell för Riskkapital

Sammanfattning

Denna studie introducerar restricted Boltzmann machines (RBMs) som rekommendationsmodell i kontexten av riskkapital. Ett nätverk av relationer används som proxy för att modellera investerares bolagspreferenser. Studiens huvudfokus är att undersöka hur RBMs kan implementeras för ett dataset bestående av relationer mellan personer och bolag, samt att undersöka om modellen går att förbättra genom att tillföra av ytterligare information.

Nätverket skapas från styrelsensammansättningar för svenska bolag. För nätverket implementeras RBMs både med och utan den extra informationen om bolagens ursprungsort. Vardera RBM-modell undersöks genom att utvärdera dess inlärningsförmåga samt förmåga att återskapa manuellt gömda relationer.

Resultatet påvisar att RBM-modellerna har en bristfällig förmåga att återskapa borttagna relationer, dock noteras god inlärningsförmåga. Genom att addera ursprungsort som extra information förbättras modellerna markant och god potential som rekommendationsmodell går att urskilja, både med avseende på inlärningsförmåga samt förmåga att återskapa gömda relationer.
Contents

1 Acknowledgement ........................................... 1

2 Introduction .................................................. 1
2.1 Background .................................................. 1
  2.1.1 Venture Capital Dynamics .............................. 2
  2.1.2 Collaborative Filtering .................................. 4
  2.1.3 Previous Research ...................................... 5
2.2 Objective ................................................... 5
2.3 Delimitation ................................................ 6
2.4 Outline ...................................................... 7

3 Mathematical Theory ......................................... 8
3.1 Boltzmann Machine .......................................... 8
3.2 Restricted Boltzmann Machine ............................ 9
3.3 Conditional Restricted Boltzmann Machine .............. 12
3.4 Learning .................................................... 13
  3.4.1 Gradient Ascent ......................................... 13
  3.4.2 Gibbs Sampling .......................................... 16
  3.4.3 Contrastive Divergence ................................ 17
  3.4.4 Contrastive Divergence for CRBM .................... 18
3.5 Making Predictions ......................................... 19

4 Method ......................................................... 21
4.1 Data ........................................................ 21
  4.1.1 Data Preprocessing ..................................... 22
  4.1.2 Training & Test Set ..................................... 24
4.2 Models & Learning .......................................... 25
  4.2.1 Missing Values .......................................... 25
  4.2.2 Learning ................................................. 26
  4.2.3 Model 1: RBM ............................................ 27
  4.2.4 Model 2: CRBM (Place of Origin) .................... 27
Chapter 1

Acknowledgement

We would like to express our sincere gratitude to Salla Franzén, Chief Data Scientist at SEB, for her valuable thoughts and input during the project. Her willingness to support and help as well as setting up meetings with people in her network has been highly appreciated. We would also like to thank SEB for providing us with the data for this project. Finally, special thanks to our supervisor at KTH, Prof. Henrik Hult, for his ideas, feedback and professional guidance that has been essential throughout the entire project.
Chapter 2

Introduction

This section aim to introduce and give a background to this thesis topic and declare the scientific contribution. In particular, venture capital dynamics will be introduced, giving an idea why a recommendation model can be of interest. Furthermore, previous research in collaborative filtering and relevant inferences will be reviewed. Finally, the objective and delimitation will be highlighted, setting the framework of the conducted research.

2.1 Background

Venture capitalists (VCs) are major players in the private equity industry. During Q3 2018 VC-backed companies raised $52 billion across 3,045 deals [26]. Pursuing the next Dropbox, Airbnb or Uber, VCs go through a process of screening and decision making based on non-generic data, that may or may not hold explanatory abilities. As a consequence, a great part of today’s VCs’ evaluation process often involve subjective analysis, call it gut feeling, of companies [27].

Many have studied the decision making of VCs. Yet the inferences drawn, can only confirm that the decision making is exposed to biases, rather than stating decision making criteria. In particular, personal experience and individuals’ social network tend to be decisive aspects terminating in whether a business gets backed by VCs or not. Many argue why this correctly reflects the nature of venture capital investments—some meaning it derive from an information bias, proposing one tend to treat large amount of information as good information. Comparatively, others claim it is the referrals, origin from a social network, themselves that render the nature of venture capital
investments. Likely being neither black nor white, previous research seem to unite around the idea that decision making in venture capital indeed is a matter of personal preference. Which strengthen the idea of a gut feeling or similar phenomenon [27].

From a data science perspective, it is likely the subjective criteria in venture capital that causes it to differ from other parts of finance, where data science and particularly machine learning have gained attention. The documented applications of machine learning in finance typically involve well-defined quantifiable data, see [18, 7]. Which frankly is not retrievable for venture capital, lacking clear routines and appropriate data [27]. It follows that machine learning is rarely mentioned in the context of venture capital, despite being a growing topic in finance overall.

One can hypothesise there exist investment patterns such as personal preference, being of complex character. Venture capital is however not alone in being exposed to decision criteria that is based on preferences. If one consider e-commerce and streaming services, it is possible to identify similar circumstances. Indeed, venture capital differ from e-commerce and streaming services in many aspects, yet many players in those industries have managed to implement recommendation models successfully. It is naive to not learn from their proof of concept—one can argue it is unlikely that preferences found in venture capital are of unpredictable nature.

In this thesis a seemingly unstudied angle of approach will be investigated. Namely how one could build a recommendation model for venture capital by applying machine learning. This thesis will approach the topic from two angles. On the one hand, there is a VC perspective, where we derive some aspects of a VCs work that could be supplemented by a recommendation model. On the other hand, there is a machine learning perspective, where we find great substance in exploring how collaborative filtering can be used on a network data set.

2.1.1 Venture Capital Dynamics

When discussing the concept of a recommendation model designed for VCs, it is essential to understand the incentives. Venture capital is based on making one-in-ten significantly profitable deals while making nine-in-ten break-even or losses. Previous research has shown that VCs strive to allocate most of their time evaluating if a deal will become profitable and avoid spending a vast amount of time searching for the deals. It follows that VCs want to
spend their time analysing the right businesses, the ones probable in becoming the next billion dollar start-up [16, 22].

Nonetheless, VCs conduct comprehensive screening prior to finding the deals they want to evaluate in depth. Typically the first step of the investment process, referred to as origination, involves dealing with the full universe of ventures that will show on a VC’s radar, while in pursuit of an investment opportunity. This is equivalent to the incoming deal flow, usually divided in subcategories depending on the origin. As an example, the subcategory active prospecting involve deal flow deriving from seminars, forums and other networking events [16, 22].

In parity to active prospecting, deal flow can appear unanticipated, such as various co-investment opportunities or individuals approaching VCs unsolicited with an idea. These applications rarely move on to subsequent phases of the investment process. This is however not necessarily a consequence of a deal being unsatisfactory. As likely it is the natural outcome of insufficient screening and less rigorous evaluations method that might feature screening processes in early phases [16, 22].

The origination phase is succeeded by the screening phase. This is where VCs decision criteria tend to be materialised. Four overlaying categories of evaluation criterias are mentioned in this context: entrepreneur capabilities, product or service, market or competition, and potential or expected returns if the venture is successful. That is, information one typically has by knowing the company in question. The goal of the screening phase can be considered to rather find which opportunities to not further evaluate, than what opportunities to evaluate, the former being a waste of time [16, 22].

Having screened the opportunities, the universe is reduced to only consist of opportunities that reflect most of a certain VC’s preferences. The succeeding phase is the actual test of the opportunity, known as the due diligence phase. One can imagine this is where VCs strive to spend the greater part of their effort in the investment process, since the data at this stage has to be carefully evaluated. The last phase in the investment process is the negotiation phase and is the result of a due diligence that proves favourable. The VC in question take action based on the findings from their screening and evaluation phases, trying to find investment terms that both parties are comfortable with [16, 22].

2.1.2 Collaborative Filtering

Being a sub-term to the larger term recommendation models, collaborative filtering is one of the most successful techniques using algorithms to model preference. A common approach is to let a community of individuals rate products, matching users by overlapping product ratings, translated to preferences, and thus create a recommendation model [8, 15]. Using the same idea, other approaches have been found fruitful. An example is the idea of matching products instead of users. This is referred to as item-to-item collaborative filtering, showing recommendations of products that ideally goes well with the previously bought and rated items [15].

Every system where recommendation models could be relevant is however not necessarily of the nature where there exist a given community actively reporting ratings. Consequently, there are two major branches in collaborative filtering, based on whether a system’s nature operates on explicit or implicit feedback from users [5]. Explicit feedback being defined by users consciously expressing their opinion of an item, found for scenarios such as movie ratings from Netflix users [2], article rating from Netnews readers [19] or customers rating products in e-commerce [15]. Whereas implicit feedback is defined by interpretation of users’ behaviour and making inferences about their preferences. Examples of collaborative filtering based on implicit feedback include assigning preference based on purchase history, watching habits or browsing activity, respectively [13].

As of today, the majority of research in the field focuses on explicit feedback. Aside from the trouble in determining the causal behaviour between users’ behaviour and their preference, the authors of [13] list a number of characteristics that need to be held in mind when designing a model for implicit feedback. For instance, one has to deal with the tendency that it is hard to determine negative feedback. It follows that an individual could avoid watching a movie due to lack of interest in it, but one could not reject the hypothesis that the individual simply was unaware of the movie’s existence. Another critical aspect is that one has to deal with the implicit feedback being noisy. An individual purchasing an item does not necessarily indicate a positive attitude towards the product, the product could for instance have been bought as a gift. Similarly, an individual watching the full length of the movie might have left the room forgetting about it [13].
2.1.3 Previous Research

One of the most prominent events related to collaborative filtering is the Netflix competition initiated 2006. Releasing a data set holding 100 million movie ratings, Netflix challenged the computer science community to beat their in-house model Cinematic [3]. The competition gained a lot of attention and many, see [21, 2], contributed to the research of alternative recommendation models by participating in the competition. Almost three years later, in 2008, the winner presented the solution based on restricted Boltzmann machines (RBMs), but also utilising k-nearest neighbour models, outperforming the existing model by 10.06% [3, 25].

RBMs have proven successful in many other applications as well, noteworthy publications include labeled or unlabeled images [11], and document representation by bag of words [20]. An other important use is when they form deep belief nets becoming learning modules [9, 21]. Aditionally, RBMs on conditional form can be succesfully applied to model high-dimensional sequence data such as videos or speech [23, 17]. Admittedly, RBMs have proven useful on a wide range of applications.

Despite being less researched, studies have shown how implicit feedback can be used to increase the performance of recommendation models. Evaluating the Netflix data set, conditional information can be derived from implicit feedback by a binary feature regarding which movies that users chose to rate [14, 21]. As an extension the author of [14] conclude that the major insight from the implementation is the model improvement that derive from successfully addressing implicit feedback. Thus providing with evidence that implicit feedback can significantly improve a recommendation model [14].

2.2 Objective

The objective of this thesis is to evaluate if it is possible to build a company recommendation model based on non-financial data. In particular, a network of connected individuals will be used as basis for the recommendation model. Due to data availability, a board member-company relationship will be used to build the network.

Given a network of connections, the goal is to construct a recommendation model with the ability to reconstruct withheld connections correctly and recommend new ones as well. Moreover, this thesis aim to implement RBMs
and investigate their applicability in recreating connections from the network. Additionally, the aim is to evaluate the possibility to boost RBMs by adding additional information, thus evaluating the extension of conditional restricted Boltzmann machines (CRBMs).

A set of explicit research questions investigated in the conducted research will be used:

I. Investigate how RBMs can be implemented to reconstruct a given network of connections.
   
i. With respect to tunable parameters, which aspects are due consideration?

II. Investigate if additional information can be utilised to boost an RBM.

2.3 Delimitation

In modelling the network of connections, the intuitive approach would be to study previous investments as a metric for preference. Yet such data is rare and can be of poor quality, featuring secret deals, withheld circumstances or similar noise. Consequently, an other approach will be evaluated, based on the concept of network connections between VCs. The network will be derived from board compositions. For the purpose of this thesis, the network will be treated as a given feature. In doing so, this thesis will not address the aspects of modelling VCs preference by a network of connected individuals by board activity. Nor will the assumption of a network of connections from board members be evaluated.

This thesis primary focus is the mathematical aspect of implementing RBMs as recommendation models on a network of connections, with the secondary focus to investigate the ability to reproduce connections. This thesis will be conducted in an exploratory manner, where parameter tuning will be considered but not optimised. For the purpose of implementing CRBMs, only place of origin will be considered as additional information used to boost RBMs.
2.4 Outline

Firstly, the mathematical aspect of the conducted research will be derived and presented. Secondly, a network of personal connections is suggested using board composition data. Thirdly, the practical implementation of RBMs and CRBMs will be carefully explained. Lastly, the results from the modelling will be presented and discussed, including conclusions and future work.
Chapter 3
Mathematical Theory

This section aim to elaborate the mathematical aspect of Boltzmann machines applied in this thesis context. The line of thought will proceed from the standard Boltzmann machine, restricted Boltzmann machines and conditional restricted Boltzmann machines. Subsequently, the learning process will be derived as well as how predictions are made.

3.1 Boltzmann Machine

A Boltzmann machine is a graphical model that takes the shape of a network of bidirectional links between nodes taking stochastic decisions of a binary state [1, 10]. The interpretation of each state is that an on-state indicates that the system of nodes accepts some hypothesis regarding the domain, while an off-state indicate the opposite. Additionally, bidirectional links are assigned with weights, creating a pairwise constraint between hypotheses. Having a positive weight between nodes is interpreted as hypotheses that tend to support one another. That is, accepting the one if the other is accepted is likely, given that their bidirectional link is positive. Equivalently, a negative weight causes the opposite [1].

The structure of a Boltzmann machine on standard form is defined by its energy,

\[ E(v) = -\left( \sum_i \theta_i s_i + \sum_{i<j} s_i s_j w_{ij} \right), \]  

(3.1)

with contribution from each node, see Figure 3.1 \( \theta_i \) is the bias with respect to node \( i \), \( w_{ij} \) is the weight between node \( i \) and node \( j \), and \( s_i \) is the binary
state for node $i$ by state vector $v_i$. The probability of being in state $v$ is given by the Boltzmann distribution

$$p(v) = \frac{1}{Z} \exp(-E(v)), \quad (3.2)$$

where $Z$ is the partition function that ensures that the probability aggregates to one,

$$Z = \sum_v \exp(-E(v)). \quad (3.3)$$

Learning a Boltzmann machine is equivalent to minimising the global energy, whereas the configuration rendering the lowest energy will be the configuration that is most compatible with the entity of the data set used for learning. That is, the system can be steered toward a configuration that satisfy the constraints to the best degree. Furthermore, the energy can be interpreted as a measurement of how greatly the constraints stated by the problem domain are satisfied. Acknowledge that (3.1) for any input vector $v$ is a function of the tuneable parameters $\theta_i$ and $w_{ij}$.  

### 3.2 Restricted Boltzmann Machine

As an extension of the Boltzmann machine, RBMs uses a restriction on the connectivity between the nodes. The nodes is divided into two layers, a visible layer, $V = (v_1, \ldots, v_M)$, representing the observations and a hidden layer, $h = (h_1, \ldots, h_J)$, representing the dependencies between the observed nodes. In a Boltzmann machine every node can connect to every other node,
independent of layer. For RBMs, the connections are restricted to only appear between the layers, hidden nodes are only connected to visible nodes and vice versa, see Figure 3.2 [1].

![Figure 3.2: An example of a restricted Boltzmann machine with 2 hidden nodes $h_j$, $j = 1, 2$, together forming the hidden layer and 3 visible nodes $v_i$, $i = 1, 2, 3$, together forming the visible layer](image)

Corresponding to the regular Boltzmann Machine, the network assigns a probability for every possible configuration $(V, h)$, given by

$$p(V, h) = \frac{1}{Z} \exp(-E(V, h)), \quad (3.4)$$

where $Z$ is the partition function,

$$Z = \sum_{V, h} \exp(-E(V, h)). \quad (3.5)$$

The network’s energy, $E(V, h)$, takes various forms with respect to its generic expression (3.1), depending on the input $V$ that can take the form of a matrix or a vector. From (3.4) an expression for the probability $p(V)$ is derived as the sum over all hidden nodes, $h$, for a given vector $V$,

$$p(V) = \frac{1}{Z} \sum_{h} \exp(-E(V, h)), \quad (3.6)$$

known as the marginal probability [9].

Lacking intra-layer connections the structure of an RBM gives

$$p(V|h) = \prod_{i=1}^{n} p(v_i|h), \quad (3.7)$$

$$p(h|V) = \prod_{j=1}^{m} p(h_j|V). \quad (3.8)$$
This follows from hidden units being independent of every other hidden units
and visible units being independent of every other visible unit.

For the standard case, a Bernoulli-Bernoulli RBM, with Bernoulli distributed
visible units and Bernoulli distributed hidden units, the input \( V \) is a vector,
denoted \( v = (v_1, \ldots, v_n) \). It follows that \( v \) has binary elements \( v_i \in \{0, 1\} \)
and the energy function is on the form

\[
E(v, h) = - \sum_{i \in \text{vis.}} a_i v_i - \sum_{j \in \text{hid.}} b_j h_j - \sum_{i,j} v_i h_j w_{ij},
\]

where \( h = \{0, 1\}^m \), \( w_{ij} \) is the weight between visible unit \( i \) and hidden unit
\( j \), and \( a_i, b_j \) are their biases respectively. Note that \( v \) and \( h \) together form
the joint configuration \( (v, h) \) \[12, 9\].

Using the energy function in (3.9) the conditional probabilities used in (3.7)
and (3.8) are derived. Given input vector \( v \) it follows that the binary state
of each hidden unit, \( h_j \), is set to 1 at probability

\[
p(h_j = 1|v) = \sigma(b_j + \sum_i v_i w_{ij}),
\]

where \( \sigma(x) = 1/(1 + \exp(-x)) \) is the sigmoid \[9\]. The full derivation for
(3.10) is provided in Appendix A. Similarly, given some hidden vector \( h \) the
probability of a visible unit \( v_i \) being set to 1 is given by

\[
p(v_i = 1|h) = \sigma(a_i + \sum_j (h_j w_{ij})),
\]

which is derived analogously to (3.10) as the visible units are Bernoulli dis-
tributed. Learning a Bernoulli-Bernoulli RBM is equivalent to tuning \( a_i, b_i \)
and \( w_{ij} \) so that \( p(v) \), defined by (3.6), is maximised \[12, 9\].

Moreover, consider a multinomial-Bernoulli RBM, where each visible unit
is multinomial. The energy function takes the form

\[
E(V, h) = - \sum_{i \in \text{vis.}} \sum_{k \in \mathcal{K}} v_i^k a_i^k - \sum_{j \in \text{hid.}} h_j b_j - \sum_{i \in \text{vis.}, j \in \text{hid.}} \sum_{k \in \mathcal{K}} h_j W_{ij}^k v_i^k,
\]

where \( \mathcal{K} = \{1, \ldots, K\} \). Thus, changing the notation by denoting the input
values as a matrix $V$ where each column is a multinomial vector, $v_i$ representing the $i$th visible node. In turn, each $v_i = (v_1^i, \ldots, v_K^i)$ where $v_k^i \in \{0, 1\}$ such that $v_k^i$ takes the value 1 for at most one unique $k$ for each unique $v_i$.

Using (3.12) the conditional probabilities, (3.7) respectively (3.8) can be derived. Given input matrix $V$, the binary state of each hidden unit, $h_j$, is set to 1 at probability

$$p(h_j = 1|V) = \sigma \left( b_j + \sum_{i=1}^{M} \sum_{k=1}^{K} v_k^i W_{ij}^k \right). \quad (3.13)$$

Similarly, given some hidden vector $h$, the probability of a visible unit $v_k^i$ being set to 1 is

$$p(v_k^i = 1|h) = \frac{\exp \left( b_k^i + \sum_{j=1}^{J} h_j W_{ij}^k \right)}{\sum_{k=1}^{K} \exp \left( b_k^i + \sum_{j=1}^{J} h_j W_{ij}^k \right)}. \quad (3.14)$$

The full derivation of (3.13) and (3.14) is provided in Appendix A.

### 3.3 Conditional Restricted Boltzmann Machine

Given additional information about the domain, it is feasible to feed it to the hidden layer directly. Essentially, there are no dimension-wise limitations to what type of information that can be fed to the RBM, for example it would be possible to use a neural net, a multinomial vector or a non-linear function. Yet the mathematics will differ slightly when considering higher-dimension information. Whereas the case that will be considered here is of the form where we are able to define the information as a vector $r$, where $r_i \in \{0, 1\}$.

The additional information $r$ is fed directly to the hidden layer, without interacting with the visible layer. Consequently, adding $r$ is only supposed to affect the conditional probability of the hidden units, causing the expressions to become conditional of $r$. To fulfil these constraints the energy function of the CRBM is transformed to,

$$E(V, h) = - \sum_{i \in \text{vis.}} \sum_{k \in \mathcal{K}} v_k^i a_i^k - \sum_{j \in \text{hid.}} h_j b_j$$

$$- \sum_{i \in \text{vis.}} \sum_{j \in \text{hid}} \sum_{k \in \mathcal{K}} v_k^i W_{ij}^k h_j - \sum_{i \in \text{vis.}} \sum_{j \in \text{hid}} r_i D_{ij} h_j. \quad (3.15)$$
Considering the multinomial-Bernoulli RBM, (3.13) is expanded to
\[ p(h_j = 1|V, r) = \sigma(b_j + \sum_{i=1}^{m} \sum_{k=1}^{K} v_i^k W_{ij}^k + \sum_{i=1}^{M} r_i D_{ij}), \] (3.16)
where elements \( D_{ij} \) models the impact of \( r_i \) on \( h_j \). Indeed, the optimisation problem is extended to maximising \( p(V) \) with respect to \( a_i, b_i, W_{ij}^k \) and \( D_{ij} \).

Deriving (3.16) follows the same procedure as for the RBMs, see Appendix A. Note that the conditional probability \( p(V|h) \) is identical to (3.14), lacking connectivity to \( r \).

3.4 Learning

Learning an RBM is a matter of finding well-suited parameters in the energy function. Typically, in mathematical statistics, when the objective is to do so for a given distribution and observation, the method of maximum likelihood estimation is commonly utilised. Moreover, the likelihood is often transformed into log-likelihood of the observations to simplify computations.

For a given data set of observations \( X = \{x_1, \ldots, x_l\} \) and some set of parameters being optimised, \( \theta \), the likelihood function of a marginal distribution \( p(x|\theta) \) is given by
\[ L(\theta|X) = \prod_{i=1}^{l} p(x_i|\theta). \] (3.17)
Maximising (3.17) being equivalent to maximising the log-likelihood
\[ \ln L(\theta|X) = \sum_{i=1}^{l} \ln p(x_i|\theta). \] (3.18)
However, (3.18) is generally intractable and lacks analytical solution when dealing with RBMs. Instead optimising the marginal distribution for an RBM requires a numerical approach [21].

3.4.1 Gradient Ascent

Gradient ascent is an iterative optimisation algorithm for maximising functions, typically used when an analytic solution is not feasible. In the context of finding optimal parameters with respect to some function, gradient ascent
means adjusting the parameters using the gradient of the function \[24\].

Using the same notation as in Chapter 3.4 where \( \mathbf{X} = \{ \mathbf{x}_1, \ldots, \mathbf{x}_l \} \) is a set of observations and \( \theta \) is some set of parameters being optimised found by iteration,

\[
\theta^{(n+1)} = \theta^{(n)} + \epsilon \frac{\partial \ln \mathcal{L}(\theta^{(n)}|\mathbf{X})}{\partial \theta^{(n)}} \equiv \Delta \theta = \epsilon \frac{\partial \ln \mathcal{L}(\theta^{(n)}|\mathbf{X})}{\partial \theta^{(n)}}. \tag{3.19}
\]

Note that \( n \in \mathbb{N} \) is the iteration number and \( n = 0 \) is the initial state for parameter \( \theta \) and \( \epsilon > 0 \) is the learning rate. The learning rate sets the step size of each update of the parameters and needs to be carefully chosen to make the optimisation algorithm efficient. The updates are iterated until a pre-determined breaking point or degree of convergence \[24\].

In the context of an RBM, given a data set of \( N \) observations as \( \mathcal{S} = \{ \mathbf{V}_1, \ldots, \mathbf{V}_N \} \), and the set of parameters desired to optimise as \( \theta \), the gradient in (3.19), is found by inserting (3.6) in (3.17). For one observation \( \mathbf{V} \in \mathcal{S} \),

\[
\frac{\partial \ln \mathcal{L}(\theta|\mathbf{V})}{\partial \theta} = \frac{\partial \ln p(\mathbf{V}|\theta)}{\partial \theta} = \frac{\partial}{\partial \theta} \ln \left( \frac{1}{Z} \sum_h \exp(-E(\mathbf{V}, h)) \right) = \frac{\partial}{\partial \theta} \left( \ln \sum_h \exp(-E(\mathbf{V}, h)) \right) - \frac{\partial \ln Z}{\partial \theta}. \tag{3.20}
\]

By inserting (3.5) in the second term of (3.20) and using the chain rule, (3.20) becomes

\[
\frac{\partial \ln p(\mathbf{V}|\theta)}{\partial \theta} = \frac{\partial}{\partial \theta} \left( \ln \sum_h \exp(-E(\mathbf{V}, h)) \right) - \frac{\partial}{\partial \theta} \left( \ln \sum_{\mathbf{V}, h} \exp(-E(\mathbf{V}, h)) \right) \\
= - \frac{1}{\sum_{\mathbf{V}', h'} \exp(-E(\mathbf{V}', h'))} \sum_h \exp(-E(\mathbf{V}, h)) \frac{\partial}{\partial \theta} E(\mathbf{V}, h) \\
+ \frac{1}{\sum_{\mathbf{V}', h'} \exp(-E(\mathbf{V}', h'))} \sum_{\mathbf{V}, h} \exp(-E(\mathbf{V}, h)) \frac{\partial}{\partial \theta} E(\mathbf{V}, h) \\
= - \sum_h \sum_{h'} \exp(-E(\mathbf{V}, h')) \frac{\partial}{\partial \theta} E(\mathbf{V}, h) + \frac{1}{Z} \sum_{\mathbf{V}, h} \exp(-E(\mathbf{V}, h)) \frac{\partial}{\partial \theta} E(\mathbf{V}, h) \tag{3.21}
\]
The expression in (3.21) can be re-written using
\[ p(h|V) = \frac{1}{Z} \exp(-E(V, h)) = \sum_h \exp(-E(V, h)) \] (3.22)
followed by (3.4), giving
\[
\frac{\partial \ln p(V|\theta)}{\partial \theta} = - \sum_h p(h|V) \frac{\partial}{\partial \theta} E(V, h) + \sum_{V, h} p(V, h) \frac{\partial}{\partial \theta} E(V, h) \] (3.23)
\[
= -E_p(h|V) \left[ \frac{\partial E(V, h)}{\partial \theta} \right] + E_{p(V, h)} \left[ \frac{\partial E(V, h)}{\partial \theta} \right]. \] (3.24)

Depending on the system’s energy function, (3.24) is evaluated for an RBM using (3.9),
\[
E(V, h) = - \sum_{i \in \text{vis.}} \sum_{k \in K} v_i^k a_i^k - \sum_{j \in \text{hid.}} h_j b_j - \sum_{i \in \text{vis.}} \sum_{j \in \text{hid}} \sum_{k \in K} W_{ij}^k h_j v_i^k
\]
the partial derivatives in (3.24) with respect to each parameter is given by
\[
\frac{\partial E(V, h)}{\partial W_{ij}^k} = -h_j v_i^k, \quad \frac{\partial E(V, h)}{\partial a_i^k} = -v_i^k, \quad \frac{\partial E(V, h)}{\partial b_j} = -h_j. \] (3.25)

As described in (3.3), the energy function changes for the CRBM by one additional term. Thus yielding an additional derivative to evaluate in the log-likelihood gradient,
\[
\frac{\partial E(V, h)}{\partial D_{ij}} = -h_j r_i. \] (3.26)

By Inserting (3.25) and (3.26) in (3.24) the log-likelihood gradient can be derived,
\[
\Delta W_{ij}^k = \epsilon \frac{\partial \ln p(V|W_{ij}^k)}{\partial W_{ij}^k} = \epsilon(\langle v_i^k h_j \rangle_{data} - \langle v_i^k h_j \rangle_{model}), \] (3.27)
\[
\Delta a_i^k = \epsilon \frac{\partial \ln p(V|a_i^k)}{\partial a_i^k} = \epsilon(\langle v_i^k \rangle_{data} - \langle v_i^k \rangle_{model}), \] (3.28)
\[
\Delta b_j = \epsilon \frac{\partial \ln p(V|b_j)}{\partial b_j} = \epsilon(\langle h_j \rangle_{data} - \langle h_j \rangle_{model}), \] (3.29)
\[
\Delta D_{ij} = \epsilon \frac{\partial \ln p(V|D_{ij})}{\partial D_{ij}} = \epsilon(\langle h_j \rangle_{data} - \langle h_j \rangle_{model}) r_i, \] (3.30)
where \( \langle \cdot \cdot \cdot \rangle_{\text{data}} \) and \( \langle \cdot \cdot \cdot \rangle_{\text{model}} \) denotes the expected value driven by the data and the model respectively. The expression in (3.24) consist of two expected values. The first term is identified as the value of the energy under the conditional distribution of the hidden layer driven by the observed data. Comparatively, the second term is the expected energy taken with respect to the distribution defined by the model. Typically, the second term is computation intensive as can be seen by using (3.25) and (3.26) in (3.24),

\[
\frac{\partial \ln p(V|W_k)}{\partial W_{kij}} = p(h_j = 1|V)v_i^k - \sum_V p(V)p(h_j = 1|V)v_i^k, \tag{3.31}
\]

\[
\frac{\partial \ln p(V|a_k)}{\partial a_k} = v_i^k - \sum_V p(V)v_i^k, \tag{3.32}
\]

\[
\frac{\partial \ln p(V|h_j)}{\partial b_j} = p(h = 1|V) - \sum_V p(V)p(h = 1|V) \tag{3.33}
\]

\[
\frac{\partial \ln p(V|D_{ij})}{\partial D_{ij}} = p(h_j = 1|V)r_i - \sum_V p(V)p(h_j = 1|V)r_i \tag{3.34}
\]

which can be dealt with by approximating it using contrastive divergence [9]. Full derivations are provided in Appendix B.

### 3.4.2 Gibbs Sampling

An essential part of contrastive divergence uses Gibbs sampling. When a joint distribution is unknown or difficult to sample from, Gibbs sampling is particularly useful. The idea of Gibbs sampling is to utilise the conditional distribution of every involved variable and progressively sample each variable, conditioned on every other variable. Gibbs sampling is by definition a Markov chain Monte Carlo method, sampling from probability distributions [6, 4].

Let \( X = (X_1, \ldots, X_n) \) be a random variable with the joint distribution \( p(x_1, \ldots, x_n) \) and \( x^{(i)} = (x_1^{(i)}, \ldots, x_n^{(i)}) \) as the \( i \)th sample of \( X \). The general algorithm of the sampler is an iterative process which can be describe with below steps for iteration \( i \):

1. Begin with an initial sample \( x^{(i-1)} \)
2. Sample each variable, of \( x^{(i)} \) where \( x_j^{(i)} \) is sampled from the conditional distribution \( p(X_j = x_j|x_1^{(i)}, \ldots, x_j^{(i)}, x_{j+1}^{(i-1)}, \ldots, x_n^{(i-1)}) \), for \( j = 1, \ldots, n \)
For any desired number of samples, $t$, these steps are repeated. The result is $t$ samples with dimension $n$. In training an RBM the Gibbs sampling becomes tractable due to the independence of the conditional probability in each layer. Therefore, given one sample of observed data the hidden states can be sampled in parallel and it is not required to sample every hidden node in turn [21].

### 3.4.3 Contrastive Divergence

The algorithm known as contrastive divergence is used to approximate the log-likelihood gradient in cases where the gradient is otherwise intractable [21]. Usually estimates of log-likelihood using Markov chain Monte Carlo methods require a vast number of steps, $t$, in order to achieve an unbiased estimate. However, the authors of [6] have been able to show that it is feasible to find a sufficiently accurate approximation using only few steps, in the learning of an RBM. As a result, the algorithm of contrastive divergence have become a common way to train an RBM. In practice, the number of steps, $t$, that the Gibbs sampling will be conducted is pre-determined. Thus retrieving a different estimation of the gradient than running the Gibbs sampler until convergence would [6].

The algorithm is initiated from a training sample, $V^{(0)}$, for the visible states in an RBM. Iteratively for $t$ steps, the states of the hidden layer, $h^{(i)}$ is sampled from the conditional distribution $p(h|V^{(i)})$, given the training sample. Next, $V^{(i+1)}$ is sampled from the conditional distribution $p(V|h^{(i)})$. The final output is a new reconstructed sample, $V^{(t)}$, after $t$ steps.

After $t$-steps Gibbs sampling from the training sample $V$ the log-likelihood gradient in (3.23), can be approximated as

$$
\frac{\partial \ln p(V^{(0)}|\theta)}{\partial \theta} \approx -\sum_h p(h|V^{(t)}) \frac{\partial E(V^{(0)}, h)}{\partial \theta} + \sum_h p(h|V^{(t)}) \frac{\partial E(V^{(t)}, h)}{\partial \theta}
$$

(3.35)
and the learning rules (3.27), (3.28), and (3.29) is approximated by

$$
\Delta W^k_{ij} = \epsilon \frac{\partial \ln p(V | \theta)}{\partial W^k_{ij}} = \epsilon (\langle v_i^k h_j \rangle_{\text{data}} - \langle v_i^k h_j \rangle_{\text{reconstruct}}), \quad (3.36)
$$

$$
\Delta a^k_i = \epsilon \frac{\partial \ln p(V | \theta)}{\partial a^k_i} = \epsilon (\langle v_i^k \rangle_{\text{data}} - \langle v_i^k \rangle_{\text{reconstruct}}), \quad (3.37)
$$

$$
\Delta b_j = \epsilon \frac{\partial \ln p(V | \theta)}{\partial b_j} = \epsilon (\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{reconstruct}}). \quad (3.38)
$$

For a training set, \( \mathcal{D} \), using one batch, the algorithm of contrastive divergence using \( t \) steps is described by the pseudocode in Algorithm 1.

**Algorithm 1** Contrastive divergence, \( t \) steps

**Input:** RBM and training set \( \mathcal{D} \)

**Output:** Gradient updates \( \Delta a \), \( \Delta b \) and \( \Delta W \)

1: for \( V \in \mathcal{D} \) do
2: \( V^{(0)} \leftarrow V \)
3: for \( \ell = 1, \ldots, t \) do
4: \( \text{Sample } h^{(\ell)} \sim p(h|V^{(\ell)}) \)
5: \( \text{Sample } V^{(\ell+1)} \sim p(V|h^{(\ell)}) \)
6: for \( i = 1, \ldots, M \) and \( j = 1, \ldots, J \) and \( k = 1, \ldots, K \) do
7: \( \Delta W^k_{ij} \leftarrow \Delta W^k_{ij} + p(h_j = 1|V^{(0)}) (v_i^k)^{(0)} - p(h_j = 1|V^{(t)}) (v_i^k)^{(t)} \)
8: for \( i = 1, \ldots, M \), and \( k = 1, \ldots, K \) do
9: \( \Delta a^k_i \leftarrow \Delta a^k_i + (v_i^k)^{(0)} - (v_i^k)^{(t)} \)
10: for \( j = 1, \ldots, J \) do
11: \( \Delta b_j \leftarrow \Delta b_j + p(h_j = 1|V^{(0)}) - p(h_j = 1|V^{(k)}) \)
12: return \( \Delta a \), \( \Delta b \) and \( \Delta W \)

3.4.4 Contrastive Divergence for CRBM

Adding a conditional vector \( r \) to the RBM only slightly changes the learning algorithm presented in Algorithm 1. The probability \( p(h_j = 1|V^{(t)}) \) becomes \( p(h_j = 1|V^{(t)}, r) \), whereas the additional parameter \( D \) is updated at the end of the algorithm. By using (3.30) and (3.35), the learning rule for \( D \) becomes

$$
\Delta D_{ij} = \epsilon \frac{\partial \ln p(V | \theta)}{\partial D_{ij}} = \epsilon (\langle h_j \rangle_{\text{data}} - \langle h_j \rangle_{\text{reconstruct}}) r_i. \quad (3.39)
$$

The algorithm of contrastive divergence using \( t \) steps for a CRBM is described by the pseudocode in Algorithm 2.
Algorithm 2 Contrastive divergence, $t$ steps for a CRBM

**Input:** RBM and training set $\mathcal{D}$

**Output:** Gradient updates $\Delta a$, $\Delta b$, $\Delta W$ and $\Delta D$

1: for $V \in \mathcal{D}$ do
2: \hspace{1em} $V(0) \leftarrow V$
3: \hspace{1em} for $\ell = 1, \ldots, t$ do
4: \hspace{2em} Sample $h^{(\ell)} \sim p(h|V^{(\ell)}, r)$
5: \hspace{2em} Sample $V^{(\ell+1)} \sim p(V|h^{(\ell)})$
6: \hspace{1em} for $i = 1, \ldots, M$ and $j = 1, \ldots, J$ and $k = 1, \ldots, K$ do
7: \hspace{2em} \hspace{1em} $\Delta W_{ij}^k \leftarrow \Delta W_{ij}^k + p(h_j = 1|V^{(0)}, r) (v_i^k)^{(0)} - p(h_j = 1|V^{(t)}, r) (v_i^k)^{(t)}$
8: \hspace{1em} for $i = 1, \ldots, M$ do
9: \hspace{2em} $\Delta a^k_i \leftarrow \Delta a^k_i + (v_i^k)^{(0)} - (v_i^k)^{(t)}$
10: for $j = 1, \ldots, J$ do
11: \hspace{1em} $\Delta b_j \leftarrow \Delta b_j + p(h_j = 1|V^{(0)}, r) - p(h_j = 1|V^{(t)}, r)$
12: for $i = 1, \ldots, M$ and $j = 1, \ldots, J$ do
13: \hspace{2em} $\Delta D_{ij} \leftarrow \Delta D_{ij} + p(h_j = 1|V^{(0)}, r) - p(h_j = 1|V^{(t)}, r)r_i$
14: \hspace{1em} return $\Delta a$, $\Delta b$, $\Delta W$ and $\Delta D$

### 3.5 Making Predictions

Given a trained RBM, predictions are made by feeding the RBM some input vector $V$ and use same idea as for the algorithm of contrastive divergence with $t = 1$ but without sampling or updating the weights. As a result, a vector of probabilities is retrieved, which is used to stipulate the predictions. Particularly, the prediction for some visible node $x$ given the input vector $V$ is determined by

$$q_j = p(h_j = 1|V) = \sigma(b_j + \sum_{i=1}^{M} \sum_{k=1}^{K} v_i^k W_{ij}^k), \quad (3.40)$$

from which it follows that

$$p(v_x^k = 1|q) = \frac{\exp \left( b_x^k + \sum_{j=1}^{J} q_j W_{xj}^k \right)}{\sum_{k=1}^{K} \exp \left( b_x^k + \sum_{j=1}^{J} q_j W_{xj}^k \right)}. \quad (3.41)$$

Using the probability in (3.41), predictions can be stipulated. In this thesis two different approaches are highlighted. Firstly, the predictions can be
derived using the expected value over all $k$,

$$y^\text{pred}_q = \mathbb{E}[v^k_x] = \sum_{k=1}^K R_k p(v^k_x = 1|q) \quad (3.42)$$

where $R_k$ denotes the $k$th value [21]. Secondly, the predictions can be derived by taking the $k$ with the greatest probability,

$$y^\text{max}_q = \arg\max_{k \in K} [p(v^k_x = 1|q)] . \quad (3.43)$$
Chapter 4

Method

This section aims to give a comprehensive explanation of the conducted work. The data preprocessing will be described, including the setup for the network of connections. The RBM implementations are described, including how the parameter tuning was executed, the learning procedure, and what evaluation metrics that were used.

4.1 Data

Table 4.1: A partition of $D_A$, showing the board composition of organisation 20109681 and 20146865 respectively

<table>
<thead>
<tr>
<th>IND_INDEX</th>
<th>ROLE_IN_ORG</th>
<th>ORG_INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>052678570009</td>
<td>LE</td>
<td>20109681</td>
</tr>
<tr>
<td>050700970009</td>
<td>VD</td>
<td>20109681</td>
</tr>
<tr>
<td>091224690009</td>
<td>VVD</td>
<td>20109681</td>
</tr>
<tr>
<td>082200970009</td>
<td>OF</td>
<td>20109681</td>
</tr>
<tr>
<td>030210040009</td>
<td>LE</td>
<td>20109681</td>
</tr>
<tr>
<td>060249340009</td>
<td>LE</td>
<td>20109681</td>
</tr>
<tr>
<td>020103580009</td>
<td>LE</td>
<td>20146865</td>
</tr>
<tr>
<td>032050090009</td>
<td>LE</td>
<td>20146865</td>
</tr>
<tr>
<td>121610760009</td>
<td>VD</td>
<td>20146865</td>
</tr>
<tr>
<td>120700490009</td>
<td>OF</td>
<td>20146865</td>
</tr>
<tr>
<td>120724040009</td>
<td>LE</td>
<td>20146865</td>
</tr>
</tbody>
</table>

The raw data used in this thesis consisted of two separate data sets, $D_A$ and $D_B$. $D_A$ consisted of the individuals involved in an organisation, for instance
CEO, board members, signatories, in Sweden by 2018-04-01. A partition is seen in Table 4.1 conceptually showing the person to organisation mapping. In $\mathcal{D}_A$ each unique observation represented a certain individual’s involvement in some organisation. In total, $\mathcal{D}_A$ consisted of 2,215,517 individuals.

$\mathcal{D}_B$ held organisation specific data from various statements, for instance annual reports, for Swedish organisations up-to-date at 2018 Q1. The data set consisted of 1,060,810 unique organisations and their data respectively, each observation represented a particular organisation.

Before processing the data and creating the network of connections, the intersect, $\mathcal{D}_A \cap \mathcal{D}_B$, between $\mathcal{D}_A$ and $\mathcal{D}_B$ was extracted. $\mathcal{D}_A \cap \mathcal{D}_B$ consisted of 771,914 unique observations, each represented an organisation and its data. No patterns were identified regarding why some organisations were only represented in one of the data sets, becoming removed by the merger. By merging the data sets, a new feature $\text{INDIVIDUALS}$ was created as an array of individuals involved in the organisations. As an example, for organisation 20146865 from Table 4.1 the feature takes the form

$$
\begin{bmatrix}
052678570009 \\
050700970009 \\
091224690009 \\
082200970009 \\
030210040009 \\
060249340009
\end{bmatrix}.
$$

(4.1)

### 4.1.1 Data Preprocessing

Network Model

Table 4.2: Fictive data set consistent of 9 individuals with aggregated involvement in 4 organisations

<table>
<thead>
<tr>
<th>ORG_INDEX</th>
<th>INDIVIDUALS</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>[1,2,3]</td>
</tr>
<tr>
<td>B</td>
<td>[3,4,5,6,7]</td>
</tr>
<tr>
<td>C</td>
<td>[7,8,9]</td>
</tr>
<tr>
<td>D</td>
<td>[6,8]</td>
</tr>
</tbody>
</table>

For the purpose of explaining how the network of connections was derived, we consider a fictive data set, see Table 4.2. Let it consist of 4 organisations and their composition in terms of individuals. The idea is to let each company
be the link between individuals, thus spanning a network of relations, see Figure 4.1.

![Conceptual model of the network for the fictive data set presented in Table 4.2](image)

From Figure 4.1 we can produce a data set designed for feeding an RBM. An individual-company data set can be constructed by letting each unique row hold three values, an individual index, a company index and the shortest path to one another. For the fictive data set the resulting rows for \( u_1 \) from Figure 4.1 is presented in Table 4.3.

**Table 4.3: The network of connections for individual \( u_1 \)**

<table>
<thead>
<tr>
<th>IND_INDEX</th>
<th>ORG_INDEX</th>
<th>RELATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>B</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>C</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>D</td>
<td>3</td>
</tr>
</tbody>
</table>

The scheme of operations described regarding the fictive data set was applied to \( D_A \), which generated the corresponding data set of first-hand, second-hand and third-hand connections for each row in \( D_A \), similar to Table 4.3. In particular, \( D_A \) and all 2 215 517 users were utilised to create a network of connections data set, \( D_C \), equivalent to 443 310 first-hand, 1 302 720 second-hand and 7 040 308 third-hand connections.
Cleaning
Following the scope of this thesis, $\mathcal{D}_{A \cap B}$ was cleaned from improper organisations. Firstly, organisations that were not defined as limit companies are removed, removing non-profit organisations, foundations and associations. Thus, we only included organisations that possibly appear in VCs’ deal flow.

Secondly, the data set was reduced by year of foundation, to only include companies founded the past 5 years (2014-2018). By doing so, we limited our data set to only include new companies that are arguably in stage where VCs normally would be interested to invest.

Thirdly, the data set was reduced to only include active companies. At this stage, we excluded companies reported as inactive, thus not relevant to include in the deal flow of a VC.

Fourthly, the data set was reduced to only include companies with one or more employees. Hence, we excluded companies that unlikely are of interest for a VC, for example holding companies, shell companies. Having cleaned the data, a data set $\mathcal{D}'_{A \cap B}$ was extracted, which consisted of 46 517 individuals and 20 137 companies.

Lastly, $\mathcal{D}_C$ was reduced to only include companies and individuals represented in $\mathcal{D}'_{A \cap B}$, giving $\mathcal{D}'_C$, which held 238 568 connections. Retrieving $\mathcal{D}_C$ before excluding by $\mathcal{D}'_{A \cap B}$ was important to include relations that otherwise would have been rejected due to insufficient knowledge of these connections. Indeed, rejecting a part of the network in that manner would give a less truthful representation of the actual network of connections.

Conditional Information: Place of Origin
Proceeding from $\mathcal{D}'_{A \cap B}$, we extracted companies’ place of origin as conditional information. Schematically, every company was grouped by place of origin, translated to individuals being active in the purposed place of origin. Each unique individual was found to have a set of locations in which they were active. From this information we created the conditional vector, $\mathbf{r}$ for each individual, where each element in $\mathbf{r}$ represented whether the individual was active in a certain company’s place of origin.

4.1.2 Training & Test Set
In preparation of the evaluation of the models, $\mathcal{D}'_C$ was divided into one training set and one test set. Roughly 10% of $\mathcal{D}'_C$ was selected at random.
as test set, restricted to only include individuals with 2 or more connections. The reason for only including individuals with multiple connections was to enable us to withhold one connection and use our recommendation model to predict it. That is, hiding one connection for each observation and use the remainder of the connections to predict the hidden one. A detailed declaration of the sets is presented by Table 4.4.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>First-hand conn.</th>
<th>Second-hand conn.</th>
<th>Third-hand conn.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{train}$</td>
<td>51 824</td>
<td>33 746</td>
<td>133 000</td>
</tr>
<tr>
<td>$D_{test}$</td>
<td>1 850</td>
<td>3 449</td>
<td>14 646</td>
</tr>
</tbody>
</table>

### 4.2 Models & Learning

The models were trained with similar schematic set up as described in Chapter 3. An investigation of parameter tuning was executed for each of the models. The process for making predictions was performed equivalently between the models as well as the calculation of the evaluation metrics. As described in Chapter 3.1, the major difference between the models is the utilisation of additional information, making the machine conditional.

#### 4.2.1 Missing Values

An important detail regarding the models trained and tested is the handling of missing values due to the sparsity of the data set. Representing the data
set as a matrix, it follows that each column represents a visible node, equivalent to the companies in the data set. The rows in the matrix is given by the individuals in the data set, giving that the matrix has the dimensions companies times individuals.

As described in Chapter 4.1, the data set consisted of 20,137 companies and 46,517 individuals, yielding that the matrix had $20,137 \cdot 46,517 \approx 900 \cdot 10^6$ elements. However, by Chapter 4.1 it follows that our data set only held 238,568 individual-company connections, yielding a completion ratio of $238,568/900 \cdot 10^6 \approx 0.03\%$.

In order to deal with the missing values problem we used a technique presented by the authors of [21]. Instead of including all $M$ visible nodes in the learning, each observation only updated weights and biases related to its specific individual-company connections, denoted as $\tilde{M}$. That is, when learning the model, weights connected to visible nodes outside of the observations actual connections, were never considered, see Figure 4.2. This is an important step as we did not want to update weights for companies that are not connected to the considered individual.

When testing the model, by making predictions regarding hidden connections, this procedure was slightly changed. Instead of only receiving predictions for already rated companies, $\tilde{M}$, the model included all trained weights and biases rendering values for all $M$ visible nodes. Further description of the predictions and recommendations is described in Chapter 4.3.

### 4.2.2 Learning

Prior to initiating the learning algorithm we set the number of hidden units $J$, the learning rate $\epsilon$, and the number of contrastive divergence steps, $t$. The former parameter, $J$, was fixed throughout the training of every model. The latter two parameters were occasionally updated across epochs, in accordance with Chapter 4.2.5.

The implementation of the learning algorithm followed the mathematical description provided in Chapter 3.4. However, as described in Chapter 4.2.1, we only updated the weights and biases related to the considered individual’s connected companies. That is, we created minor RBMs in parallel with shared weights and hidden nodes. The RBMs were collected at the end of each epoch and the parameters of the main RBM were updated.
4.2.3 Model 1: RBM

Proceeding from the training data set, $\mathcal{D}_{\text{train}}$, the implementation of the RBM was straightforward from Chapter 3.2, but some declaration is necessary. The implementation used multinomial visible units, with $K = 3$, and Bernoulli hidden units. Moreover, the visible layer consisted of the $M = 20\,137$ unique companies found in $\mathcal{D}_C$. Comparatively, the hidden layer was set to $J$ hidden units, whereas various $J$ were tested. Each observation in $\mathcal{D}_{\text{train}}$ was an individual and the corresponding connections, whereas missing values was be handled as described in Chapter 4.2.1.

From the structure of the hidden and visible layers, it follows that the energy function in (3.12) took the form

$$E(V, h) = -\sum_{i=1}^{M} \sum_{k=1}^{3} v_i^k a_i^k - \sum_{j=1}^{J} h_j b_j - \sum_{i=1}^{M} \sum_{j=1}^{J} \sum_{k=1}^{3} W_{ij}^k h_j v_i^k.$$  

(4.2)

As described in Chapter 4.2.2, we considered each observation as a single RBM and merged the updates in the end of each epoch. Moreover, from (4.2) it follows that (3.13) and (3.14) is given by

$$p(h_j = 1|V) = \sigma(b_j + \sum_{i=1}^{M} \sum_{k=1}^{3} v_i^k W_{ij}^k),$$

$$p(v_i^k = 1|h) = \frac{\exp (b_i^k + \sum_{j=1}^{J} h_j W_{ij}^k)}{\sum_{l=1}^{3} \exp (b_i^l + \sum_{j=1}^{J} h_j W_{ij}^l)}.$$  

After choosing the number of steps, $t$, for the contrastive divergence we trained the RBM using $\mathcal{D}_{\text{train}}$ as input in Algorithm 1 for some given number of epochs. When the learning was finished we acquired the weights $W$ and biases $a$ and $b$, respectively.

4.2.4 Model 2: CRBM (Place of Origin)

Proceeding from the training data sets, $\mathcal{D}_{\text{train}}$, the implementation of the CRBM was straightforward from Chapter 3.2, but some declaration is necessary. The implementation used multinomial visible units, with $K = 3$, and Bernoulli hidden units. Moreover, the visible layer consisted of the $M = 20\,137$ unique companies found in $\mathcal{D}_C$. Comparatively, the hidden layer was set to $J$ hidden units, whereas various $J$ were tested. Each observation in $\mathcal{D}_{\text{train}}$ was an individual and the corresponding connections, whereas missing values was handled as described in Chapter 4.2.1. In addition we added
conditional information as \( r \) where \( r_i \in \{0, 1\} \) for \( i = 1, \ldots, M \). \( r_i = 1 \) was equivalent to the individual being active in the location where company \( i \) originates from, \( r_i = 0 \) being the contrary.

From the structure of the hidden and visible layers, it follows that the energy function in \((3.12)\) took the form

\[
E(V, h) = - \sum_{i=1}^{M} \sum_{k=1}^{3} v_i^k a_i^k - \sum_{j=1}^{J} h_j b_j - \sum_{i=1}^{M} \sum_{j=1}^{J} \sum_{k=1}^{3} W_{ij}^k h_j v_i^k - \sum_{i=1}^{M} \sum_{j=1}^{J} h_j D_{ij} r_i.
\]

\[(4.3)\]

As described in Chapter 4.2.2, we considered each observation as a single CRBM and merged the updates at the end of each epoch. Moreover, from \((4.3)\) it follows that \((3.13)\) and \((3.14)\) is given by

\[
p(h_j = 1 | V, r) = \sigma(b_j + \sum_{i=1}^{M} \sum_{k=1}^{3} v_i^k W_{ij}^k + \sum_{i=1}^{M} r_i D_{ij}),
\]

\[
p(v_i^k = 1 | h) = \frac{\exp (b_i^k + \sum_{j=1}^{J} h_j W_{ij}^k)}{\sum_{l=1}^{3} \exp (b_l^k + \sum_{j=1}^{J} h_j W_{ij}^l)}.
\]

After choosing the number of steps, \( t \), for the contrastive divergence the CRBM was trained using \( D_{\text{train}} \) as input to Algorithm 2 for some given number of epochs. When the learning was finished we acquired the weights \( W \) and \( D \), and biases \( a \) and \( b \), respectively.

### 4.2.5 Parameter Grid

Due to the lack of previous research on data sets similar to the network of connection data set used in this thesis, a parameter exploration was executed. In addition, the approach of reducing the learning rate as well as increasing the steps of contrastive divergence pragmatically as the learning proceed was investigated.

**Hidden units**

The number of hidden units was evaluated for the standard RBM as well as the CRBM. Particularly, the following set was tested:

\[
J = \{30, 70, 120, 150, 210\}
\]

\[(4.4)\]
Learning rate
The tuning included evaluating how varying the learning rate affected the learning, initially the following learning rates were tested

$$\epsilon \in \{0.005, 0.01, 0.025, 0.05, 0.1, 0.2\}, \quad (4.5)$$

followed by reducing the learning rate over epochs. The latter executed by letting the learning rate decay by half every fifth epoch. \(\epsilon_0\) was defined as the initial learning rate and updated for epoch \(n\) by

$$\epsilon_n = \begin{cases} 
\epsilon_0 \cdot 2^0, & 1 \leq n \leq 5, \ n \in \mathbb{N} \\
\epsilon_0 \cdot 2^{-1}, & 5 < n \leq 10, \ n \in \mathbb{N} \\
\epsilon_0 \cdot 2^{-2}, & 10 < n \leq 15, \ n \in \mathbb{N} \\
\epsilon_0 \cdot 2^{-3}, & 15 < n, \ n \in \mathbb{N} 
\end{cases}, \quad (4.6)$$

Different initial learning rates, \(\epsilon_0\), were investigated, particularly the set provided by (4.5) was tested.

Steps of contrastive divergence
Similarly to evaluating the learning rate, the steps of contrastive divergence used was first evaluated by fixed values and also evaluated when increased progressively. Moreover, the following values for steps were tested

$$\mathcal{T} = \{1, 3, 5\}. \quad (4.7)$$

When evaluating how gradually increasing the steps, \(t\), of contrastive divergence, the following updating rule was applied for epoch \(n\):

$$t_n = \begin{cases} 
1, & 1 \leq n < 5, \ n \in \mathbb{N} \\
5, & 5 \leq n, \ n \in \mathbb{N} 
\end{cases}. \quad (4.8)$$

4.3 Making Predictions
The models’ predictions was executed throughout the learning, using the test set to evaluate how the performance was affected over the epochs. Due to the increased computational time caused by making predictions, the predictions were limited to only be executed every fifth epoch on 50\% of the test set. After the last epoch the entire test set were used to evaluate the prediction ability, yielding prediction matrices. As mentioned earlier, the predictions had no impact on the actual learning since they were performed without updating the weights.
4.4 Evaluation

The evaluation of the methods was done schematically and across epochs. Three primary metrics were used to compare the RBM versus the CRBM, evaluating how adding conditional information potentially boosted the model. By taking metrics across epochs it was also possible to evaluate the learning aspect of the models.

Furthermore, it was desired to measure how well the models are able to find the best recommendations. However, lacking universal best practise, it was problematic to find righteous metrics for recommendation models purpose. Consequently explanatory metrics were considered, deriving the models’ ability to recreate hidden connections accurately. In doing so, it was also possible to evaluate which connections the model typically failed to predict.

4.4.1 Reconstruction & Prediction Error

For the purpose of this thesis, the reconstruction error was used to monitor learning aspects and evaluate predictions since the metric is aligned with the objective of the project. The reconstruction error was calculated as the root mean square difference between the sampled visible layer and the actual observation of the visible layer. For a set of observations \( S = \{ V_1, \ldots, V_N \} \), each observation \( V_i \), consisted of various number of connections \( V_i = (v_i^1, \ldots, v_i^{\tilde{M}_i}) \) where \( \tilde{M}_i \) denotes the length of observation \( V_i \). The reconstruction error was calculated by

\[
\text{REC. ERR.} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \frac{1}{\tilde{M}_i} \sum_{n=1}^{\tilde{M}_i} (\mathbb{E}[v_n^i] - \mathbb{E}[(v_n^i)^{(t)}])^2}
\]

where \( t \) denotes number of contrastive divergence steps for reconstruction. \( \mathbb{E}[\cdots] \) was calculated using

\[
\mathbb{E}[v] = \sum_{k=1}^{K} R_k v_k
\]

where \( v \) is a \( k \)-dimensional vector and \( R_k \) is the type of connection. After each epoch the reconstruction error was calculated using the training set, \( D_{\text{train}} \), and was used to measure how fast the model was learning and possibly converging.

For the prediction error, we denote the actual connections as \( \hat{y} \) and denote
the predicted connections, found using (3.42), as y. The prediction error 
is calculated as the root mean square error between the actual and the predicted connection,

\[
PRED. ERR. = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y - \hat{y})^2}.
\] (4.11)

4.4.2 Prediction Accuracy

Given the actual connections, \(\hat{y}\), and the predicted connections, y, we determined the prediction accuracy by calculating the number of predictions being correct. In particular

\[
PRED. ACC. = \frac{1}{N} \sum_{i=1}^{N} 1\{y_i = \hat{y}_i\},
\] (4.12)

where N is the number of withheld connections.

As an extension of the accuracy, the type of errors was further investigated by measuring the frequency of the predicted versus the actual connections. Let M be our prediction matrix, having dimensions 3 × 3, given by the number of considered type of connections. The diagonal is equivalent to the predictions being accurate and the upper respective lower triangles are erroneous predictions. Each predicted connection \(k\), was tested versus each possible connection \(\hat{k}\) such that

\[
M_{k,\hat{k}} = \begin{cases} 
\sum_{i=1}^{N} 1\{y_i = k\}, & k = \hat{k}, \\
\sum_{i=1}^{N} 1\{y_i = \hat{k}\}, & k \neq \hat{k},
\end{cases}
\] (4.13)
y, being the actual connection for each observation in the considered set.

4.4.3 Relative Weight Change

For each epoch the change of the weights, \(W\) was determined. Given the update, \(\Delta W\), the relative mean weight change was determined by

\[
\frac{\Delta W}{W} = \frac{1}{JKM} \sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{i=1}^{M} \frac{\Delta W_{ij}}{W_{ij}}.
\] (4.14)

The weight change indicates convergence of the model, as well as which learning rates that induces satisfactory learning. Ideally the weight change is stable without major peaks.
Chapter 5

Result

This section aims to highlight the characteristics discovered during the conducted research. The structure will follow the previous parts of the report, starting from the RBM model and moving forth to the CRBM model. Inferences drawn with respect to the results will be discussed in Chapter 6.

5.1 RBM

(a) Effects of changing the learning rate with respect to the reconstruction error
(b) Effects of changing the learning rate with respect to the relative weight change

Figure 5.1: How adjusting the learning rate affect an RBM, with respect to the reconstruction error and the relative weight change for an RBM with 120 hidden units using 1 contrastive divergence step

In Figure 5.1a it is observable that a low learning rate is able to render a lower reconstruction error at the cost of needing a higher number of epochs. Similarly, a higher learning rate causes the RBM model to make a larger
initial jump which flattens out and seem to get very close to its final reconstruction error after only a few number of epochs. It is noteworthy that a lower learning rate has a reconstruction error strictly decreasing across every epoch, while a higher learning causes more ambiguous decrements. The effect of the shape of the decrements is that the lowest learning rate resulted in a lower final reconstruction error than both of the other learning rates considered.

Observing Figure 5.1b it follows that the weight change finds some value it oscillates around. Increasing the learning rate significantly lowers that value, causing smaller relative changes.

![Figure 5.2](image)

Figure 5.2: How a decaying learning rule, dividing by half every fifth epoch affect an RBM with 70 respectively 120 hidden units, with respect to the reconstruction error. Initial learning rate 0.05 and 1 contrastive divergence step, $\epsilon^-$ denotes decaying learning rate.

Figure 5.2 shows the effects of letting the learning decay over epochs. It can be observed that by using a decaying learning rate, different reconstruction errors are found. Although, the decayed learning rate is only slowing down the reduction of the error. The effects are substantial for an RBM with 70 as well as 120 hidden units. Note that the dip seen in 5.1a is reduced by lowering the number of hidden units, having the same learning rate. It follows by the decay rule that the learning rate is unchanged the first five epochs and henceforth reduced, causing the overlapping behaviour over the initial epochs.
Figure 5.3: How adjusting the number of hidden units affects the reconstruction error respectively prediction error for an RBM with learning rate 0.05 and 1 contrastive divergence step.

In Figure 5.3a we notice that increasing the number of hidden units can render a lower reconstruction error. Higher number of hidden units is observed to also yield steeper learning. A dip is observed for every setting of hidden units except for $J = 30$.

Figure 5.3 shows that increasing the number of hidden units has positive effect on the prediction error until some threshold, identified by $J = 150$ having worse error than $J = 120$. Note that the prediction error increases significantly for $J = 150$ between the fifth and the final epoch compared to the corresponding for $J = 30$, $J = 70$ and $J = 120$.

Figure 5.4: Effects from increasing the number of contrastive divergence steps with respect to the reconstruction error for an RBM with 30 hidden units and learning rate 0.01.

From Figure 5.4, it follows that increasing the number of contrastive divergence-
steps increases how quickly the model learns during the initial epochs. In addition, it is observable that the number steps slightly increases the final reconstruction error.

## 5.2 CRBM (Place of Origin)

![Graph](image)

**Figure 5.5:** The impact of adding place of origin as conditional vector \( r \) with respect to the reconstruction error, using 120 hidden units, learning rate 0.01 and 1 contrastive divergence step

From Figure 5.5, it follows that adding conditional information regarding the place of origin lowers the reconstruction error. The learning in sense of reconstruction error is quicker and converges at an earlier epoch for the CRBM, compared to the RBM.

![Graph](image)

(a) With respect to the prediction error  
(b) With respect to the relative weight change

**Figure 5.6:** Learning aspects of adding place of origin as conditional vector \( r \) to a CRBM compared to an RBM, using 120 hidden units, learning rate 0.01 and 1 contrastive divergence step

35
By observing Figure 5.6a we acknowledge that the CRBM is trained to achieve a lower prediction error than the RBM with a breaking point between epoch 10 and 15. In addition, the CRBM has a strictly decreasing prediction error during the first 15 epochs, unlike the RBM where the prediction error increases between epoch number 10 and 15.

Figure 5.6b show a tendency of oscillating relative weight change for the CRBM and the RBM respectively, whereas the RBM oscillates around a slightly lower value. In addition, the RBM changes its weights significantly the initiating epochs, compared to the CRBM.

<table>
<thead>
<tr>
<th>Predicted</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>106</td>
<td>36</td>
<td>52</td>
<td>1</td>
<td>146</td>
<td>51</td>
</tr>
<tr>
<td>(11%)</td>
<td>(4%)</td>
<td>(5%)</td>
<td>(0%)</td>
<td>(15%)</td>
<td>(5%)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>66</td>
<td>70</td>
<td>0</td>
<td>131</td>
<td>43</td>
</tr>
<tr>
<td>(3%)</td>
<td>(7%)</td>
<td>(7%)</td>
<td>(0%)</td>
<td>(14%)</td>
<td>(5%)</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>54</td>
<td>108</td>
<td>425</td>
<td>6</td>
<td>208</td>
<td>364</td>
</tr>
<tr>
<td>(6%)</td>
<td>(11%)</td>
<td>(45%)</td>
<td>(1%)</td>
<td>(22%)</td>
<td>(38%)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.7: Predictions as expected value from a CRBM using 120 hidden units, learning rate 0.01 and 1 contrastive divergence step on the test set. The aggregated accuracy is 63%, where percentages are fractions with respect to the total number of observations.

From Figure 5.7 we recognise that the CRBM is diversified in its predictions. It follows that by using place of origin as conditional variable, the model is able to identify every type of connection. Reading the values from Figure 5.8 it follows that the RBM rarely identifies first-hand relations. The RBM model predicts a first-hand relation 7 times in total, but only accurately once, despite the fact that the total number of first-hand relations was 198. The RBM seem inclined to make second-hand predictions, doing so 56% of the time. Similar findings could be observed when using the maximum probability for retrieving predictions, the RBM managed to get a slightly higher overall accuracy of 54% with that predictions method.
Chapter 6

Discussion

This section aims to discuss the selected findings from Chapter 5 and address parts of the conducted research that could be improved. The purpose of this section is to elaborate our thoughts regarding which findings are of particular interest and make inferences that relate to the objective of this thesis.

6.1 RBM & CRBM

Comparing the observations from Figure 5.1a and Figure 5.1b, we can draw some inference regarding what an efficient learning rate would be. Indeed $\epsilon = 0.05$ was the quickest at finding its final reconnection error, yet by observing Figure 5.1b, we identify that the weights barely changed after the fourth epoch. We suspect this is due to a local maximum in the gradient ascent. This is implied by the observations using learning rate $\epsilon = 0.01$, achieving a lower reconstruction error, thus indicating that the model can learn from adding additional information to the hidden layer. If this is the case, it could explain why we see an initial dip in the reconstruction error for some models, as the model would take a too large step, overshoot, and find a new gradient inconsistent with the previous. Needless to say, the other models in Figure 5.1a seem to converge in terms of reconstruction error after 20 epochs. It could be of relevance to run a higher number of epochs, as the recorded relative weight changes and the reconstruction error implies that the model is not fully learned.

The behaviour of oscillating weights is troublesome to map. We identified a similar behaviour for the CRBM in Figure 5.6. We reckon this unlikely to depend on the parameter settings, as the relative weight depend on the learning rate by proportion and the number of hidden units is at worst close
to the brink of what would be overfitting, see Figure 5.3. We hypothesise that the data set is too sparse to model in its unconditioned form.

However, the learning cannot solely be evaluated by converging relative weight change. Comparing the results from the learning of RBM versus the CRBM, we note a major difference. Both models showed a behaviour of oscillating weights that did not converge. Yet, the CRBM had strictly decreasing prediction error, in contrary to the RBM. Clearly, the RBM eventually stops its process of learning—it did not improve its ability to predict the networks connections. Indeed by adding conditional information, we are able to come closer to a sufficient amount of data for predicting the networks’ connections.

As it appear, adding place of origin as conditional information improved the speed of the learning, identified by comparing the models in Figure 5.5. The model using conditional data is exposed to a greater amount of data to learn from, in each epoch. Given that the conditional data is not contradictory with respect to the training data, the model is intuitively expected to perform greater.

Comparing the prediction matrices, Figure 5.7 and Figure 5.8, for the RBM and the CRBM respectively, we find additional information strengthening the claim that the data is insufficient in its unconditional form. The RBM model only managed to get a slight perception of what a third-hand relation look like, while being contemptible in predicting other connections. The outcome, predicting considerably to the upper-hand of second-hand relations could be a mathematical coincidence, 2 being the mean of 1 and 3. Indeed this makes most of the model’s predictions unreliable, whereas the maximum probability method of retrieving predictions would be worthwhile considering.

The effects of increasing the number of contrastive divergence steps are clearly visible in Figure 5.4. It seems slightly beneficial to perform this to improve the learning. As of how this relates to our results, it would not affect the output aside from left-shifting the learning progress in terms of the reconstruction error. If the purpose was to fit the best model for the given data, the contrastive divergence steps would be of great relevance to evaluate. The con of doing so is the aspect of computation intensity. Potentially large-scale sampling across the epochs makes the model obese, which we experienced markedly. Depending on data set and other setup parameters, one could reduce the number of epochs before achieving convergence by increasing the number of contrastive divergence steps. As mentioned earlier, this inference have been stipulated by previous research, see [1, 2].
Letting the learning rate decay across epochs appear to have significant result on the learning in terms of reconstruction error. Previous research has shown this is an easy-to-implement method improving the rate of learning. We can confirm these results and add additional findings to why this works. By observing Figure 5.1a it is unambiguous that a high learning rate is quicker initially, but at the cost of potentially reducing the final reconstruction error. The intersect between $\epsilon = 0.005$ and $\epsilon = 0.01$ indicate why decaying learning rate causes decreased learning speed. Without doubt, one could optimise the learning by initially letting the model use learning rate $\epsilon = 0.01$ the initial epochs and henceforth let it decay. By doing so, the perks of each learning rate would be utilised. That being said, as discussed earlier, one need to be considerate choosing the initial value, as a too high learning rate could trap the learning where it would have trouble to learn the model additionally.

### 6.2 Method

In the data preparation the data was reduced considerably. Despite aligning the possible application of an RBM recommendation model in venture capital, this has high impact on which type of connections that the model will be exposed to. Particularly, a larger time frame would be of relevance to explore. One can speculate that an individual involved in more mature companies have a larger network, thus contributing stronger to the network.

Regarding the network of connections, we address that a too short propagation might have been used. We mention data sparsity as a troubling matter, which could have been handled by adding fourth-hand, fifth-hand or further connections. It is not certain that the models would benefit from adding such information, but it would be worth investigating.

The data partitioning into the test and training set could be improved. Truly, the test set does not reflect the training set in terms of distribution of connections. In one sense, we told our models that the reality looks the one way and tested it on something different. For practical reasons, we are able to motivate our split yet it is desirable to manipulate the split additionally to correctly reflect the training set.

In addition, we want to highlight our thoughts regarding the evaluation metrics used. The metrics used are considered the best tools available for comparing an RBM model versus a CRBM model as well as assess the learning
aspect. On the one hand, one should not forget that the models’ purpose is to make new recommendations of companies of interest rather than predicting old ones. In doing so, prediction error, reconstruction error and prediction matrices are poor tools in analysing the results. On the other hand, there is no way to evaluate whether a certain recommendation is good or bad. Fully assessing a recommendation model would likely require more rigorous methods of evaluation as well as statistical testing.

Finally, a critical step in the implementation of RBMs and CRBMs respectively as recommendation models based on a network of connections, is the handling of missing values. In this thesis we handled the problem by letting each individual have their own smaller RBM or CRBM, with shared hidden layer and weights. This makes the implementation a bit trickier, although it is crucial to have in consideration. An other approach that could be of interest to evaluate is to use a mean value for every non-connected company and use the connections as deviations from the mean.
Chapter 7

Conclusions

In our thesis we successfully implemented RBMs to a network of connections. We are able to conclude that the model is fairly sensible to the parameters used. From our findings it follows that the RBM was not particularly useful in predicting withheld connections from our data set. However, further research is necessary to identify whether our findings are spurious or systematic.

Regarding the models’ parameters, the choice of learning rate has high impact on the final model. A too high learning rate can trap the model where running additional epochs do not learn the model further, with absent weight updates as a consequence. Considering a decaying learning rate is recommended. Additionally, the number of hidden units likely need tuning, having impact on the prediction error as well as the reconstruction error. Lastly, the number of contrastive divergence steps can be set to increase after some epochs to improve the learning speed.

We found evidence implying that by adding place of origin as conditional information, the RBM can be improved. The improvements were identified in every considered evaluation metrics, whereas the ability to reconstruct withheld connections is particularly noteworthy. Although, additional research is required to determine if our findings are a recurring behaviour.
Chapter 8

Future Work

For future work we highlight three branches of improvements to further elaborate the objective considered in this thesis. Firstly, the data preparation and preprocessing can be improved. For this, two major corrections are particularly noteworthy, to expand the data set to include a wider range of years and to expand the data set by including additional connections, such as fourth-hand and fifth-hand connections.

Secondly, additional work regarding conditional information to boost the existing network of connections is feasible. Including data not mentioned in this thesis to improve the network of connections is worthwhile investigating.

Thirdly, the statistical validity of the results can be assessed. This thesis include promising results and interesting sightings that need statistical testing to possibly impact the research of RBMs as recommendation model for a network of connections.

Lastly, a topic to elaborate further outside of the objective presented in this thesis, is the way of assessing the VC perspective. This thesis used company board compositions as approximation for the network of connections. As a first step it would be interesting to explore a VCs network by their previous investments. Subsequently, one could possibly evaluate if the network of connections proposed in this thesis can be implemented as conditional information for a VC recommendation model.
Bibliography


Appendices
Appendix A

Derivation of the Conditional Probabilities

Hidden layer conditioned on the visible layer

The conditional probability of the hidden layer given the visible nodes is given by

\[
p(h|V) = \frac{p(V, h)}{p(V)} = \frac{\frac{1}{Z} \exp(-E(V, h))}{\frac{1}{Z} \sum_h \exp(-E(V, h))} = \frac{\exp(-E(V, h))}{\sum_h \exp(-E(V, h))},
\]

where \( \sum_h \) is the sum over every \( h = \{0, 1\}^J \). Inserting the energy function for the RBM yields

\[
p(h|V) = \frac{\exp(\sum_i \sum_k v_i^k a_i^k + \sum_j h_j b_j + \sum_i \sum_j \sum_k W_{ij}^k h_j v_i^k)}{\sum_h \exp(\sum_i \sum_k v_i^k a_i^k + \sum_j h_j b_j + \sum_i \sum_j \sum_k W_{ij}^k h_j v_i^k)}.
\]

For the \( j \)th element of \( h \),

\[
p(h_j = 1|V) = \frac{\exp(\sum_i \sum_k v_i^k a_i^k + \sum_j h_j b_j + \sum_i \sum_k W_{ij}^k v_i^k)}{\exp(\sum_i \sum_k v_i^k a_i^k) + \exp(\sum_i \sum_k v_i^k a_i^k + \sum_j h_j b_j + \sum_i \sum_k W_{ij}^k v_i^k)}
\]

\[
= \frac{\exp(b_j + \sum_i \sum_k W_{ij}^k v_i^k)}{1 + \exp(b_j + \sum_i \sum_k W_{ij}^k v_i^k)} = \sigma(b_j + \sum_i \sum_k W_{ij}^k v_i^k).
\]

For the conditional RBM the derivations are the analogous except one additional term consisting of \( D_{ij}v_i \) that is inherited through the derivations.
Visible layer conditioned on the hidden layer

The conditional probability of the visible layer given the hidden nodes is given by

\[
p(V \mid h) = \frac{p(V, h)}{p(h)} = \frac{\frac{1}{Z} \exp(-E(V, h))}{\frac{1}{Z} \sum_V \exp(-E(V, h))} = \frac{\exp(-E(V, h))}{\sum_V \exp(-E(V, h))},
\]

where the sum over \(V\) denotes the sum over every multinomial state \(k\) for each \(v_i\) in \(v = (v_1, \ldots, v_i)\). Inserting the energy function for the RBM yields

\[
p(V \mid h) = \frac{\exp(\sum_i \sum_k v_i^k a_i^k + \sum_j h_j b_j + \sum_i \sum_j \sum_k W_{ij}^k h_j v_i^k)}{\sum_V \exp(\sum_i \sum_k v_i^k a_i^k + \sum_j h_j b_j + \sum_i \sum_j \sum_k W_{ij}^k h_j v_i^k)} = \frac{\prod_i \prod_k \exp(a_i^k + \sum_j W_{ij}^k h_j)v_i^k}{\sum_V \prod_i \prod_k \exp(a_i^k + \sum_j W_{ij}^k h_j)v_i^k},
\]

Considering only the \(i\)th vector \(v\) where \(v_i^k = 1\), (A.11) is reduced to

\[
p(v_i^k = 1 \mid h) = \frac{\exp(a_i^k + \sum_j W_{ij}^k h_j)}{\sum_k \exp(a_i^k + \sum_j W_{ij}^k h_j)},
\]

where the denominator is the sum over every \(k\) for \(v_i\), normalising the probability. (A.12) is known as the softmax function.
Appendix B

Derivations of the Log-likelihood Gradient

The general log-likelihood gradient is stated,

$$\frac{\partial \ln p(V|\theta)}{\partial \theta} = -\sum_h p(h|V) \frac{\partial}{\partial \theta} E(V, h) + \sum_{V,h} p(V, h) \frac{\partial}{\partial \theta} E(V, h) \quad \text{(B.1)}$$

For an RBM \( \text{(B.1)} \) is evaluated by inserting \( \theta = w_{ij} \) and the derivative

$$\frac{\partial E(V, h)}{\partial w_{ij}^k} = -h_j v_i^k, \quad \text{(B.2)}$$

yielding

$$\frac{\partial \ln p(V|w_{ij}^k)}{\partial w_{ij}^k} = \sum_h p(h|V) h_j v_i^k - \sum_{V,h} p(V, h) h_j v_i^k. \quad \text{(B.3)}$$

By using

$$p(V, h) = p(h|V)p(V) \quad \text{(B.4)}$$

\( \text{(B.3)} \) is rewritten as

$$\frac{\partial \ln p(V|w_{ij}^k)}{\partial w_{ij}^k} = \sum_h p(h|V) h_j v_i^k - \sum_{V,h} p(V) \sum_h p(h|V) h_j v_i^k. \quad \text{(B.5)}$$

From the RBM structure it follows that

$$p(h|V) = \prod_j p(h_j|V) \quad \text{(B.6)}$$
and for any set, \( \ell \), of hidden bernoulli nodes that
\[
\sum_{h_{\ell}} p(h_{\ell} | V) = 1 \quad (B.7)
\]
giving
\[
\sum_{h} p(h | V)h_jv_i^k = \sum_{h} \prod_{j} p(h_j | V)h_jv_i^k = \sum_{h_{j-}} p(h_{j-} | V) \sum_{h_j} p(h_j | V)h_jv_i^k = \sum_{h_j} p(h_j | V)h_jv_i^k = p(h_j = 1 | V)v_i^k, \quad (B.8)
\]
where \( h_{j-} \) denotes every hidden node except \( h_j \). Thus, by using (B.8) in (B.5),
\[
\frac{\partial \ln p(V | w_{ij}^k)}{\partial w_{ij}^k} = p(h_j = 1 | V)v_i^k - \sum_V p(V)p(h_j = 1 | V)v_i^k \quad (B.9)
\]
Analogously, for \( \theta = a_i^k \) and the derivative
\[
\frac{\partial E(V, h)}{\partial a_i^k} = -v_i^k \quad (B.10)
\]
(B.1) gives
\[
\frac{\partial \ln p(V | a_i^k)}{\partial a_i^k} = \sum_{h} p(h | V)v_i^k - \sum_{V, h} p(V, h)v_i^k = \sum_{h} p(h | V)v_i^k - \sum_{V} p(V) \sum_{h} p(h | V)v_i^k = v_i^k - \sum_{V} p(V)v_i^k, \quad (B.11)
\]
For \( \theta = b_j \) and the derivative
\[
\frac{\partial E(V, h)}{\partial b_j} = -h_j, \quad (B.12)
\]
(B.1) gives
\[
\frac{\partial \ln p(V | b_j)}{\partial b_j} = \sum_{h} p(h | V)h_j - \sum_{V, h} p(V, h)h_j = \sum_{h} p(h | V)h_j - \sum_{V} p(V) \sum_{h} p(h | V)h_j. \quad (B.13)
\]
By using same technique as in (B.8) the expression remaining is

\[
\frac{\partial \ln p(V|b_j)}{\partial b_j} = p(h_j = 1|V) - \sum_V p(V)p(h_j = 1|V). \tag{B.14}
\]

For the CRBM, using \( \theta = D_{ij} \) and the derivative

\[
\frac{\partial E(V, h)}{\partial D_{ij}} = -h_j r_i, \tag{B.15}
\]

in (B.1) yields

\[
\frac{\partial \ln p(V|D_{ij})}{\partial D_{ij}} = \sum_h p(h|V)h_j r_i - \sum_{V,h} p(V, h)h_j r_i \\
= \sum_h p(h|V)h_j r_i - \sum_V p(V)\sum_h p(h|V)h_j r_i \tag{B.16}
\]

\[
= p(h_j = 1|V)r_i - \sum_V p(V)p(h_j = 1|V)r_i. \tag{B.17}
\]