Co-occurrence Rate Networks

Towards separate training for undirected graphical models

Zhemín Zhú

Co-occurrence Rate Networks

Towards separate training for undirected graphical models

Zhemín Zhu

PhD dissertation committee:

Chairman and Secretary:

Prof. dr. Peter M. G. Apers

University of Twente, NL

Supervisor:

Prof. dr. Peter M. G. Apers

University of Twente, NL

Co-supervisor:

Dr. ir. Djoerd Hiemstra

University of Twente, NL

Members:

Dr. Ingo Frommholz

University of Bedfordshire, UK

Prof. dr. Tom Heskes

Radboud University Nijmegen, NL

Prof. dr. Dirk K. J. Heylen

University of Twente, NL

Prof. dr. ir. Raymond N. J. Veldhuis

University of Twente, NL

Prof. dr. Arjen P. de Vries

Delft University of Technology /

Centrum Wiskunde & Informatica, NL

CTIT Ph.D. Thesis Series No. 15-372

Centre for Telematics and Information Technology

University of Twente

P.O. Box 217, 7500 AE

Enschede, The Netherlands.

SIKS Dissertation Series No. 2015-22

The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

COMMIT/

This research has been supported by the Dutch national program COMMIT/.

ISBN: 978-90-365-3932-6

ISSN: 1381-3617 (CTIT Ph.D. Thesis Series No. 15-372)

DOI: 10.3990/1.9789036539326

Copyright © 2015, Zhemin Zhu, Enschede, The Netherlands.

CO-OCCURRENCE RATE NETWORKS

TOWARDS SEPARATE TRAINING FOR UNDIRECTED GRAPHICAL MODELS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
Prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties,
in het openbaar te verdedigen
op vrijdag 16 oktober 2015 om 12.45 uur

door

Zhemin Zhu

geboren op 7 november, 1981
te Zhejiang, China

Dit proefschrift is goedgekeurd door:
Prof. dr. Peter M. G. Apers (promotor)
Dr. ir. Djoerd Hiemstra (assistent-promotor)

CO-OCCURRENCE RATE NETWORKS

TOWARDS SEPARATE TRAINING FOR UNDIRECTED GRAPHICAL MODELS

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus,
Prof. dr. H. Brinksma,
on account of the decision of the graduation committee,
to be publicly defended
on Friday, October 16, 2015 at 12:45

by

Zhemin Zhu

born on November 07, 1981
in Zhejiang, China

This dissertation is approved by:
Prof. dr. Peter M. G. Apers (supervisor)
Dr. ir. Djoerd Hiemstra (assistant supervisor)

Acknowledgments

First and foremost, I express my deepest gratitude to my supervisor Prof. dr. Peter Apers and daily supervisor Dr. ir. Djoerd Hiemstra. I give special thanks to Djoerd for his guidance, expertise, encouragements, patience and continuous support during the four-year PhD life. Peter is a very nice director who encouraged me to go further in research based on my interests.

It is fortunate to have Dr. Ingo Frommholz, Prof. dr. Tom Heskes, Prof. dr. Dirk K. J. Heylen, Prof. dr. ir. Raymond Veldhuis and Prof. dr. Arjen de Vries in my graduation committee. I give thanks to all of them for reviewing my thesis during the 2015 summer holidays.

I give special thanks to Prof. dr. Tom Heskes for his expertise, guidance and inspiring discussions. I give big thanks to Prof. dr. Arjen de Vries for discussions with him and his group members at CWI in Amsterdam. I give thanks to Prof. dr. ir. Raymond Veldhuis for offering me the opportunity to explain my work face to face.

My excellent colleagues in the database group deserve my thanks for creating a very nice working environment. My thanks go to Maarten Fokkinga for teaching me tips of Latex, some mathematical knowledge related to category theory and MapReduce; Maurice van Keulen for helping me fill out progress reports of the COMMIT/ project each time; Robin Aly for discussing some interesting research topics; Suse Engbers and Ida den Hamer for saving me from doing complex paper works; Jan Flokstra for solving tough engineering problems; Andreas Wombacher for bringing me to Enschede; Juan Amiguet for teaching me tips of programming, and a lot of random stuff each time I came to his office; Mena Habib for co-operation on interesting topics and inspiring discussions; Brend Wanders for inspiring discussions and helping me build a web-based interface for annotation (See Figure 7.8 in this thesis). I give special thanks to Brend for helping me with the Dutch translation of the abstract. In the Dutch translation, some terminologies are kept in English because the Dutch translations seem weird for these terminologies. Victor de Graaff for bringing laughs to the group with jokes; Lei Wang for talking with

me in Chinese; Mohammad Khelghati for locking the door when I forgot my key; Iwe Muiser for sharing his happy events, such as sailing or music, with us; Sergio Duarte for inspiring discussions about his research; Ghita Berrada for teaching me Arabic vocabularies and pronunciations; Rezwan Huq for collecting us together for lunch; Kien Tjin-Kam-Jet for enjoying lunch together; Almer Tigelaar for his advice on how to do PhD research smoothly.

Many thanks go to Dong Nguyen and Dolf Trieschnigg at Human Media Interaction (HMI) group for organizing reading group meetings. I also give thanks to Dong who provided proofreading for some parts of this thesis.

I would like to say “Dank u wel!” to all Dutch tax payers. Without their financial support, I could not start and finish the PhD programme. The research presented in this thesis has been supported by the Dutch national program COMMIT/ and conducted within the Centre for Telematics and Information Technology (CTIT), EEMCS faculty at University of Twente.

I give thanks to all my friends around. Last but not least, my thanks go to my family members for their unconditional love and support.

Zhemin Zhu
Enschede, NL
September 2015

Abstract

Dependence is a universal phenomenon which can be observed everywhere. In machine learning, probabilistic graphical models (PGMs) represent dependence relations with graphs. PGMs find wide applications in natural language processing (NLP), speech processing, computer vision, biomedicine, information retrieval, etc. Many traditional models, such as hidden Markov models (HMMs), Kalman filters, can be put under the umbrella of PGMs. The central idea of PGMs is to decompose (factorize) a joint probability into a product of local factors. Learning, inference and storage can be conducted efficiently over the factorization representation.

Two major types of PGMs can be distinguished: (i) Bayesian networks (directed graphs), and (ii) Markov networks (undirected graphs). Bayesian networks represent *directed dependence* with *directed edges*. Local factors of Bayesian networks are *conditional probabilities*. Directed dependence, directed edges and conditional probabilities are all *asymmetric* notions. In contrast, Markov networks represent *mutual dependence* with *undirected edges*. Both of mutual dependence and undirected edges are *symmetric* notions. For general Markov networks, based on the Hammersley–Clifford theorem, local factors are *positive functions* over maximum cliques. These local factors are explained using intuitive notions like ‘compatibility’ or ‘affinity’. Specially, if a graph forms a clique tree, the joint probability can be reparameterized into a junction tree factorization.

In this thesis, we propose a novel framework motivated by the Minimum Shared Information Principle (MSIP):

We try to find a factorization in which the information shared between factors is minimum. In other words, we try to make factors as independent as possible.

The benefit by doing this is that we can train factors separately without paying a lot of efforts to guarantee consistency between them. To achieve this goal, we develop a theoretical framework called co-occurrence rate networks (CRNs) to obtain such a factorization. Briefly, given a joint probability, the CRN factorization is obtained as follows. We first strip off singleton probabilities from

the joint probability. The quantity left is called co-occurrence rate (CR). CR is a symmetric quantity which measures mutual dependence among variables involved. Then we further decompose the joint CR into smaller and independent CRs. Finally, we obtain a CRN factorization whose factors consist of all singleton probabilities and CR factors. There exist two kinds of independencies between these factors: (i) a singleton probability is independent (Here independent means two factors do not share information.) of other singleton probabilities; (ii) a CR factor is independent of other CR factors conditioned by singleton probabilities. Based on a CRN factorization, we propose an efficient two-step separate training method: (i) in the first step, we train a separate model for each singleton probability; (ii) given singleton probabilities, we train a separate model for each CR factor. Experimental results on three important natural language processing tasks show that our separate training method is two orders of magnitude faster than conditional random fields, while achieving competitive quality (often better on the overall quality metric F1).

The second contribution of this thesis is applying PGMs to a real-world NLP application: open relation extraction (ORE). In open relation extraction, two entities in a sentence are given, and the goal is to automatically extract their relation expression. ORE is a core technique, especially in the age of big data, for transforming unstructured information into structured data. We propose our model SimpleIE for this task. The basic idea is to decompose an extraction pattern into a sequence of simplification operations (components). The benefit by doing this is that these components can be re-combined in a new way to generate new extraction patterns. Hence SimpleIE can represent and capture diverse extraction patterns. This model is essentially a sequence labeling model. Experimental results on three benchmark data sets show that SimpleIE boosts recall and F1 by at least 15% comparing with seven ORE systems.

As tangible outputs of this thesis, we contribute open source implementations of our research results as well as a annotated data set: (i) Co-occurrence rate networks on chain-structured graphs¹. (ii) SimpleIE for open relation extraction.² (iii) Annotated data for fostering the research on open relation extraction.

¹<https://github.com/zheminzhu/Co-occurrence-Rate-Networks>

²SimpleIE and the annotated data are available upon request (zhuzhemin@gmail.com).

Samenvatting

Afhankelijkheid is een universeel fenomeen dat overal geobserveerd kan worden. In machine learning worden afhankelijkheidsrelaties gerepresenteerd door probabilistic graphical models (PGMs). PGMs hebben een breed toepassingsgebied in natural language processing (NLP), spraakherkenning, computer vision, biomedicine, information retrieval, etc. Veel traditionele modellen, zoals hidden Markov models (HMMs) of Kalman filters, kunnen gevat worden onder de term PGM. Het centrale idee is om een simultane kansverdeling te opdelen (factorizeren) naar een product van lokale factoren. Leren, inferentie en opslag kunnen allen efficient uitgevoerd worden op de representatie van de factorisatie.

We onderscheiden twee types PGMs: (i) Bayesiaanse netwerken (gerichte grafen), en (ii) Markov netwerken (ongerichten grafen). Bayesiaanse netwerken representeren *gerichte afhankelijkheid* with *gerichte zijden*. Lokale factoren in een Bayesiaans netwerk zijn *voorwaardelijke kansen*. Gerichte afhankelijke, gerichte zijden en voorwaardelijke kansen zijn allen *asymmetrische* noties. In contrast, Markov netwerken representeren *mutuele afhankelijkheid* met *ongerichte zijden*. Zowel mutuele afhankelijkheid als ongerichte zijden zijn *symmetrische* noties. Voor Markov netwerken, gebaseerd op het Hammersley-Clifford theorema, lokale factoren zijn *positieve functies* over maximum cliques. Deze lokale factoren worden uitgelegd met intuïtieve noties als 'compatibiliteit' of 'affiniteit'. In het bijzonder, als de graaf een clique tree is, kan de simultane kansverdeling hergeparameteriseerd worden naar een junction tree factorization.

In deze proefschrift stellen we een nieuw framework voor dat gemotiveerd is met het Minimum Shared Information Principle (MSIP):

We proberen een factorisatie te vinden waarvoor de informatie die gedeeld is tussen factoren minimaal is. In andere woorden: we maken factoren zo onafhankelijke mogelijk.

Het voordeel van deze aanpak is dat we factoren los kunnen trainen zonder veel aandacht te besteden aan het garanderen van consistentie tussen de factoren. Om dit doel te bereiken en een dergelijk factorisatie te verkrijgen hebben we een theoretisch raamwerk ontwikkeld dat we co-occurrence rate networks (CRNs)

noemen. In het kort, gegeven een simultane verdeling, verkrijgen we de CRN factorisatie als volgt: We delen de simultane verdeling op in univariate verdelingen. De overgebleven waarde noemen we de co-occurrence rate (CR). CR is een symmetrische waarde die de mutuele afhankelijkheid tussen de gemoeide variabelen aanduidt. Daarna wordt de simultane CR verder gedeclineerd naar kleinere onafhankelijke CRs. Uiteindelijk komen we tot een CRN factorisatie waarvan de factoren alleen nog maar univariate verdelingen en CR factoren zijn. Er bestaan twee soorten onafhankelijkheid tussen deze factoren: (i) een univariate verdeling is onafhankelijk, hier bedoeld als twee factoren die geen informatie delen, van andere univariate verdelingen; (ii) een CR factor is onafhankelijk van andere CR factoren gegeven door univariate verdelingen. We stellen een efficiënt twee-staps gescheiden trainingsmethode voor gebaseerd op CRN factorisatie: (i) in de eerste stap trainen we een los model voor elke univariate verdeling; (ii) we trainen, gegeven de univariate verdelingen, een los model voor elke CR factor. Experimentele resultaten van drie belangrijke natural language processing taken tonen dat onze gescheiden trainingsmethode twee orde groottes sneller is dan conditional random fields, en een competitieve kwaliteit bereikt (en vaak betere scoort op de algemene kwaliteitsgraad F1).

De tweede bijdrage van deze proefschrift is het toepassen van PGMs op een bestaande NLP toepassing: open relation extraction (ORE). In open relation extraction worden twee entiteiten in een zin aangegeven, en het doel is het automatisch de relatie tussen de twee entiteiten te bepalen. ORE is een centrale techniek, zeker in het tijdperk van big data, voor het transformeren van ongestructureerde naar gestructureerde data. We stellen ons model SimpleIE voor voor deze taak. Het basale idee is het opdelen van een extractiepatroon naar een opeenvolging van simplificerende operaties (componenten). Het voordeel van deze aanpak is dat deze componenten op andere wijzen kunnen worden gecombineerd om zo nieuwe extractiepatronen te produceren. Hierdoor kan SimpleIE uiteenlopende extractiepatronen representeren en beschrijven. Dit model is in essentie een *sequence labelling model*. Experimentele resultaten op drie benchmark data sets tonen dat SimpleIE een recall en F1 waardes verbeterd met minstens 15% ten opzichte van zeven ORE systemen.

Als tastbare producten van deze proefschrift dragen we zowel open source implementaties van onze resultaten, als een geannoteerde data set: (i) Co-occurrence rate networks on chain-structured graphs.³ (ii) SimpleIE voor open relation extraction.⁴ (iii) Geannoteerde data voor het vooruitbrengen van onder-

³<https://github.com/zheminzhu/Co-occurrence-Rate-Networks>

⁴SimpleIE en de geannoteerde data set zijn beschikbaar op aanvraag (zhuzhem@gmail.com).

zoek naar open relation extraction.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Questions	5
1.3	Contributions	6
1.4	Thesis Structure	7
I	Co-occurrence Rate Networks	9
2	Probabilistic Graphical Models	11
2.1	Motivation: the Decomposition Strategy	11
2.2	Conditional Independence and Probability Factorization	15
2.3	Bayesian Networks	18
2.4	Markov Networks	21
2.5	Inference	23
2.6	Learning	33
3	Co-occurrence Rate Networks	43
3.1	Co-occurrence Rate	44
3.2	Examples	53
3.3	The Hypertree Representation	62
3.4	Co-occurrence Rate Networks	66
3.5	Inference	69
3.6	Learning	69
4	Two-step Separate Training for CRNs	71
4.1	Maximum Likelihood Estimation of Co-occurrence Rate Networks	71
4.2	Separate Models	72
4.3	Consistency	74
4.4	Summary	77

5	Experiments on Chain-structured CRNs	79
5.1	Named Entity Recognition	79
5.2	Part-of-speech Tagging	82
5.3	Related Models	84
5.4	CRNs are Immune to the Label Bias Problem	89
5.5	Training and Decoding	90
5.6	Experiments	91
5.7	Summary	94
II	Open Relation Extraction	95
6	A Review of Open Relation Extraction	97
6.1	Introduction	97
6.2	Quality Metrics	99
6.3	Technical Aspects	103
7	SimpleIE: a Simplification Model for Open Relation Extraction	109
7.1	Motivated by Examples	110
7.2	The Model: SimpleIE	117
7.3	Wikipedia Dataset	123
7.4	Experiments	125
7.5	Noun Phrase Recognition	130
7.6	Related Work on Sentence Simplification	131
7.7	Summary	132
III	Conclusion	133
8	Conclusions and Future Work	135
8.1	General Conclusions	135
8.2	Research Questions Revisited	136
8.3	Future Work	139
	Appendices	143
A	Appendix	145
A.1	Axioms of Probability	145
A.2	Proof of $\mathcal{I}_l(G) \Leftrightarrow F_{BN}(G)$	145
A.3	Proof of the Hammersley-Clifford Theorem	146

CONTENTS	xvii
Bibliography	149
Publications by the Author	155
SIKS Dissertation Series	159

Introduction

1.1 Motivation

Applications A wide range of applications in natural language processing (NLP), speech processing, computer vision, biomedicine, information retrieval [1], and many other areas desire structured outputs [2]. For example, named entity recognition (Section 5.1) and part-of-speech tagging (Section 5.2) assign a sequence of labels to words in a sentence. The outputs are chain-structured labels. A syntax parser transforms sentences to parse trees. The outputs are tree-structured labels. The task of predicting structured outputs is called structured prediction. Structured prediction is significantly different from the ordinary classification task which normally predicts a single label. The difficulty of structured prediction is that multiple labels need to be predicted together and there are dependence relations between these labels.

Probabilistic graphical models Structured prediction can¹ be put under the umbrella of probabilistic graphical models (PGMs) [3, 4]. It turns out that many traditional models, such as hidden Markov models, Kalman filters, language models, etc., which were previously developed in different areas, can be put under the general framework of PGMs. PGMs ground on systematic and solid theories. The central idea of PGMs is to decompose a joint probability into a product of local factors based on (conditional) independence relations. A local factor reflects dependence relations among the variables involved in the local factor. Two major types of PGMs can be distinguished:² directed graphs

¹There are exceptions. For example, structural SVMs [25] are also popular models for structured prediction, which essentially apply decomposition to kernels. But due to their lack of an obvious probabilistic interpretation, they cannot be easily put under PGMs.

²Another type of PGMs are factor graphs. Factor graphs can be used to represent the factorization of a Bayesian network or a Markov network. Variables involved in the same factor are grouped

(Bayesian networks) and undirected graphs (Markov networks).

Bayesian Networks The graphical representation of Bayesian networks is a directed acyclic graph (DAG). *Directed edges* are naturally *asymmetric*, i.e., the edge $A \rightarrow B$ is distinguished from the edge $B \rightarrow A$. Directed edges are suitable to model *directed dependence*, e.g., causality³. Directed dependence is an asymmetric concept. Hence, directed edges fits the asymmetry of directed dependence. For Bayesian networks, the local factors are conditional probabilities⁴, which are also asymmetric. We summarize the model aspects of Bayesian networks in Table 1.1.

Table 1.1: Model Aspects of Bayesian Networks

Relation	Representation	Factors	Symmetry
Directed dependence	Directed edges	Conditional probabilities	Asymmetric

Markov Networks In contrast, Markov networks are represented with undirected graphs. *Undirected edges* are suitable to model *mutual dependence*. For mutual dependence, we cannot specify a direction. For example, in named entity recognition, two adjacent labels affect each other mutually. They are at equal position. Mutual dependence is a *symmetric* notion. Generally, based on the Hammersley–Clifford theorem [5], a joint probability over a Markov network can be decomposed into a product of positive functions over maximum cliques. Unfortunately, unlike conditional probabilities which are used within Bayesian networks, these positive functions do not have a direct probabilistic interpretation. They are related but not sufficient to specify marginals over maximum cliques. Because we still need information from adjacent factors to obtain marginals from these positive functions. Normally they are explained using intuitive notions like ‘compatibility’ or ‘affinity’. Specially, for a graph which forms (or is triangulated into) a clique tree, its joint probability has an

together. Factor graphs are suitable for inference and learning, but not for modeling independence because they do not directly encode conditional independencies between variables. In this sense, factor graphs are significantly different from Bayesian networks and Markov networks. Therefore, we do not put factor graphs together with Bayesian networks and Markov networks.

³For a causal relation, the effect depends on the cause. But the cause does not necessarily depend on the effect. Note that a causal relation is a directed dependence relation. But a directed dependence relation is not necessarily a causal relation.

⁴For a node without any parents, the local factor is a singleton probability.

alternative representation which is called junction tree factorization (reparameterization) [6, 7, 8, 9] in which a product of marginals over cliques is divided by a product of marginals over overlapped parts (called separator sets) between cliques⁵. We summarize the model aspects of general Markov networks and clique trees in Table 1.2 and Table 1.3, respectively.

Table 1.2: Model Aspects of General Markov Networks

Relation	Representation	Factors	Symmetry
Mutual dependence	Undirected edges	Positive functions	Symmetric

Table 1.3: Model Aspects of Clique Trees

Relation	Representation	Factors	Symmetry
Mutual dependence	Undirected edges	Marginals	Symmetric

Co-occurrence Rate Networks In this thesis, we propose a novel framework motivated by the Minimum Shared Information Principle (MSIP):

Given a joint probability, we try to find a factorization in which the information shared between factors is minimum. In other words, we try to make factors as independent as possible.

The benefit by doing this is that we can train factors separately without paying a lot of efforts to guarantee consistency between them. The shared information between two factors can be intuitively defined as the information which needs to be calibrated between two factors to achieve consistency. For example, between two factors $P(X, Y)$ and $P(Y, Z)$, the shared information is $P(Y)$. Towards this goal, we develop a theoretical framework called co-occurrence rate networks (CRNs) to obtain such a factorization. Briefly, given a joint probability, its CRN factorization is obtained as follows. We first strip off singleton probabilities from the joint probability. The quantity left is called co-occurrence rate (CR). CR is a symmetric quantity which measures mutual dependence among variables involved. Then we further decompose the joint CR into smaller and independent CRs if it is possible. Finally, we obtain a CRN factorization whose factors consist of all singleton probabilities and CR factors.

⁵Here we consider the collection of separator sets as a multiset (bag).

The important properties of a CRN factorization are described as follows. There exist two kinds of independencies between the factors in a CRN factorization: (i) a singleton probability is independent⁶ of other singleton probabilities; (ii) a CR factor is independent of other CR factors conditioned by singleton probabilities involved. We summarize the model aspects of co-occurrence rate networks in Table 1.4.

Table 1.4: Model Aspects of Co-occurrence Rate Networks

Relation	Representation	Factors	Symmetry
Mutual dependence	Undirected edges	Co-occurrence rates & singleton probabilities	Symmetric

Benefits The benefit of minimizing the shared information between factors is that we can train independent factors separately. This leads to a more efficient training algorithm. The strategy of separate training is not a new one. Piecewise training [10] which follows tree re-weighted parameterization [6] is just based on this strategy. Based on the properties of a CRN factorization, we propose a two-step separate training algorithm: (i) In the first step, we train a singleton (univariate) probability separately. Comparing with multi-variate marginals, singleton probabilities are relatively easier to train. (ii) In the second step, we fix the learned singleton probabilities and train CR factors separately. This is allowed because conditioned by singleton probabilities, CR factors are independent of each other. In the decoding step, we assemble these separate models together for prediction. Experimental results on three important natural language processing tasks, i.e., named entity recognition, part-of-speech tagging and open relation extraction, show that our separate training method is almost two orders of magnitudes faster than conditional random fields while achieving competitive quality.

The main model aspects of Bayesian networks, Markov networks, clique trees and co-occurrence rate networks described above are summarized in Table 1.5.

⁶Here independent means two factors do not share information.

Table 1.5: Bayesian Networks, (general) Markov Networks, Clique Trees and Co-occurrence Rate Networks

	BN	MN	CT	CRN
Dependence	directed	mutual	mutual	mutual
Representation	directed	undirected	undirected	undirected
Factors	(conditional) probabilities	positive functions	marginals	co-occurrence rates & singleton probabilities
Symmetry	asymmetric	symmetric	symmetric	symmetric
Normalization	local	global	local	local
Closed-form MLE	yes	no	yes	yes

1.2 Research Questions

This thesis aims at two research goals: (i) On the theoretical side, we propose a systematic framework for obtaining a factorization in which factors share minimum information; (ii) On the application side, we are interested in applying graphical models to open relation extraction. In open relation extraction, given two entities in a sentence, the goal is to automatically extract their relation expression. ORE is a core technique, especially in the age of big data, to transform unstructured information into structured data. We may manually compile a set of rules (extraction patterns) for extracting relation expressions. But due to the diversity of extraction patterns, manually compiled extraction patterns normally cannot achieve a high recall (see experiments in Chapter 7), and also manually compiling is labor intensive. To achieve these two research goals, we need to explore the following specific research questions.

1.2.1 Part I: Co-occurrence Rate Networks

- Q1** *How to obtain a factorization in which factors share minimum information? More specifically, what independence semantics should be endowed? Is the factorization equivalent to the given independencies?*
- Q2** *How to prove the theory of co-occurrence rate networks is sound? Soundness can be verified by proofs or experiments.*
- Q3** *What are the advantages of co-occurrence rate networks? Do CRNs bring any added value?*

1.2.2 Part II: Open Relation Extraction

- Q4** *How to model diverse extraction patterns with graphical models?* Considering the diversity of extraction patterns, the model should be general enough to represent diverse patterns. Also the model is expected to be able to automatically learn extraction patterns from training data.
- Q5** *How to evaluate the system?* To evaluate our system, we need to compare it with other systems on benchmark datasets.
- Q6** *How well do CRNs perform on the task of open relation extraction comparing with Markov networks?* Are the results consistent with our expectations.

1.3 Contributions

Our major contributions in this thesis can be concluded as follows:

- C1** Motivated by the Minimum Shared Information Principle (MSIP), we propose co-occurrence rate networks (CRNs) to obtain a factorization in which factors share minimum information between each other. And based on CRNs we propose a separate training method which is efficient and achieves good quality. This is supported by real-world experiments. A CRN factorization can be considered a special case of the hypertree factorization proposed by Wainwright et al. [6]. The specificity of CRNs stems from the emphasis on MSIP.
- C2** We propose a general model called SimpleIE for open relation extraction. This model can represent and capture diverse extraction patterns in training data.
- C3** We implement chain-structured co-occurrence rate networks, and make this software open source, which can be downloaded at <https://github.com/zheminzhu/Co-occurrence-Rate-Networks>.
- C4** We implement SimpleIE for open relation extraction, and make the software open source. This software is available upon request by sending a message to zhuzhemin@gmail.com.
- C5** We annotate a Wikipedia dataset for fostering research on open relation extraction. This data set is available upon request by sending a message to zhuzhemin@gmail.com.

1.4 Thesis Structure

This thesis consists of two parts. Part I presents the framework of co-occurrence rate networks and the two-step separate training method. Part II describes our open relation extraction system SimpleIE. The remainder of this thesis is organized as follows.

1.4.1 Part I: Co-occurrence Rate Networks

- Chapter 2. *Probabilistic Graphical Models*. In this chapter, we review fundamental results in PGMs. This chapter provides the general background for reading Part I.
- Chapter 3. *Co-occurrence Rate Networks*. In this chapter, we develop the quantity co-occurrence rate for modeling mutual dependence, and give its nice properties. Upon co-occurrence rate, we build co-occurrence rate networks.
- Chapter 4. *Two-step Separate Training for CRNs*. In this chapter we propose a two-step separate training method for CRNs.
- Chapter 5. *Experiments*. We apply CRNs to two important natural language processing tasks: named entity recognition and part-of-speech tagging, and compare CRNs with Markov networks.

1.4.2 Part II: Open Relation Extraction

- Chapter 6. *A Review of Open Relation Extraction*. This chapter introduces the task of open relation extraction and reviews related work.
- Chapter 7. *SimpleIE: a Simplification Model for Open Relation Extraction*. In this chapter, we develop our model SimpleIE for open relation extraction. We also compare our model with 7 state-of-the-art systems on 3 benchmark data sets.

1.4.3 Part III: Conclusion

- Chapter 8. *Conclusions and Future Work*.

Part I

Co-occurrence Rate Networks

*To see a world in a grain of sand, and a heaven in a wild flower.
Hold infinity in the palm of your hand, and eternity in an hour.
- William Blake*

CHAPTER 2

Probabilistic Graphical Models

Outline

In this chapter, we review fundamental results in probabilistic graphical models (PGMs). This chapter provides the general background for reading the following chapters in Part I. Koller and Friedman [3], Bishop [4] and other references provide the main results described in this chapter. This chapter also explains a bit of my perception on this topic.

This chapter is organized as follows. We first introduce the motivation and basic ideas of PGMs in Section 2.1. Then following the traditional presentation structure in this area, we discuss the representation, inference and learning aspects of directed graphs (Bayesian networks) and undirected graphs (Markov networks). In this thesis, we focus on categorical random variables. Data are assumed fully observed, and structures of graphs are assumed known.

2.1 Motivation: the Decomposition Strategy

PGMs represent (in)dependence relations with graphs. PGMs ground on two basic ideas:

1. *Decomposition.* A joint probability is decomposed into a product of local factors.
2. *Visualization.* Independence relations and the equivalent factorization can be read from graph structures.

Decomposition is the most fundamental idea of PGMs. Visualization provides a WYSIWYG (what you see is what you get) representation of abstract concepts. A vivid description of PGMs is “Graphical models are a marriage between

probability theory and graph theory.” given by Michael I. Jordan, UC Berkeley, 1998.

2.1.1 Decomposition

One general and powerful strategy to attack a complex object is *to decompose it into simple components* which we can handle. Decomposition is just the most fundamental idea of PGMs. In fact, we widely use this strategy in everyday life as well as in mathematics and science, both explicitly or implicitly. The complex object can be a bed in our bedroom, a matrix in linear algebra, a function in harmonic analysis, a compact set in topology, a joint probability in PGMs, etc.

Bed Suppose we relocate to another city, and need to move a large bed with us. The bed as a whole is too big to be put in our car. The idea everyone can think of is to decompose the bed into smaller components, such as legs, frames, slats, etc. Then we load these components to our car and deliver them to the destination. Hopefully, if nothing is lost in decomposition and delivery, we can get the original bed back by assembling its components. Note that we should decompose the bed following its structure rather than chopping it into pieces brutally.

Readers may skip the following three mathematical examples (Function, Matrix and Compact Set) if they have never heard of them. These examples are given to convince readers that the decomposition strategy is one¹ fundamental idea in mathematics. These three examples are not related to other parts of this thesis.

Function To understand a function, we can decompose it into a sum of simple functions which we know well. In *Fourier transformation*, a wave-like function is decomposed as a sum of sines and cosines. Sines and cosines are simple and well known functions to us. Hence, instead of treating the original difficult function, we can manipulate its Fourier series. Another kind of simple functions are polynomials. The famous Stone–Weierstrass theorem (see 11.15 in the textbook by Apostol [11]) states any continuous function on a closed interval can be uniformly approximated as closely as desired by polynomials. This theorem allows us to (approximately) decompose a function into a sum of polynomials. This is very useful in practice, e.g., linear regression using polynomials as

¹Beside decomposing, there are other powerful strategies, such as mapping, limiting (extremely approximating), constraining, disturbing, etc. But decomposition is the focus of this thesis.

basis functions. Also note that we prefer to use orthogonal basis functions in decomposing a function. Because in this way the components are completely independent of each other (the projection is zero). In other words, the components do not share information between each other. This simplifies the representation. In spirit, our Minimum Shared Information Principle (MSIP) proposed in this thesis, i.e., finding a factorization in which factors share minimum information, also follows this principle.

Matrix Similarly, the decomposition strategy plays a critical role in linear algebra [12]. A fundamental result in linear algebra is that: given an ordered basis, there exists an isomorphism (one to one, onto and linear mapping) between linear transformations and matrix representations.² This result allows us to represent a linear transformation by a matrix. Hence, to understand a linear transformation we can study its matrix representation which is more tangible. But a matrix can still be difficult to understand. In such cases, we can decompose the matrix into a product of several simple matrices. This is called *matrix factorization*. This corresponds to decomposing a linear transformation into a composition of several simple linear transformations. For different purposes, there are different ways to factorize a matrix. This is because ‘simple’ has different meaning for different purposes. But they share the common motivation: simplifying by decomposition. We give two examples of them:

1. *Solving linear systems* To solve a system of linear equations: $AX = Y$, where A is a matrix which transforms a vector X into a vector Y by applying the left multiplication to X , we decompose $A = E_1 E_2 \dots E_n G$, where $\{E_i : i = 1, \dots, n\}$ are elementary matrices which are matrix representations of Gaussian elimination operations, and G is a matrix in reduced row echelon form. Elementary and echelon matrices are simple matrices for this task. As elementary matrices are invertible, we obtain $GX = E_n^{-1} \dots E_1^{-1} Y = Y'$. $GX = Y'$ is simple enough because G is in reduced row echelon form. We can obtain the solution directly by back-substitution. As we see, in solving linear systems, the key step is to decompose A into a product of elementary matrices and a matrix in reduced row echelon form.
2. *Diagnolization* Another example is diagonalization $A = PDP^{-1}$, where A is the matrix that we want to understand, D is a diagonal matrix, P is the

²In other words, the action of a linear transformation on a vector space (with a finite dimension) is uniquely determined by its action on an ordered basis of the vector space.

change of basis matrix which consists of eigenvectors of A . A diagonal matrix is simple because its behavior in left multiplication is well known to us: just scaling by its diagonal elements which are eigenvalues. P is also simple which changes the basis. The behavior of A is decomposed into three steps: (1) P^{-1} changes coordinates to the eigenvector basis; (2) D scales by its diagonal elements; (3) finally, P changes coordinates back to the original basis. Each of these three steps is simple. Hence, we well understand A .

Compact set Compactness plays an extremely important role in mathematical analysis. This concept is derived from the Heine–Borel theorem (see 3.11 in the textbook by Apostol [11]), and was introduced into mathematics by Maurice Fréchet³ in his PhD dissertation. For a set (of points), we can use a collection of small open sets to cover the set. In other words, a set can be decomposed into a collection of open sets in the sense of covering. For a compact set, any infinite open cover (an infinite collection of open sets) can be reduced to a finite sub-cover, which is a finite subset of the infinite collection of open sets. It turns out that on a compact set, the local information contained in these small open sets can be passed to the global information contained in the whole compact set. Compactness is a bridge from local to global. For example, the continuity of a function, which is a local property considered within a small open set surrounding a point, can be passed to the uniform continuity of the function, which is a global property considered over the whole set, on a compact set. That is a continuous function on compact set is uniformly continuous. Here we see the basic idea is to decompose a set into a collection of small open sets.

Practical models We use decomposition everywhere. Many widely used practical models happen to follow this strategy:

1. *Language models* [13] decompose the probability of a sequence of tokens into a product of conditional probabilities over adjacent words in the segment.
2. *Statistical machine translation models* [14] decompose the translation function into a composition of a segmentation function, a re-ordering function and a substitution function.

³Maurice Fréchet also introduced the statistical framework Copula. The continuous co-occurrence rate proposed in this thesis happens to be the density function of Copula. See Chapter 3 for details.

3. *Mixture models* [4] decompose a target distribution into a sum of simple distributions, e.g., Gaussians.
4. *Deep learning models* [15] decompose a target function into a composition of multiple levels of non-linear functions.
5. In the *MapReduce* framework [16], a complex task is first decomposed into independent sub-tasks. Then the Map procedure processes these independent sub-tasks separately. Finally, the Reduce procedure assembles outputs from the Map procedure to form the final result for the original task.

Probability Probability is the target object of PGMs. High-dimensional joint probability is not easy to handle. Learning, inference and storage can be inefficient for a high-dimensional joint probability. PGMs decompose a joint probability into a *product of local factors*. This is called *probability factorization*. It turns out that learning, inference and storage can be conducted much more efficiently⁴ over the factorization representation than over the original joint probability. Recall that in decomposing a bed, we follow the structure of the bed rather than chopping it into pieces. Similarly, to decompose a joint probability we also need to follow the structure of the joint probability. Naturally, independence relations serve as the structure of the joint probability.

2.1.2 Visualization

Visualization is the second basic idea of PGMs. PGMs use graphs to encode a set of independencies as well as an equivalent factorization, which makes these abstract concepts more tangible. The benefit by doing this is that we can read the independencies and the factorization from the graph structure in a predefined way. Moreover the independencies and the factorization read are guaranteed to be equivalent.

2.2 Conditional Independence and Probability Factorization

Conditional independence is an important concept in PGMs. They serve as the structure of a joint probability. Given this structure, we can decompose a joint

⁴when the graph is sparse

probability into a product of local factors. We first introduce notations which will be used throughout this thesis.

Notation A set or a vector of random variables is denoted by calligraphy symbols, such as $\mathcal{X}$ and $\mathcal{Y}$. $\mathcal{X}_i$ is the i th component of $\mathcal{X}$ if it is a vector. $\mathbf{x}$ is an assignment to $\mathcal{X}$. Correspondingly, $\mathbf{x}_i$ is an assignment to $\mathcal{X}_i$. $\text{Val}(\mathcal{X})$ is the set of all possible values that can be assigned to $\mathcal{X}$. We also use X and Y to represent single random variables, and their assignments are denoted by x and y , respectively. Let $\mathcal{S}$ be a set, then $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$ which indicates the number of elements in $\mathcal{S}$. Also we sometimes use ‘variable’ as an abbreviation for ‘random variable’.

Definition 1 (Conditional Independence).

$$(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z}) \Leftrightarrow P(\mathcal{X}, \mathcal{Y} \mid \mathcal{Z}) = P(\mathcal{X} \mid \mathcal{Z})P(\mathcal{Y} \mid \mathcal{Z}) \quad (2.1)$$

In this definition, $(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z})$ is a conditional independency which means $\mathcal{X}$ is independent of $\mathcal{Y}$ given $\mathcal{Z}$. A special case is unconditioned independence:⁵ $(\mathcal{X} \perp\!\!\!\perp \mathcal{Y}) \Leftrightarrow P(\mathcal{X}, \mathcal{Y}) = P(\mathcal{X})P(\mathcal{Y})$. In this thesis we use unconditioned independence and conditional independence interchangeably. $(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z})$ in Equation 2.1 is called an independency, and $P(\mathcal{X}, \mathcal{Y} \mid \mathcal{Z}) = P(\mathcal{X} \mid \mathcal{Z})P(\mathcal{Y} \mid \mathcal{Z})$ is called a factorization. They are defined to be equivalent ($\Leftrightarrow$).

Equivalence Note the bi-direction of the equivalence between the independency $(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z})$ and the factorization $P(\mathcal{X}, \mathcal{Y} \mid \mathcal{Z}) = P(\mathcal{X} \mid \mathcal{Z})P(\mathcal{Y} \mid \mathcal{Z})$ in Equation 2.1. This not only means the independency implies the factorization, but also means the factorization implies the independency. A more complex factorization can imply a set of sub-factorizations, and hence it can imply a set of independencies. For example,

$$\begin{aligned} P(X, Y, Z) &= P(X)P(Y)P(Z) \\ &\Rightarrow \{P(X, Y) = P(X)P(Y), P(Y, Z) = P(Y)P(Z), P(X, Z) = P(X)P(Z)\} \\ &\Rightarrow \{(X \perp\!\!\!\perp Y), (Y \perp\!\!\!\perp Z), (Z \perp\!\!\!\perp X)\}, \end{aligned}$$

where the first step can be obtained by marginalization, and the second step follows directly from Definition 1.

Therefore, given a set of independencies $\mathcal{I}$ and a factorization F , a question naturally arising is: are they equivalent? Or using symbols, $\mathcal{I} \Leftrightarrow F$? If $\mathcal{I} \Leftrightarrow F$, we say $\mathcal{I}$ is an equivalent independency set of F .

⁵In other words, the assignment of $\mathcal{Z}$ is just the whole sample space.

Completeness Let F be a factorization. The set of *all* independencies implied by F is denoted by $\mathcal{I}(F)$. We say $\mathcal{I}(F)$ is complete to F . Any true subset of $\mathcal{I}(F)$, denoted by $\mathcal{I} \subset \mathcal{I}(F)$, is not complete to F .

Theorem 1. *If there exists $\mathcal{I}$ such that $\mathcal{I} \Leftrightarrow F$, then $\mathcal{I}(F) \Leftrightarrow F$.*

This theorem can be proved as follows.

Proof. Suppose there is a set of independencies $\mathcal{I}$ satisfying $\mathcal{I} \Leftrightarrow F$. Hence $F \Rightarrow \mathcal{I}$, we have $\mathcal{I} \subseteq \mathcal{I}(F)$, thus $\mathcal{I}(F) \Rightarrow \mathcal{I}$. Because $F \Rightarrow \mathcal{I}(F)$ and $\mathcal{I}(F) \Rightarrow \mathcal{I} \Rightarrow F$, we have $\mathcal{I}(F) \Leftrightarrow F$. $\square$

This theorem implies that $\mathcal{I}(F) \Leftrightarrow F$ if we can find any $\mathcal{I}$ that is equivalent to F . Hence we can say $\mathcal{I}(F)$ is the maximum equivalent independency set of F . The following example shows that even though a set of independencies $\mathcal{I}$ is not complete to F , $\mathcal{I}$ can be equivalent to F . In other words, an equivalent independency set of F is not necessarily complete to F .

Example Given $\mathcal{I} = \{(X \perp\!\!\!\perp Y, Z)\}$, we have:

$$\begin{aligned} \mathcal{I} &= \{(X \perp\!\!\!\perp Y, Z)\} \\ &\Rightarrow F : P(X, Y, Z) = P(X)P(Y, Z) \\ &\Rightarrow \{P(X, Y, Z) = P(X)P(Y, Z), P(X, Y) = P(X)P(Y), P(X, Z) = P(X)P(Z)\} \\ &\Rightarrow \mathcal{I}(F) = \{(X \perp\!\!\!\perp Y, Z), (X \perp\!\!\!\perp Y), (X \perp\!\!\!\perp Z)\} \\ &\Rightarrow \mathcal{I} \end{aligned}$$

The first step follows directly from Definition 1, and the second step can be obtained by marginalization. In this example, we have $\mathcal{I} \Leftrightarrow F$, because $\mathcal{I} \Rightarrow F$ and $F \Rightarrow \mathcal{I}$. But $\mathcal{I}$ is not complete to F , because $\mathcal{I} \subset \mathcal{I}(F)$. This example shows that there exist equivalent independency set of F that are not the maximum equivalent independency set of F .

Hence, given a factorization F , and an equivalent independency set of F , denoted by $\mathcal{I}$, it is interesting to check if $\mathcal{I}$ is the maximum (complete) equivalent independency set of F .

Remark Naturally, there is another interesting independency set: minimum equivalent independency set of F . In contrast to the maximum (complete) equivalent independency set of F discussed above, minimum equivalent independency set of F is equivalent to F but contains minimum number of independencies.

Equivalence and *completeness* are two important considerations in the representation of PGMs. In PGMs, we endow a set of independencies $\mathcal{I}$ and an equivalent factorization F to a graph. $\mathcal{I}$ and F are encoded by a graph structure. Hence, we can decode (read) $\mathcal{I}$ and F from the graph. Furthermore, if $\mathcal{I}$ is complete to F , we can read all independencies implied by F from the graph structure.

2.3 Bayesian Networks

The graphical representation of a Bayesian network is a directed acyclic graph (DAG), denoted by G . Each node in G represents a random variable. A directed edge between two nodes indicates there is a directed dependency between them. Section 2.3.1 gives an example of Bayesian networks.

We first endow a set of independencies $\mathcal{I}_l$ and a factorization F_{BN} to a graph G . Then we show: (i) $\mathcal{I}_l$ and F_{BN} are equivalent; but normally (ii) $\mathcal{I}_l$ is not complete to F_{BN} .

Definition 2 (Local Independencies $\mathcal{I}_l$). *Let G be a directed acyclic graph, and $\mathcal{X}$ be nodes of G . We endow the following set of independencies to G :*

$$\mathcal{I}_l = \{(\mathcal{X}_i \perp\!\!\!\perp \text{ND}_i - \text{Pa}_i \mid \text{Pa}_i) : \forall \mathcal{X}_i \in \mathcal{X}\}, \quad (2.2)$$

where Pa_i is the set of all parents of $\mathcal{X}_i$, and ND_i is the set of all non-descendants of $\mathcal{X}_i$.

That is for each node in the graph, it is independent of its non-descendants (see Section 2.3.1 for an example) given its parents.

Definition 3 (Bayesian Network Factorization F_{BN}). *Let G be a directed acyclic graph, and $\mathcal{X}$ be nodes of G , we endow the following factorization to G :*

$$F_{\text{BN}} : P(\mathcal{X}) = \prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i \mid \text{Pa}_i) \quad (2.3)$$

where Pa_i is the set of all parents of $\mathcal{X}_i$ in G .

Given a graph G , $\mathcal{I}_l$ and F_{BN} can be read from the graph structure according to the Definition 2 and Definition 3.

Theorem 2 (Relationship Between $\mathcal{I}_l$ and F_{BN}).

$$\begin{aligned} \mathcal{I}_l &\Leftrightarrow F_{\text{BN}} \\ \mathcal{I}_l &\subseteq \mathcal{I}(F_{\text{BN}}) \end{aligned}$$

The proof of this theorem is given in Appendix A.2. The first equation means $\mathcal{I}_l$ is equivalent to F_{BN} . And the second equation means $\mathcal{I}_l$ is generally not complete to F_{BN} . In other words, $\mathcal{I}_l$ is not the maximum equivalent independency set of F_{BN} . Hence, we continue to find the complete independency set of F_{BN} . $\mathcal{I}_d$ (Definition 6) which is defined based on the notion of d-separation is complete to F_{BN} .

Definition 4 (Active Path). *Let G be a directed acyclic graph, and $\mathcal{X}$ be the nodes of G . $[\mathcal{X}_i, \dots, \mathcal{X}_j]$ is a path (both directions are allowed for edges in the path) from node $\mathcal{X}_i$ to $\mathcal{X}_j$. We say the path $[\mathcal{X}_i, \dots, \mathcal{X}_j]$ is active given $\mathcal{Z} \subset \mathcal{X}$ if it satisfies both of*

1. $\forall (\mathcal{X}_{k-1} \rightarrow \mathcal{X}_k \leftarrow \mathcal{X}_{k+1})$ in $[\mathcal{X}_i, \dots, \mathcal{X}_j]$, $\mathcal{X}_k$ or one of its descendants are in $\mathcal{Z}$.
2. No other node of the path is in $\mathcal{Z}$.

$(\mathcal{X}_{k-1} \rightarrow \mathcal{X}_k)$ means the edge is from $\mathcal{X}_{k-1}$ to $\mathcal{X}_k$, and $(\mathcal{X}_k \leftarrow \mathcal{X}_{k+1})$ is the edge from $\mathcal{X}_{k+1}$ to $\mathcal{X}_k$. $(\mathcal{X}_{k-1} \rightarrow \mathcal{X}_k \leftarrow \mathcal{X}_{k+1})$ forms a V structure. See Section 2.3.1 for an example.

Definition 5 (D-separation). *Let G be a directed acyclic graph, and $\mathcal{X}$ be the nodes of G . And $\mathcal{V}, \mathcal{W}$ and $\mathcal{U}$ are subsets of $\mathcal{X}$. We say $\mathcal{V}$ and $\mathcal{W}$ are d-separated by $\mathcal{U}$, denoted by $d\text{-sep}(\mathcal{V}; \mathcal{W} \mid \mathcal{U})$, if $\forall \mathcal{V}_i \in \mathcal{V}, \mathcal{W}_j \in \mathcal{W}$, there is no active path between $\mathcal{V}$ and $\mathcal{W}$ given $\mathcal{U}$.*

D-separation stands for ‘directed separation’. We further define the set of independencies judged by the d-separation criterion as follows.

Definition 6 (D-separation Independencies $\mathcal{I}_d$). *Let G be a directed acyclic graph, and $\mathcal{X}$ be the nodes of G . And $\mathcal{V}, \mathcal{W}$ and $\mathcal{U}$ are subsets of $\mathcal{X}$.*

$$\mathcal{I}_d = \{(\mathcal{V} \perp\!\!\!\perp \mathcal{W} \mid \mathcal{U}) : \forall d\text{-sep}(\mathcal{V}; \mathcal{W} \mid \mathcal{U})\}$$

Theorem 3. $\mathcal{I}_d$ is the complete equivalent independency set of F_{BN} . That is

$$\mathcal{I}_d \Leftrightarrow F_{\text{BN}} \quad \text{and} \quad \mathcal{I}_d = \mathcal{I}(F_{\text{BN}})$$

The proof is too technical to be included in this thesis. A rough sketch of the proof is given in Theorem 3.4 in the textbook by Koller and Friedman [3]. We summarize the relationship between $\mathcal{I}_l$, F_{BN} , and $\mathcal{I}_d$ as follows:

Equivalence:	$\mathcal{I}_l \Leftrightarrow F_{\text{BN}} \Leftrightarrow \mathcal{I}_d$
Completeness:	$\mathcal{I}_l \subseteq \mathcal{I}(F_{\text{BN}}) = \mathcal{I}_d$

2.3.1 An Example of Bayesian Networks

In this section, we give an example to explain the concepts described above. Figure 2.1 depicts a Bayesian network.

Figure 2.1: A Bayesian Net

The dependence assumptions in this Bayesian network are described as follows. Incidence of Earthquake (E) or Burglary (B) depends on the City (C). An earthquake or burglary can be detected by sensors and causes an Alarm (A) in the police office. Also earthquake will cause a Radio (R) report. There is no direct dependence between earthquake and burglary.

In this Bayesian network, the node E has one parent $\text{Pa}_E = \{C\}$. Its non-descendants are $\text{ND}_E = \{C, B\}$. Hence $\text{ND}_E - \text{Pa}_E = \{B\}$. The local independencies $\mathcal{I}_l$ are given as follows:

$$\mathcal{I}_l = \{(E \perp\!\!\!\perp B \mid C), (B \perp\!\!\!\perp E, R \mid C), (A \perp\!\!\!\perp C, R \mid E, B), (R \perp\!\!\!\perp C, B, A \mid E)\}$$

The factorization endowed to this Bayesian network is:

$$F_{\text{BN}} : P(C, E, B, A, R) = P(C)P(E \mid C)P(B \mid C)P(A \mid E, B)P(R \mid E)$$

The path $[E, A, B]$ forms a V structure. $[E, A, B]$ is an active path given $\{A\}$. In other words, if we focus on $\{E, A, B\}$ and omitting C temporally, we have $(E \not\perp\!\!\!\perp B \mid A)$, but $(E \perp\!\!\!\perp B)$ without the condition A . In contrast, $[E, C, B]$ does not form a V structure, and $[E, C, B]$ is not active given $\{C\}$. In other words, in this path, $(E \perp\!\!\!\perp B \mid C)$. The d-separation independencies $\mathcal{I}_d$ of this Bayesian network are given as follows:

$$\mathcal{I}_d = \{(E \perp\!\!\!\perp B \mid C), (B \perp\!\!\!\perp E, R \mid C), (A \perp\!\!\!\perp C, R \mid E, B), (R \perp\!\!\!\perp C, B, A \mid E), \\ (B \perp\!\!\!\perp R \mid E, A), (B \perp\!\!\!\perp R \mid C), \dots\}$$

There are many more independencies in $\mathcal{I}_d$. We do not list all of them here. But this is already enough to show that $\mathcal{I}_l \subset \mathcal{I}_d$ in this example. In other words, in this example $\mathcal{I}_l$ is equivalent but not complete to F .

2.4 Markov Networks

The representation of Markov networks is based on undirected graphs. An example of Markov networks is illustrated in Section 2.4.1.

We first endow a set of independencies $\mathcal{I}_{\text{MN}}$ and a factorization F_{MN} to the graph. They can be proved equivalent. The independencies are judged by the u-separation (undirected separation) criterion.

Definition 7 (U-separation). *Let G be an undirected graph, and $\mathcal{X}$ be all nodes of G . $\mathcal{V}$, $\mathcal{W}$ and $\mathcal{U}$ are disjoint subsets of $\mathcal{X}$. We say $\mathcal{U}$ separates $\mathcal{V}$ and $\mathcal{W}$, denoted by $\text{sep}(\mathcal{V}; \mathcal{W} | \mathcal{U})$, if $\forall \mathcal{V}_i \in \mathcal{V}, \mathcal{W}_j \in \mathcal{W}$, if for any path between $\mathcal{V}$ and $\mathcal{U}$, at least one node of the path is in $\mathcal{U}$.*

Remark The definition of u-separation just formalizes our intuition of separation.

Definition 8 (Markov Network Independencies $\mathcal{I}_{\text{MN}}$). *Let G be an undirected graph, and $\mathcal{X}$ be all nodes of G . $\mathcal{V}$, $\mathcal{W}$ and $\mathcal{U}$ are subsets of $\mathcal{X}$.*

$$\mathcal{I}_{\text{MN}} = \{(\mathcal{V} \perp\!\!\!\perp \mathcal{W} | \mathcal{U}) : \forall \text{sep}(\mathcal{V}; \mathcal{W} | \mathcal{U})\}. \quad (2.4)$$

Then we endow the following factorization to the graph.

Definition 9 (Markov Network Factorization F_{MN}). *Let G be an undirected graph, and $\mathcal{X}$ be all nodes of G . $\mathcal{C}$ are all (maximum) cliques of G :*

$$F_{\text{MN}} : P(\mathcal{X}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi_c(c) \quad (2.5)$$

where the factors ϕ_c are positive functions defined over c , and Z is a normalizing constant.

Cliques are complete sub-graphs. In other words, in a clique any pair of nodes are connected by an edge.

Theorem 4 (Hammersley-Clifford Theorem [5]).

$$\mathcal{I}_{\text{MN}} \Leftrightarrow F_{\text{MN}}$$

We give an elegant proof of this theorem based on the theory of co-occurrence rate in Section 3.2.5. And it is obvious that $\mathcal{I}_{MN} = \mathcal{I}(F_{MN})$. Hence $\mathcal{I}_{MN}$ is equivalent and complete to F_{MN} . There are other two sets of independencies which are equivalent but not complete to F_{MN} . These incomplete equivalent independency sets could be useful in drawing the graph structure.

Definition 10 (Pairwise Independencies $\mathcal{I}_p$). *Let G be an undirected graph, $\mathcal{X}$ be all nodes of G , and $\mathcal{E}$ be all edges of G . We define the following set of independencies:*

$$\mathcal{I}_p = \{(X \perp\!\!\!\perp Y \mid \mathcal{X} - \{X, Y\}) : \forall X, Y \in \mathcal{X}; (X, Y) \notin \mathcal{E}\}$$

Definition 11 (Local Independencies $\mathcal{I}_l$). *Let G be an undirected graph, and $\mathcal{X}$ be all nodes of G . We define the following set of independencies:*

$$\mathcal{I}_l = \{(X \perp\!\!\!\perp \mathcal{X} - \{X\} - \mathcal{N}_X \mid \mathcal{N}_X) : \forall X \in \mathcal{X}\}$$

where $\mathcal{N}_X$ are neighbours of X in G .

It is easy to prove $\mathcal{I}_p \Leftarrow \mathcal{I}_{MN}$, $\mathcal{I}_l \Leftarrow \mathcal{I}_{MN}$, and $\mathcal{I}_p \Leftarrow \mathcal{I}_l$. Hence, we have

$$\mathcal{I}_p \Leftarrow \mathcal{I}_l \Leftarrow \mathcal{I}_{MN}$$

If we prove $\mathcal{I}_p \Rightarrow \mathcal{I}_{MN}$, we obtain a complete cycle which implies $\mathcal{I}_p \Leftrightarrow \mathcal{I}_l \Leftrightarrow \mathcal{I}_{MN}$. The proof can be found in Theorem 4.4 in the textbook by Koller and Friedman [3] which shows $\mathcal{I}_p \Rightarrow \mathcal{I}_{MN}$ on positive distributions. The relationship between them can be summarized as follows:

Equivalence: $\mathcal{I}_p \Leftrightarrow \mathcal{I}_l \Leftrightarrow \mathcal{I}_{MN} \Leftrightarrow F_{MN}$ Completeness: $\mathcal{I}_l \subseteq \mathcal{I}_{MN} = \mathcal{I}(F_{MN}), \mathcal{I}_p \subseteq \mathcal{I}_{MN}$
--

2.4.1 An Example of Markov Networks

In this section, we borrow the Example 3.8 in the text book by Koller and Friedman [3] to explain the concepts described above. Figure 2.2 depicts this example.

Suppose four students $\{S_A, S_B, S_C, S_D\}$ discuss the correctness of an explanation given by their professor. (S_A, S_B) are friends and they communicate with each other. (S_B, S_C) , (S_C, S_D) and (S_D, S_A) are also friends. But (S_A, S_C) and (S_B, S_D) have bad relations. They do not talk with each other directly. The binary random variables $\{A, B, C, D\}$ in the Figure 2.2 represent their judgments. The judgment can be ‘correct’ or ‘incorrect’.

Figure 2.2: A Markov Network

There are four mutual dependencies: (A, B) , (B, C) , (C, D) and (D, A) in this graph. For example, A affects B and also B affects A . Hence, undirected edges should be used here. A and C are separated by $\{B, D\}$, and also B and D are separated by $\{A, C\}$. Hence, the Markov network independencies can be written as follows:

$$\mathcal{I}_{\text{MN}} = \{(A \perp\!\!\!\perp C \mid B, D), (B \perp\!\!\!\perp D \mid A, C)\}.$$

There are four maximum cliques (complete sub-graphs) in this graph: $\{A, B\}$, $\{B, C\}$, $\{C, D\}$, and $\{D, A\}$. Hence, the Markov network factorization can be written as follows:⁶

$$F_{\text{MN}} : P(A, B, C, D) = \frac{1}{Z} \phi_1(A, B) \phi_2(B, C) \phi_3(C, D) \phi_4(D, A)$$

For this simple graph, pairwise independencies $\mathcal{I}_p$ and local independencies $\mathcal{I}_l$ happen to be equal to $\mathcal{I}_{\text{MN}}$.

2.5 Inference

Given a joint probability $P(\mathcal{X})$, inference is the process to draw sub-probabilities, such as marginal probabilities (marginals) or conditional probabilities, from $P(\mathcal{X})$. The joint probability $P(\mathcal{X})$ contains all information of its sub-probabilities. To make a decision, normally we are interested in some sub-probabilities rather than the joint probability. In this case, we need inference. Also inference is often employed as a sub-routine in the learning process. If we consider $P(\mathcal{X})$ as a database, basically, there can be three types of queries:

⁶The factorization representation is not unique. See the remark below the Theorem 16 in Chapter 3 for details.

1. *Marginals*: $P(\mathcal{Y})$, where $\mathcal{Y} \subseteq \mathcal{X}$.
2. *Conditional probability*: $P(\mathcal{Y} | \mathcal{E} = \mathbf{e})$, where $\mathcal{Y}, \mathcal{E} \subseteq \mathcal{X}$, and $\mathbf{e}$ is the observed evidence. As $P(\mathcal{Y} | \mathcal{E} = \mathbf{e}) = \frac{P(\mathcal{Y}, \mathbf{e})}{P(\mathbf{e})}$ and $P(\mathbf{e}) = \sum_{\mathbf{y} \in \text{Val}(\mathcal{Y})} P(\mathbf{y}, \mathbf{e})$, the original query $P(\mathcal{Y} | \mathcal{E} = \mathbf{e})$ can be reduced to $P(\mathcal{Y}, \mathbf{e})$.
3. *Maximum conditional probability*: $\arg\max_{\mathbf{y} \in \text{Val}(\mathcal{Y})} P(\mathbf{y} | \mathcal{E} = \mathbf{e})$, where $\mathcal{Y}, \mathcal{E} \subseteq \mathcal{X}$. That is to find an assignment of $\mathcal{Y}$ which maximizes a conditional probability.

When the joint probability is of very high dimensions, inference can be intractable. For PGMs, inference can be done much more efficiently (if the graph is sparse) over the factorization representation, which shows the benefit of decomposition. In the rest of this section, we summarize two important methods for inference on PGMs: variable elimination (VE) and belief propagation (BP), which can be applied to both Bayesian networks and Markov networks. In VE and BP, factors are treated as positive functions. In other words, when VE or BP is applied to a Bayesian network, the probability interpretation of the factors, which are conditional probabilities, is ignored.

2.5.1 Variable Elimination

2.5.1.1 Marginals

The idea of variable elimination is to apply the marginalization operation to the factorization representation of a joint probability to eliminate variables and obtain marginals. The marginalization operation holds two important properties: commutative property and localizable property. These two properties allow us to localize (restrict) a singleton marginalization operation to a few of factors. Localization leads to more efficient computation. We first define the marginalization operation. Then we show the marginalization operation is commutative and localizable. Finally, with these two properties on hand, we apply the marginalization operation to the factorization representation of a joint probability to obtain the general variable elimination process.

Definition 12 (Marginalization Operation $\sum_{\mathcal{X}}$).

$$\sum_{\mathcal{X}} : \mathcal{F}(\mathcal{X}, \mathcal{Y}) \rightarrow \mathcal{G}(\mathcal{Y}) \quad f(\mathcal{X}, \mathcal{Y}) \mapsto g(\mathcal{Y}) = \sum_{\mathbf{x} \in \text{Val}(\mathcal{X})} f(\mathbf{x}, \mathcal{Y}),$$

where $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ are all functions defined on $(\mathcal{X}, \mathcal{Y})$, and $\mathcal{G}(\mathcal{Y})$ are all functions defined on $\mathcal{Y}$.

That is the marginalization operation $\sum_{\mathcal{X}}$ is a function whose input is a function $f(\mathcal{X}, \mathcal{Y})$ and output is another function $g(\mathcal{Y})$. The output $g(\mathcal{Y})$ is calculated by summing $f(\mathcal{X}, \mathcal{Y})$ over all values of $\mathcal{X}$.

Theorem 5 (Marginalization of Probability). *Let $P(\mathcal{X}, \mathcal{Y})$ be a joint probability. Then,*

$$P(\mathcal{Y}) = \sum_{\mathcal{X}} P(\mathcal{X}, \mathcal{Y}),$$

where $\sum_{\mathcal{X}}$ is the marginalization operation defined in Definition 12.

The proof can be sketched as follows. The collection of events $\{(\mathcal{X} = \mathbf{x}, \mathcal{Y} = \mathbf{y}) : \mathbf{x} \in \text{Val}(\mathcal{X})\}$ forms a partition (mutually exclusive and exhaustive) of the event $(\mathcal{Y} = \mathbf{y})$. Then this theorem directly follows the third axiom of probability (Appendix A.1).

Theorem 6 (Commutative Property of Marginalization). *$P(\mathcal{X}, \mathcal{Y})$ is a joint probability. $[\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n]$ is an arbitrary permutation of all variables in $\mathcal{X}$. Then,*

$$\sum_{\mathcal{X}} P(\mathcal{X}, \mathcal{Y}) = \sum_{\mathcal{X}_1} \sum_{\mathcal{X}_2} \dots \sum_{\mathcal{X}_n} P(\mathcal{X}_1, \dots, \mathcal{X}_n, \mathcal{Y}) = P(\mathcal{Y}),$$

$\sum_{\mathcal{X}_i}$ is a singleton marginalization operation. This theorem states that we can arbitrarily reorder singleton marginalization operations without changing the result. The proof is obvious.

Definition 13 (Factor Product). *Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be disjoint sets of variables. $\phi_1(\mathcal{X}, \mathcal{Y})$ and $\phi_2(\mathcal{Y}, \mathcal{Z})$ are two real-valued functions. Their product $\phi(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ is defined as:*

$$\phi : (\text{Val}(\mathcal{X}), \text{Val}(\mathcal{Y}), \text{Val}(\mathcal{Z})) \rightarrow \mathcal{R} \quad (\mathbf{x}, \mathbf{y}, \mathbf{z}) \mapsto \phi_1(\mathbf{x}, \mathbf{y})\phi_2(\mathbf{y}, \mathbf{z})$$

For convenience, $\phi(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ is also denoted by $\phi_1(\mathcal{X}, \mathcal{Y})\phi_2(\mathcal{Y}, \mathcal{Z})$.

Note that this definition implies the factor product is only defined when the assignments to the shared variables $\mathcal{Y}$ in ϕ_1 and ϕ_2 are identical. Otherwise, the product is undefined. According to this definition, we have that $\phi_1(\mathcal{X})[\phi_2(\mathcal{Y}) + \phi_3(\mathcal{Y})] = \phi_1(\mathcal{X})\phi_2(\mathcal{Y}) + \phi_1(\mathcal{X})\phi_3(\mathcal{Y})$, no matter $\mathcal{X}$ and $\mathcal{Y}$ are disjoint or not. The following localizable property is critical to make the inference over a factorization representation more efficient than over the original joint probability.

Theorem 7 (Localizable Property of Marginalization over Factor Product). *Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be sets of variables, where $\mathcal{X} \cap \mathcal{Z} = \emptyset$. ϕ_1 and ϕ_2 are real-valued functions defined over $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then,*

$$\sum_{\mathcal{Z}} [\phi_1(\mathcal{X}) \phi_2(\mathcal{Y})] = \phi_1(\mathcal{X}) \sum_{\mathcal{Z}} \phi_2(\mathcal{Y}). \quad (2.6)$$

$\mathcal{X}$ and $\mathcal{Y}$ can be disjoint or not. This theorem states that the marginalization operation $\sum_{\mathcal{Z}}$ can be localized to $\phi_2(\mathcal{Y})$ when $\mathcal{X} \cap \mathcal{Z} = \emptyset$. This theorem can be proved as follows.

Proof. According to Definition 12, we have

$$\begin{aligned} & \sum_{\mathcal{Z}} [\phi_1(\mathcal{X}) \cdot \phi_2(\mathcal{Y})] \\ &= \sum_{\mathbf{z} \in \text{Val}(\mathcal{Z})} [\phi_1(\mathcal{X}) \cdot \phi_2(\mathbf{z}, \mathcal{Y} - \mathcal{Z})] \\ &= \phi_1(\mathcal{X}) \sum_{\mathbf{z} \in \text{Val}(\mathcal{Z})} \phi_2(\mathbf{z}, \mathcal{Y} - \mathcal{Z}) \\ &= \phi_1(\mathcal{X}) \sum_{\mathcal{Z}} \phi_2(\mathcal{Z}, \mathcal{Y} - \mathcal{Z}) \\ &= \phi_1(\mathcal{X}) \sum_{\mathcal{Z}} \phi_2(\mathcal{Y}) \end{aligned}$$

□

We note that in Equation 2.6 the computation of the right hand side is more efficient than the left hand side. For example, $a(b + c)$ can be calculated more efficiently than $(ab + ac)$. There are one addition and one multiplication in $a(b + c)$. But there are one addition and two multiplications in $(ab + ac)$. The marginalization operation is localized to b and c in $a(b + c)$.

With the commutative and localizable properties on hand, we derive the variable elimination process as follows. This process is also called sum-product algorithm. Without loss of generality, let $P(\mathcal{X}) = \frac{1}{Z} \prod_i \phi_i(S_i)$, where $S_i \subseteq \mathcal{X}$ are the variables in factor ϕ_i . $P(\mathcal{Y})$, where $\mathcal{Y} \subset \mathcal{X}$, is the marginal probability to be inferred, and $\mathcal{M} = \mathcal{X} - \mathcal{Y} = \{\mathcal{X}_1, \dots, \mathcal{X}_k\}$, and $[\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k]$ is an arbitrary permutation of $\mathcal{M}$. We first partition all factors into $k + 1$ groups $[G_1, G_2, \dots, G_k, G_r]$ with respect to the order $[\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k]$. The group G_i w.r.t. X_i consists of all factors containing X_i that have not been grouped yet. The remained factors

form an additional group G_r . We denote the product of all factors in G_i by $\prod(G_i)$. If $G_i = \emptyset$, we define $\prod(G_i) = 1$. Then the variable elimination process can be described as follows.

$$\begin{aligned}
 P(\mathcal{Y}) &= \sum_{\mathcal{M}} P(\mathcal{X}) \\
 &= \sum_{\mathcal{X}_k} \dots \sum_{\mathcal{X}_2} \sum_{\mathcal{X}_1} \left[\frac{1}{Z} \prod_i \phi_i(\mathcal{S}_i) \right] \\
 &= \sum_{\mathcal{X}_k} \dots \sum_{\mathcal{X}_2} \sum_{\mathcal{X}_1} \left[\frac{1}{Z} \prod(G_r) \prod(G_k) \dots \prod(G_2) \prod(G_1) \right] \\
 &= \frac{1}{Z'} \prod(G_r) \left[\sum_{\mathcal{X}_k} \prod(G_k) \dots \left[\sum_{\mathcal{X}_2} \prod(G_2) \left[\sum_{\mathcal{X}_1} \prod(G_1) \right] \dots \right] \right],
 \end{aligned} \tag{2.7}$$

where Z' is the new normalizing constant which equals the sum over $\mathcal{Y}$. These equations follow directly from the commutative and the localizable properties of the marginalization operation. In this process, the joint marginalization operation is decomposed into an arbitrary order of singleton marginalization operations. Then these singleton marginalization operations are localized to groups. Each group consists of a few factors in the factorization.

Remark Besides the marginalization operation described in this section, there are other operations holding commutative and localizable properties. The maximization operation defined in Section 2.5.1.3 and the expectation operation defined in Theorem 14 are two more examples. As long as an operation holds these two properties, it can be localized to reduce computation costs.

2.5.1.2 Conditional Probability

Variable elimination can also be used to draw a conditional probability $P(\mathcal{Y}|\mathcal{E} = \mathbf{e})$ which can be reduced to $P(\mathcal{Y}, \mathcal{E} = \mathbf{e})$. $P(\mathcal{Y}, \mathcal{E} = \mathbf{e})$ can be obtained by fixing $\mathcal{E} = \mathbf{e}$ and apply the same marginalization process to $P(\mathcal{X})$ to eliminate variables in $(\mathcal{X} - \mathcal{Y} - \mathcal{E})$.

2.5.1.3 Maximum Conditional Probability

We find an assignment which maximizes a conditional probability: $\arg\max_{\mathcal{Y}} P(\mathcal{Y}|\mathcal{E} = \mathbf{e})$. This task is equivalent to: $\max_{\mathcal{Y}} P(\mathcal{Y}, \mathcal{E} = \mathbf{e})$. Because during the maximizing process, we can record the corresponding assignment and fix $\mathcal{E} = \mathbf{e}$.

Similar to the marginalization operation, the maximization operation also holds the commutative and localizable properties. Hence, a maximization operation can also be localized to a few factors in the factorization representation to make the computation more efficient.

In this section, we first define the maximization operation. Then we show it is commutative and localizable. Finally, we derive the max-sum algorithm which localizes maximization operations to a few factors. The max-sum algorithm is very similar to the variable elimination process. This only major difference is that in variable elimination we use marginalization operation, and max-sum algorithm we use the maximization operation.

Definition 14 (Maximization Operation).

$$\max_{\mathcal{X}} : \mathcal{F}(\mathcal{X}, \mathcal{Y}) \rightarrow \mathcal{G}(\mathcal{Y}) \quad f(\mathcal{X}, \mathcal{Y}) \mapsto g(\mathcal{Y}) = \max\{f(\mathbf{x}, \mathcal{Y}) : \mathbf{x} \in \text{Val}(\mathcal{X})\}.$$

Theorem 8 (Commutative Property of Maximization). *$P(\mathcal{X})$ is a joint probability. $[\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_n]$ is an arbitrary permutation of all variables in $\mathcal{X}$. Then,*

$$\max_{\mathcal{X}} P(\mathcal{X}) = \max_{\mathcal{X}_1} \max_{\mathcal{X}_2} \dots \max_{\mathcal{X}_n} P(\mathcal{X}_1, \dots, \mathcal{X}_n).$$

The proof of this theorem is obvious.

Definition 15 (Factor Plus). *Let $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$ are disjoint sets of variables. $\theta_1(\mathcal{X}, \mathcal{Y})$ and $\theta_2(\mathcal{Y}, \mathcal{Z})$ are two real-valued functions. The plus $\theta(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ is defined as follows.*

$$\theta : (\text{Val}(\mathcal{X}), \text{Val}(\mathcal{Y}), \text{Val}(\mathcal{Z})) \rightarrow \mathcal{R} \quad (\mathbf{x}, \mathbf{y}, \mathbf{z}) \mapsto \theta_1(\mathbf{x}, \mathbf{y}) + \theta_2(\mathbf{y}, \mathbf{z}).$$

For convenience, $\theta(\mathcal{X}, \mathcal{Y}, \mathcal{Z})$ is denoted by $\theta_1(\mathcal{X}, \mathcal{Y}) + \theta_2(\mathcal{Y}, \mathcal{Z})$.

Note that this definition implies that the plus of θ_1 and θ_2 is only defined when the assignments to the shared variables $\mathcal{Y}$ in θ_1 and θ_2 are identical. Otherwise, the plus is undefined.

Theorem 9 (Localizable Property of Maximization over Factor Plus). *Let $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$ be sets of random variables, where $\mathcal{X} \cap \mathcal{Z} = \emptyset$. $\theta_1(\mathcal{X})$ and $\theta_2(\mathcal{Y})$ are two real-valued functions. Then,*

$$\max_{\mathcal{Z}} [\theta_1(\mathcal{X}) + \theta_2(\mathcal{Y})] = \theta_1(\mathcal{X}) + \max_{\mathcal{Z}} \theta_2(\mathcal{Y}).$$

Proof. According to the Definition 14, we have

$$\begin{aligned}
 & \max_{\mathcal{Z}} [\theta_1(\mathcal{X}) + \theta_2(\mathcal{Y})] \\
 &= \max \{ \theta_1(\mathcal{X}) + \theta_2(\mathbf{z}, \mathcal{Y} - \mathcal{Z}) : \mathbf{z} \in \mathcal{Z} \} \\
 &= \theta_1(\mathcal{X}) + \max \{ \theta_2(\mathbf{z}, \mathcal{Y} - \mathcal{Z}) : \mathbf{z} \in \mathcal{Z} \} \\
 &= \theta_1(\mathcal{X}) + \max_{\mathcal{Z}} \theta_2(\mathcal{Z}, \mathcal{Y} - \mathcal{Z}) \\
 &= \theta_1(\mathcal{X}) + \max_{\mathcal{Z}} \theta_2(\mathcal{Y})
 \end{aligned}$$

□

A max-sum process which is similar to the variable elimination (Equation 2.7) can be derived as follows. The only difference is replacing $\sum_{\mathcal{X}}$ with $\max_{\mathcal{X}}$.

$$\begin{aligned}
 \max_{\mathcal{X}} P(\mathcal{X}) &\propto \max_{\mathcal{X}} \ln P(\mathcal{X}) \\
 &= \max_{\mathcal{X}} \ln \left[\frac{1}{Z} \prod_i \phi_i(\mathcal{S}_i) \right] \\
 &\propto \max_{\mathcal{X}} \ln \left[\prod_i \phi_i(\mathcal{S}_i) \right] \\
 &= \max_{\mathcal{X}} \left[\sum_i \theta_i(\mathcal{S}_i) \right] \\
 &= \max_{\mathcal{X}_k} \dots \max_{\mathcal{X}_2} \max_{\mathcal{X}_1} \left[\sum_i \theta_i(\mathcal{S}_i) \right] \\
 &= \max_{\mathcal{X}_k} \dots \max_{\mathcal{X}_2} \max_{\mathcal{X}_1} \left[\sum (G_k) + \dots + \sum (G_2) + \sum (G_1) \right] \\
 &= \max_{\mathcal{X}_k} (G_k) \dots \left[\max_{\mathcal{X}_2} \sum (G_2) \left[\max_{\mathcal{X}_1} \sum (G_1) \right] \dots \right]
 \end{aligned}$$

2.5.2 Belief Propagation

A variable elimination (VE) process can only answer one query. The intermediate results, which can be re-used for answering other queries, are lost. Belief propagation is a more efficient way to answer a set of queries in a dynamic programming paradigm. Belief propagation on tree-structured graphs can result in exact marginals. But on general graphs, beliefs may not be exact marginals.

2.5.2.1 Belief Propagation on Clique Trees

Belief propagation on a clique tree can be considered as running multiple VE processes asynchronously. The intermediate results are stored in nodes and edges of the clique tree. They can be re-used between multiple VE processes. Hence, belief propagation can answer a set of queries more efficiently than executing multiple VE processes separately.

A clique tree is denoted by $T(\mathcal{C}, \mathcal{E})$, where $\mathcal{C}$ are nodes and $\mathcal{E}$ are edges. A clique tree can be constructed from a VE process as follows. Suppose the VE process eliminates variables in the order of $[\mathcal{X}_1, \dots, \mathcal{X}_k]$. Let $\Phi = \{\phi_1, \phi_2, \dots, \phi_n\}$ be all factors in the factorization representation of a joint probability. Φ can be partitioned into $k + 1$ groups $[G_1, G_2, \dots, G_k, G_{k+1}]$ as described in the last several paragraphs in Section 2.5.1.1. For each group G_i , we set a node $\mathcal{C}_i$ in the clique tree. Hence, there are $k + 1$ nodes $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k, \mathcal{C}_{k+1}\}$ in the a clique tree. For each node $\mathcal{C}_i$, we set an undirected edge $\mathcal{E}_{ij}$ between $\mathcal{C}_i$ and $\mathcal{C}_j$. $\mathcal{C}_j$ is found as follows. let $\mathcal{M} = \{m : \mathcal{C}_i \cap \mathcal{C}_m \neq \emptyset, i < m\}$, then $j = \min(\mathcal{M})$. We label the edge $\mathcal{E}_{ij}$ with $S_{ij} = \mathcal{C}_i \cap \mathcal{C}_j$ which is called *sepsset* (separate set).

The clique tree constructed from a VE process meets the following two important properties.

Definition 16 (Family Preservation). *Each factor ϕ_i is assigned to one and only one node in the clique tree.*

This is obvious because in the first step, all factors are partitioned into groups (mutually exclusive and exhaustive). And then each group is mapped to one node in the clique tree.

Definition 17 (Running Intersection Property). *If $\mathcal{X}_k \in \mathcal{C}_i$ and $\mathcal{X}_k \in \mathcal{C}_j$, then $\mathcal{X}_k$ is in every node of the (unique) path between $\mathcal{C}_i$ and $\mathcal{C}_j$.*

First, there is a unique path between $\mathcal{C}_i$ and $\mathcal{C}_j$. Because T is tree-structured. Suppose $\mathcal{C}_{\mathcal{X}_k}$ is the node where the variable $\mathcal{X}_k$ is eliminated. Then $\mathcal{X}_k$ must exist between $\mathcal{C}_i$ and $\mathcal{C}_{\mathcal{X}_k}$, and also $\mathcal{X}_k$ must exist between $\mathcal{C}_j$ and $\mathcal{C}_{\mathcal{X}_k}$. Hence, $\mathcal{X}_k$ is in every node between $\mathcal{C}_i$ and $\mathcal{C}_j$. The belief propagation algorithm on a clique tree can be described as follows.

Definition 18 (Initial Potential). *For a node $\mathcal{C}_i$ in a clique tree, its initial potential is the product of all factors assigned to this node, denoted by $\Phi_{\mathcal{C}_i}$.*

Definition 19 (Message). *The message from $\mathcal{C}_i$ to $\mathcal{C}_j$ is defined as follows.*

$$\delta_{i \rightarrow j} = \sum_{\mathcal{C}_i - S_{ij}} \Phi_{\mathcal{C}_i} \prod_{\mathcal{C}_k \in \text{NB}_i - \mathcal{C}_j} \delta_{k \rightarrow i},$$

where NB_i are all the neighbors of C_i , and $S_{ij} = C_i \cap C_j$.

Definition 20 (Belief). *The belief of C_i is defined as:*

$$\beta(C_i) = \Phi_{C_i} \prod_{C_k \in \text{NB}_i} \delta_{k \rightarrow i}.$$

The belief of a sepset S_{ij} is defined as:

$$\mu(S_{ij}) = \delta_{i \rightarrow j} \delta_{j \rightarrow i}.$$

Definition 21 (Ready Clique). *If C_i is ready to transmit message to C_j , that means C_i got messages from all its neighbors except C_j .*

```

for each node  $C_i$  in the clique tree do
  | Assign initial potential  $\Phi_{C_i}$  to  $C_i$ .
end
while There exists  $C_i$  which is ready to transmit message to  $C_j$  do
  | Passing message  $\delta_{i \rightarrow j}$ .
end
for each node  $C_i$  in the clique tree do
  | Calculate node belief  $\beta(C_i)$ .
end
for each edge  $\mathcal{E}_{ij}$  in the clique tree do
  | Calculate edge belief  $\mu(S_{ij})$ .
end
return node beliefs and edge beliefs

```

Algorithm 1: Belief Propagation on a Clique Tree

On clique trees, it is guaranteed that we can always find a ready clique to continue the message passing process before all messages are calculated. This is because clique trees are acyclic. It is obvious that the resulted beliefs are exact marginals because belief propagation on clique trees just executes multiple VE processes asynchronously.

2.5.2.2 Junction Tree Factorization

Traditionally, clique trees are also called junction trees (See Section 10.2.3 of [3]). According to the Definition 9, we have:

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{c \in \mathcal{C}'} \phi_c(c)$$

Note that in the inference step, all parameters of $\phi_c(c)$ have already been learned. They have already been calibrated in the learning process. Let $\tilde{P}(\mathcal{X})$ denote the un-normalized distribution. According to Definition 18, we have:

$$\tilde{P}(\mathcal{X}) = \prod_{c \in \mathcal{C}'} \phi_c(c) = \prod_{\mathcal{C}_i \in \mathcal{C}} \Phi_{\mathcal{C}_i}(\mathcal{C}_i)$$

According to Definition 20, we have the following equation immediately by canceling the common message terms in the numerator and denominator:

$$\tilde{P}(\mathcal{X}) = \frac{\prod_{\mathcal{C}_i \in \mathcal{C}} \beta_i(\mathcal{C}_i)}{\prod_{S_{ij} \in \mathcal{S}} \mu_{ij}(S_{ij})} \quad (2.8)$$

If we plug in the definitions of $\beta_i(\mathcal{C}_i)$ and $\mu_{ij}(S_{ij})$ given in Definition 20 and cancel common message terms in the numerator and denominator, we obtain Equation 2.8. Note that here $\mathcal{S}$ is a multiset (bag) which allows multiple instances. Equation 2.8 is general to all Markov networks. This is because the Definition 9 as well as Definition 18 are general to any Markov networks. When we specialize to junction trees, we obtain the following result because $\beta_i(\mathcal{C}_i)$ and $\mu_{ij}(S_{ij})$ converge to marginal probabilities on a tree graph:

$$P(\mathcal{X}) = \frac{\prod_{\mathcal{C}_i \in \mathcal{C}} P(\mathcal{C}_i)}{\prod_{S_{ij} \in \mathcal{S}} P(S_{ij})} \quad (2.9)$$

Equation 2.9 is called junction tree factorization (reparameterization) [6, 7, 8, 9].

2.5.2.3 Loopy Belief Propagation on General Cluster Graphs

A more general case to clique trees are cluster graphs. The only difference between them is that the sepset $S_{ij} \subseteq \mathcal{C}_i \cap \mathcal{C}_j$ in cluster graphs in contrast to $S_{ij} = \mathcal{C}_i \cap \mathcal{C}_j$ in clique trees. Cluster graphs also hold the family preservation property and running intersection property. But they are not necessary to be tree-structured. In other words, they may contain loops. This leads to a problem that there may be no ready clique existing before all marginals are calculated. In such a case, the loopy belief propagation algorithm iteratively updates messages until some convergence criterion is satisfied.

The convergence criterion can be $\forall \mathcal{E}_{ij} \in \mathcal{E}, \delta_{i \rightarrow j} = \delta_{j \rightarrow i}$, where $\mathcal{E}$ are all edges of the cluster graph. This convergence criterion implies calibration as defined in Definition 22.

Definition 22 (Calibrated Cluster Graph).

$$\forall \mathcal{E}_{ij} \in \mathcal{E}, \sum_{\mathcal{C}_i - S_{ij}} \beta(\mathcal{C}_i) = \sum_{\mathcal{C}_j - S_{ij}} \beta(\mathcal{C}_j).$$

```

for each node  $\mathcal{C}_i$  in the clique tree do
  | Assign initial potential  $\Phi_{\mathcal{C}_i}$  to  $\mathcal{C}_i$ .
end
while the convergence criterion not satisfied do
  | Passing message  $\delta_{i \rightarrow j}$ . //  $i \rightarrow j$  is selected according to some strategy.
end
for each node  $\mathcal{C}_i$  in the clique tree do
  | Calculate node belief  $\beta(\mathcal{C}_i)$ .
end
for each edge  $\mathcal{E}_{ij}$  in the clique tree do
  | Calculate edge belief  $\mu(S_{ij})$ .
end
return node beliefs and edge beliefs

```

Algorithm 2: Belief Propagation on a General Cluster Graph

Unfortunately, the convergence criterion does not necessarily imply exact marginals. Hence, belief propagation normally results in approximate marginals on general graphs. There are several questions arising naturally about this algorithm. For example, can this algorithm stop (existence of fixed points)? If this algorithm can stop, does it always stop at the same point (uniqueness of the fixed point)? There are many developments on this topic, such as Heskes [17, 18] and Mooij and Kappen [19].

2.6 Learning

2.6.1 The Framework of Supervised Learning

Learning of PGMs can be set in the supervised learning paradigm. Supervised learning is to find a predicted function $\tilde{f}$ to approximate a true function f^* using the data generated by f^* .⁷ There are three key components in the framework of supervised learning.

⁷Here we only consider the point estimation.

1. *Hypothesis space.* Hypothesis space, denoted by $\mathcal{H}$, is a set of functions consisting of candidates of $\tilde{f}$.
2. *Distance function.* Distance function⁸, denoted by $d(\tilde{f}, f^*)$, measures the distance between $\tilde{f}$ and f^* . But the problem is that in practice the true function f^* is unknown. Otherwise, we do not need to predict it. Thus it is impossible to measure the *real distance* $d(\tilde{f}, f^*)$. Instead we can calculate the *empirical distance* using the data $\mathcal{D}$ generated by f^* to approximate the real distance.
3. *Optimization.* Optimization is the process to search the best function in the hypothesis space with respect to the distance function.

Particularly, in PGMs, the true function to be approximated is a joint distribution $P^*(\mathcal{X})$. Hence, the learning task is also called density estimation. Let $\mathcal{D} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)})$ be IID (Independent and identically distributed) samples generated from $P^*(\mathcal{X})$. IID implies the following facts:

1. $P(\mathcal{X}^{(i)}) = P^*(\mathcal{X}), \quad i = 1, 2, \dots, N$
2. $P(\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \dots, \mathcal{X}^{(N)}) = \prod_{i=1}^N P(\mathcal{X}^{(i)}) = \prod_{i=1}^N P^*(\mathcal{X}^{(i)})$

Relative Entropy Relative entropy is widely used for measuring distance between P^* and $\tilde{P}$:

$$\begin{aligned}
 d(P^*, \tilde{P}) &:= D(P^* \| \tilde{P}) \\
 &:= E_{\mathcal{X} \sim P^*} \left[\ln \frac{P^*(\mathcal{X})}{\tilde{P}(\mathcal{X})} \right] \\
 &= E_{\mathcal{X} \sim P^*} [\ln P^*(\mathcal{X}) - \ln \tilde{P}(\mathcal{X})] \\
 &= E_{\mathcal{X} \sim P^*} [\ln P^*(\mathcal{X})] - E_{\mathcal{X} \sim P^*} [\ln \tilde{P}(\mathcal{X})]
 \end{aligned}$$

where E is the expectation. The term $E_{\mathcal{X} \sim P^*} [\ln P^*(\mathcal{X})]$ only contains the true function P^* . It is a constant with respect to $\tilde{P}$. Hence we have:

$$d(P^*, \tilde{P}) \propto -E_{\mathcal{X} \sim P^*} [\ln \tilde{P}(\mathcal{X})].$$

⁸Here the distance function is only required to be non-negative. This is different from the mathematical definition of a distance function in metric space which requires symmetry and triangle inequality as well.

To minimize the distance between $\tilde{P}$ and P^* , we can maximize $E_{\mathcal{X} \sim P^*} [\ln \tilde{P}(\mathcal{X})]$. But P^* in $E_{\mathcal{X} \sim P^*} [\ln \tilde{P}(\mathcal{X})]$ is unknown. We can use the empirical distribution $P^{\mathcal{D}}$ from data $\mathcal{D}$ to approximate it:

$$E_{\mathcal{X} \sim P^*} [\ln \tilde{P}(\mathcal{X})] \approx E_{\mathcal{X} \sim P^{\mathcal{D}}} [\ln \tilde{P}(\mathcal{X})].$$

Hence, we obtain $\tilde{P}_{\text{RE}}(\mathcal{X})$ using relative entropy as the distance function:

$$\begin{aligned} \tilde{P}_{\text{RE}} &= \operatorname{argmax}_{\tilde{P} \in \mathcal{H}} E_{\mathcal{X} \sim P^{\mathcal{D}}} [\ln \tilde{P}(\mathcal{X})] \\ &= \operatorname{argmax}_{\tilde{P} \in \mathcal{H}} \sum_{\mathbf{x} \in \text{Val}(\mathcal{X})} \left[\frac{\#(\mathbf{x}; \mathcal{D})}{|\mathcal{D}|} \ln \tilde{P}(\mathbf{x}) \right] \\ &= \operatorname{argmax}_{\tilde{P} \in \mathcal{H}} \sum_{\mathbf{x} \in \text{Val}(\mathcal{X})} [\#(\mathbf{x}; \mathcal{D}) \ln \tilde{P}(\mathbf{x})] \end{aligned}$$

where $\#(\mathbf{x}; \mathcal{D})$ is the frequency of $\mathbf{x}$ in data $\mathcal{D}$.

Maximum Likelihood Estimation Maximum likelihood estimation finds a distribution which maximizes the probability over IID data $\mathcal{D} = (\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)})$:

$$\tilde{P} = \operatorname{argmax}_{\tilde{P} \in \mathcal{H}} \prod_{i=1}^N \tilde{P}(\mathbf{x}^{(i)}).$$

$\prod_{i=1}^N \tilde{P}(\mathbf{x}^{(i)})$ is called likelihood function. To avoid underflow, in practice we use the log-likelihood function:

$$l(\mathcal{D}; \tilde{P}) = \sum_{i=1}^N \ln \tilde{P}(\mathbf{x}^{(i)})$$

Hence,

$$\begin{aligned} \tilde{P}_{\text{MLE}} &= \operatorname{argmax}_{\tilde{P} \in \mathcal{H}} l(\mathcal{D}; \tilde{P}) \\ &= \operatorname{argmax}_{\tilde{P} \in \mathcal{H}} \sum_{i=1}^N \ln \tilde{P}(\mathbf{x}^{(i)}). \end{aligned}$$

Remark It seems that there is no distance function defined in MLE. If this is true, MLE is out of the supervised learning framework described above. In fact, MLE implicitly uses the relative entropy distance. Because maximizing the log-likelihood function is equivalent to finding a distribution which is closest to the empirical distribution of the observed data:

$$\tilde{P}_{\text{MLE}} = \operatorname{argmax}_{\bar{P} \in \mathcal{H}} \sum_{i=1}^N \ln \bar{P}(\mathbf{x}^{(i)}) = \operatorname{argmax}_{\bar{P} \in \mathcal{H}} E_{\mathcal{X} \sim P^{\mathcal{D}}} [\ln \bar{P}(\mathcal{X})] = \tilde{P}_{\text{RE}}.$$

The proof is simple.

2.6.2 Maximum Likelihood Estimation for Bayesian Networks

Let G be a Bayesian network with nodes $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N\}$, and $P^*(\mathcal{X})$ be the true distribution to be estimated. Let $[\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \dots, \mathcal{X}^{(M)}]$ be IID random samples from $P^*(\mathcal{X})$. The dataset $\mathcal{D} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(M)}]$ is an assignment to the random samples. According to Definition 3, the factorization representation of a Bayesian network can be written as follows:

$$P(\mathcal{X}) = \prod_{i=1}^N P(\mathcal{X}_i | \mathcal{U}_i),$$

where $\mathcal{U}_i$ are all parents of $\mathcal{X}_i$. Hence, the hypothesis space of P^* can be set as follows:

$$\mathcal{H}_{P^*} = \{Q(\mathcal{X}) : Q(\mathcal{X}) = \prod_{i=1}^N f_i(\mathcal{X}_i, \mathcal{U}_i)\},$$

where $\{f_i(\mathcal{X}_i, \mathcal{U}_i) : i = 1, 2, \dots, N\}$ are models of the local probabilities $\{P(\mathcal{X}_i | \mathcal{U}_i) : i = 1, 2, \dots, N\}$. These local models should meet the three axioms of probability (Appendix A.1). The third axiom is about how to construct probabilities over events with multiple elements. f_i is defined over a singleton event. Hence, it only needs to meet the first two axioms. That is for all f_i :

$$\begin{aligned} f_i(\mathcal{X}_i, \mathcal{U}_i) &\geq 0, \\ \sum_{\mathbf{x}_i \in \text{Val}(\mathcal{X}_i)} f_i(\mathbf{x}_i, \mathcal{U}_i) &= 1. \end{aligned}$$

We call the first kind of constraints as *non-negativity* constraints, and the second as *local normalization* constraints.

The hypothesis space is also under the *global normalization* constraint. That is:

$$\sum_{\mathbf{x} \in \text{Val}(\mathcal{X})} Q(\mathbf{x}) = 1.$$

But this global normalization constraint is implied by the local normalization constraints. In other words, these local constraints are more strict than the global constraint. Hence, we only need to consider the local constraints henceforth. This can be proved by using marginalization operations as follows.

Proof.

$$\begin{aligned} \sum_{\mathcal{X}} Q(\mathcal{X}) &= \sum_{\mathcal{X}} \prod_{i=1}^N f_i(\mathcal{X}_i, \mathcal{U}_i) \\ &= \sum_{\mathcal{X}_N} f_N(\mathcal{X}_N, \mathcal{U}_N) \dots \sum_{\mathcal{X}_1} f_1(\mathcal{X}_1, \mathcal{U}_1) \\ &= 1, \end{aligned}$$

where $[\mathcal{X}_N, \dots, \mathcal{X}_1]$ are in a topological order of G . □

The log-likelihood function can be written as follows:

$$\begin{aligned} \mathcal{L}_Q(\mathcal{D}) &= \ln \left[\prod_{j=1}^M Q(\mathbf{x}^{(j)}) \right] \\ &= \ln \left[\prod_{j=1}^M \prod_{i=1}^N f_i(\mathbf{x}_i^{(j)}, \mathbf{u}_i^{(j)}) \right] \\ &= \ln \left[\prod_{i=1}^N \prod_{j=1}^M f_i(\mathbf{x}_i^{(j)}, \mathbf{u}_i^{(j)}) \right] \\ &= \ln \left[\prod_{i=1}^N [f_i(\mathbf{x}_i, \mathbf{u}_i)^{\#(\mathbf{x}_i, \mathbf{u}_i)}] \right] \\ &= \sum_{i=1}^N [\#(\mathbf{x}_i, \mathbf{u}_i) \ln f_i(\mathbf{x}_i, \mathbf{u}_i)] \end{aligned}$$

where $\#(\mathbf{x}_i, \mathbf{u}_i)$ is the frequency of $(\mathbf{x}_i, \mathbf{u}_i)$ in $\mathcal{D}$.

Let $f_i(\mathbf{x}_i, \mathbf{u}_i) = \theta_{\mathbf{x}_i|\mathbf{u}_i}$. To obtain the MLE, we need to maximize the $\mathcal{L}_Q(\mathcal{D})$ under the non-negativity and local normalization constraints. If we ignore the non-negativity constraints, this can be formalized as a constrained optimization problem as follows:

$$\begin{aligned} & \underset{\Theta}{\text{maximize}} && \sum_{i=1}^N \#(\mathbf{x}_i, \mathbf{u}_i) \ln \theta_{\mathbf{x}_i|\mathbf{u}_i} \\ & \text{subject to} && \sum_{\mathbf{x}_i \in \text{Val}(\mathcal{X}_i)} \theta_{\mathbf{x}_i|\mathbf{u}_i} = 1, \quad \forall i = 1, 2, \dots, N \end{aligned}$$

By the Lagrange multiplier, this constrained optimization problem can be transformed to the following un-constrained problem:

$$\underset{\Theta}{\text{maximize}} \quad \sum_{j=1}^N \#(\mathbf{x}_j, \mathbf{u}_j) \ln \theta_{\mathbf{x}_j|\mathbf{u}_j} + \sum_{i=1}^N \lambda_i \left(1 - \sum_{\mathbf{x}_i} \theta_{\mathbf{x}_i|\mathbf{u}_i}\right)$$

For each θ and λ we calculate its partial derivative and set it to 0. We can obtain a system of linear equations with respect to a fixed $\mathbf{u}_j$:

$$\begin{aligned} \forall \mathbf{x}'_j \in \text{Val}(\mathcal{X}_j), \quad & \frac{\#(\mathbf{x}'_j, \mathbf{u}_j)}{\theta_{\mathbf{x}'_j|\mathbf{u}_j}} - \lambda_j = 0 \\ 1 - \sum_{\mathbf{x}'_j \in \text{Val}(\mathcal{X}_j)} & \theta_{\mathbf{x}'_j|\mathbf{u}_j} = 0 \end{aligned}$$

Solving this linear system, we obtain the MLE of parameters for $j = 1, 2, \dots, N$:

$$\begin{aligned} \lambda_j &= \sum_{\mathbf{x}'_j \in \text{Val}(\mathcal{X}_j)} \#(\mathbf{x}'_j, \mathbf{u}_j) = \#(\mathbf{u}_j) \\ \hat{\theta}_{\mathbf{x}_j|\mathbf{u}_j} &= \frac{\#(\mathbf{x}_j, \mathbf{u}_j)}{\lambda_j} = \frac{\#(\mathbf{x}_j, \mathbf{u}_j)}{\#(\mathbf{u}_j)} \end{aligned}$$

It turns out the MLEs are just the relative frequencies in the training data $\mathcal{D}$. Hence, Bayesian networks have a closed-form MLE. Fortunately, the non-negativity constraints ignored before are automatically satisfied. Because relative frequencies can not be negative.

Remark From the derivation above, we can see that the local normalization constraints (The factors are conditional probabilities. Hence they can be normalized to 1.) is the key which leads to the closed-form MLE of Bayesian networks.

2.6.3 Maximum Likelihood Estimation for General Markov Networks

The factors of Bayesian networks are under local normalization constraints. In contrast, the factors of Markov networks cannot be locally normalized. Markov networks resorts to global normalization constraints. This is the only difference between Bayesian networks and Markov networks in learning. This difference is significant. Local normalization leads to closed-form MLE of Bayesian networks. But there is no closed-form MLE for Markov networks. The MLE of Markov networks can be derived as follows.

Let G be a Markov network with nodes $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N\}$, and $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_K\}$ be all the maximum cliques of G . $P^*(\mathcal{X})$ is the true distribution to be estimated. Let $[\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \dots, \mathcal{X}^{(M)}]$ be IID random samples from $P^*(\mathcal{X})$. The dataset $\mathcal{D} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(M)}]$ is an assignment to the random samples.

According to Definition 9, the factorization representation of a Markov network can be written as follows:

$$P^*(\mathcal{X}) = \frac{1}{Z} \prod_{i=1}^K \phi_{\mathcal{C}_i}(\mathcal{X}_{\mathcal{C}_i}),$$

where

$$Z = \sum_{\mathbf{x} \in \text{Val}(\mathcal{X})} \prod_{i=1}^K \phi_{\mathcal{C}_i}(\mathcal{X}_{\mathcal{C}_i}),$$

Z is the global normalization constraint. Hence, the hypothesis space of P^* can be set as follows:

$$\mathcal{H}_{P^*} = \{Q(\mathcal{X}) : Q(\mathcal{X}) = \prod_{i=1}^K \phi_{\mathcal{C}_i}(\mathcal{X}_{\mathcal{C}_i})\},$$

with constraints:

$$\begin{aligned}\phi_{\mathcal{C}_i}(\mathcal{X}_{\mathcal{C}_i}) &\geq 0, i = 1, 2, \dots, K \\ \sum_{\mathbf{x} \in \text{Val}(\mathcal{X})} \prod_{i=1}^K \phi_{\mathcal{C}_i}(\mathcal{X}_{\mathcal{C}_i}) &= 1.\end{aligned}$$

The log-likelihood function can be written as:

$$\begin{aligned}\mathcal{L}_Q(\mathcal{D}) &= \ln \left[\prod_{j=1}^M Q(\mathbf{x}^{(j)}) \right] \\ &= \ln \left[\prod_{j=1}^M \frac{1}{Z} \prod_{i=1}^K \phi_{\mathcal{C}_i}(\mathbf{x}_{\mathcal{C}_i}^{(j)}) \right] \\ &= \ln \left[\prod_{i=1}^K \prod_{j=1}^M \phi_{\mathcal{C}_i}(\mathbf{x}_{\mathcal{C}_i}^{(j)}) \right] - M \ln Z \\ &= \ln \left[\prod_{i=1}^K \phi_{\mathcal{C}_i}(\mathbf{x}_{\mathcal{C}_i})^{\#(\mathbf{x}_{\mathcal{C}_i})} \right] - M \ln Z \\ &= \sum_{i=1}^K [\#(\mathbf{x}_{\mathcal{C}_i}) \phi_{\mathcal{C}_i}(\mathbf{x}_{\mathcal{C}_i})] - M \ln Z,\end{aligned}$$

where M is the size of the training data.

This log-likelihood function contains the global normalization constraints by itself. Hence, we do not use the Lagrange multiplier as we did for Bayesian networks⁹. Let $\phi_{\mathcal{C}_i}(\mathbf{x}_{\mathcal{C}_i}) = \theta_{\mathbf{x}_{\mathcal{C}_i}}$. The partial derivative with respect to $\theta_{\mathbf{x}_{\mathcal{C}_i}}$ is:

$$\frac{\partial \mathcal{L}_Q(\mathcal{D})}{\partial \theta_{\mathbf{x}_{\mathcal{C}_i}}} = \frac{\#(\mathbf{x}_{\mathcal{C}_i})}{\theta_{\mathbf{x}_{\mathcal{C}_i}}} - \frac{M}{Z} \frac{\partial Z}{\partial \theta_{\mathbf{x}_{\mathcal{C}_i}}} \quad (2.10)$$

We denote $\mathcal{X} - \mathcal{X}_{\mathcal{C}_i}$ by $\mathcal{X}_{\bar{\mathcal{C}}_i}$. Then,

$$\frac{\partial Z}{\partial \theta_{\mathbf{x}_{\mathcal{C}_i}}} = \sum_{\mathbf{x}_{\bar{\mathcal{C}}_i} \in \text{Val}(\mathcal{X}_{\bar{\mathcal{C}}_i})} \prod_{\substack{k=1 \\ k \neq i}}^K \theta_{\mathbf{x}_{\mathcal{C}_k}}. \quad (2.11)$$

⁹Of course, with the Lagrange multiplier we can obtain the same results.

Plug Equation 2.11 into Equation 2.10, and set it to 0. We obtain the convergence condition for MLE of Markov networks:

$$\begin{aligned}
 \frac{\#(\mathbf{x}_{C_i})}{M} &= \frac{\theta_{\mathbf{x}_{C_i}}}{Z} \sum_{\mathbf{x}_{\bar{C}_i} \in \text{Val}(\mathcal{X}_{\bar{C}_i})} \prod_{\substack{k=1 \\ k \neq i}}^K \theta_{\mathbf{x}_{C_k}} \\
 &= \sum_{\mathbf{x}_{\bar{C}_i} \in \text{Val}(\mathcal{X}_{\bar{C}_i})} \left[\frac{\theta_{\mathbf{x}_{C_i}}}{Z} \prod_{\substack{k=1 \\ k \neq i}}^K \theta_{\mathbf{x}_{C_k}} \right] \\
 &= \sum_{\mathbf{x}_{\bar{C}_i} \in \text{Val}(\mathcal{X}_{\bar{C}_i})} \tilde{P}(\mathbf{x}_{C_i}, \mathbf{x}_{\bar{C}_i}) \\
 &= \tilde{P}(\mathbf{x}_{C_i})
 \end{aligned}$$

The left hand side is simply the empirical probability of $\mathbf{x}_{C_i}$. On the right hand side, it sums over variables in cliques except C_i . This will result in the expected marginal probability of $\mathbf{x}_{C_i}$. When the empirical probability equals the expected marginal probability, the training process will stop.

2.6.4 Summary

From the derivation in Section 2.6.2 and Section 2.6.3, we can see that for Bayesian networks, their factors can be locally normalized, which leads to a closed-form MLE. In contrast, general Markov networks do not have a closed-form MLE because their factors are known as positive functions which cannot be locally normalized.

Co-occurrence Rate Networks

Parts of this chapter have been published by Zhu et al. [20] and achieved by Zhu et al. [21].

Outline

Motivation. Co-occurrence rate networks is motivated by the Minimum Shared Information Principle (MSIP): *for a joint probability, we try to find a factorization in which the information shared between factors is minimum. In other words, we try to make factors as independent as possible.* The benefit by doing this is that we can train factors separately without paying a lot of efforts to guarantee the consistency between separate models. Towards this goal, we develop a systematic framework called co-occurrence rate networks (CRNs). Briefly, given a joint probability, a CRN factorization can be obtained as follows. We first strip off all singleton probabilities from the joint probability. The quantity left is called co-occurrence rate (CR). We continue to decompose the joint CR into smaller and independent CRs. Finally, we obtain a CRN factorization in which factors consist of CRs and all singleton probabilities. The important properties of a CRN factorization are given as follows. In a CRN factorization, (i) a singleton probability is independent of other singleton probabilities; (ii) a CR factor is independent of other CR factors conditioned by singleton probabilities.

Major related works. It turns out that a CRN factorization is a special case of the hypertree factorization proposed by Wainwright et al. [6]. Hypertrees provide a convenient representation for a family of factorizations which can be reduced (by canceling common terms in numerator and denominator) to a tree reparameterization [6, 7, 8, 9]. A CRN factorization can be reduced to a tree reparameterization and can be represented using a hypertree. Hence, a CRN factorization is a special hypertree factorization. But the general hypertree

factorization does not emphasize on MSIP, which leads to the specificity of a CRN factorization.

The idea of separate training is not a new one. In spirit, the tree-based reparameterization (TRP) [6] decomposes a general graph into hypertrees¹, and updates (trains) these hypertrees one by one in succession. The update operation is restricted to the parameters within a hypertree. This implies that a hypertree is updated separately from other hypertrees. Hence, TRP can be considered as a kind of separate training. Following TRP, piecewise training proposed by Sutton and McCallum [22] also implements the idea of separate training.

Structure. This chapter is organized as follows. Section 3.1.1 gives definitions of CR and related concepts. Then we show CR has some simple and nice properties in Section 3.1.2. In Section 3.2, examples are given to illustrate the concepts and properties of CR. In Section 3.4, we describe co-occurrence rate networks. Random variables are assumed discrete in this thesis unless explicitly stated otherwise.

Notation. A set or a vector of random variables is denoted by calligraphy symbols, such as $\mathcal{X}$ and $\mathcal{Y}$. If $\mathcal{X}$ is a vector, $\mathcal{X}_i$ is its i th component. If $\mathcal{X}$ is a set, $\mathcal{X}_i$ is an element of $\mathcal{X}$. An element of $\mathcal{X}$ can also be denoted by $\mathcal{X}_a$ or $\mathcal{X}_b$. A subset of $\mathcal{X}$ is represented by symbols with calligraphy subscripts, such as $\mathcal{X}_{\mathcal{U}}$, where $\mathcal{U}$ is an indexing set. $\mathbf{x}$ is an assignment to $\mathcal{X}$. Correspondingly, $\mathbf{x}_i$ is the assignment to $\mathcal{X}_i$. $\text{Val}(\mathcal{X})$ is the range of $\mathcal{X}$. In other words, $\text{Val}(\mathcal{X})$ is the set of all possible values that can be assigned to $\mathcal{X}$. We also use X, Y and Z to represent a singleton (univariate) random variable. Their assignments are denoted by x, y and z , respectively. If $\mathcal{S}$ is a set, $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$ which indicates the number of elements in $\mathcal{S}$. Also sometimes we use ‘variable’ as an abbreviation of ‘random variable’.

3.1 Co-occurrence Rate

In this section, we first give definitions of co-occurrence rate (CR) in Section 3.1.1. Then we go on to describe some nice properties of CR in Section 3.1.2.

¹Here we consider an ordinary tree as a special hypertree.

3.1.1 Definition of Co-occurrence Rate

Definition 23 (Pairwise Co-occurrence Rate). *Let X and Y be random variables. Co-occurrence rate between X and Y is defined as:*

$$\text{CR}(X ; Y) = \frac{P(X, Y)}{P(X)P(Y)}$$

Co-occurrence rate has intuitive interpretation:

1. If $\text{CR}(X = x ; Y = y) = 1$, then events x and y occur *independently*.
2. If $\text{CR}(X = x ; Y = y) > 1$, then x and y occur *attractively*.
3. If $0 \leq \text{CR}(X = x ; Y = y) < 1$, then x and y occur *repulsively*.

Co-occurrence rate measures the strength of dependence. And in contrast to conditional probability, i.e. $P(X | Y) = \frac{P(X, Y)}{P(Y)}$, which is used by Bayesian networks for modeling *directed dependence*, co-occurrence rate is a symmetric quantity. That is we can exchange the positions of X and Y without affecting the value of CR. In this sense, because of its symmetry, CR is suitable for measuring the strength of *mutual dependence*. Also in contrast to the positive functions used by Markov networks, CR has a probabilistic definition. CR is non-negative, and only defined when the singleton probabilities (univariate marginals) in the denominator are non-zero (For a probability mass function, non-zero implies positive). Pairwise CR happens to be the exponential function of pointwise mutual information (PMI). PMI was first introduced into computer science by Church and Hanks [23] which instantiates mutual information (MI) [24] to specific events.

Remark The underlying philosophy of CR is given as follows. The joint distribution $P(X, Y)$ contains two kinds of information: (i) information of the singleton probabilities $P(X)$ and $P(Y)$; and (ii) information of the dependence between X and Y . When the singleton probabilities are positive², these two kinds of information are separable. In this case, we can decompose $P(X, Y)$ into three atomic elements: $\text{CR}(X ; Y)$, $P(X)$ and $P(Y)$. $P(X)$ and $P(Y)$ do not share information. Hence they are independent of each other. But $\text{CR}(X ; Y)$ is not independent of $P(X)$ and $P(Y)$, because they are under a normalization constraint $\sum_{(x, y) \in \text{Val}(X, Y)} \text{CR}(x ; y)P(x)P(y) = 1$.

For convenience, we define the singleton and empty CR as follows.

²Here a positive distribution $P(X)$ means $\forall x \in \text{Val}(X), P(X = x) > 0$.

Definition 24 (Singleton Co-occurrence Rate).

$$\text{CR}(X) = 1$$

Definition 25 (Empty Co-occurrence Rate).

$$\text{CR}(\emptyset) = 1$$

By these definitions, the properties of CR given in Section 3.1.2 can be extended to the singleton or empty CR. We further extend the pairwise CR to the multivariate CR:

Definition 26 (Co-occurrence Rate). *Let $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N\}$ be a set of random variables. The co-occurrence rate is defined as:*

$$\text{CR}(\mathcal{X}_1 ; \mathcal{X}_2 ; \dots ; \mathcal{X}_N) = \frac{P(\mathcal{X}_1 ; \mathcal{X}_2 ; \dots ; \mathcal{X}_N)}{P(\mathcal{X}_1)P(\mathcal{X}_2)\dots P(\mathcal{X}_N)} \quad (3.1)$$

Continuous Co-occurrence Rate We define continuous CR in Definition 32 in F5 of Section 8.3. Continuous CR happens to be the density function of the Copula. Copula is a statistical framework for describing dependence, in which a joint cumulative distribution is constructed from singleton (univariate) cumulative distributions. Copula was started by Maurice Fréchet [25]. The Sklar's theorem [26] is the most fundamental result in Copula. Based on the Sklar's theorem, we obtain a fact that: *a continuous CR contains almost all information of the dependence relations among variables involved.* This result justifies the use of CR for modeling dependence in the continuous case. See F5 in Section 8.3 for details.

Note that we distinguish the following two notations. Let X , Y and Z be singleton variables. Then

$$\text{CR}(X ; Y ; Z) = \frac{P(X, Y, Z)}{P(X)P(Y)P(Z)}$$

is distinguished from

$$\text{CR}(X ; Y, Z) = \frac{P(X, Y, Z)}{P(X)P(Y, Z)}$$

The first CR is between three singleton variables. In contrast, the second CR is between two variables: a singleton variable X and a joint variable (Y, Z) .

Let $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N\}$ and $\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, \dots, \mathcal{Y}_M\}$. In a CR, we denote the string $\mathcal{X}_1 ; \mathcal{X}_2 ; \dots ; \mathcal{X}_N$ by $\text{Sem } \mathcal{X}$. In contrast, the string $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N$ is just denoted by $\mathcal{X}$. That is

$$\begin{aligned}\text{Sem } \mathcal{X} &:= \mathcal{X}_1 ; \mathcal{X}_2 ; \dots ; \mathcal{X}_N \\ \mathcal{X} &:= \mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N\end{aligned}$$

For example,

$$\begin{aligned}\text{CR}(\text{Sem } \mathcal{X}) &= \text{CR}(\mathcal{X}_1 ; \mathcal{X}_2 ; \dots ; \mathcal{X}_N) = \frac{P(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N)}{P(\mathcal{X}_1)P(\mathcal{X}_2)\dots P(\mathcal{X}_N)} \\ \text{CR}(\mathcal{X} ; \mathcal{Y}) &= \text{CR}(\mathcal{X}_1, \dots, \mathcal{X}_N ; \mathcal{Y}_1, \dots, \mathcal{Y}_M) = \frac{P(\mathcal{X}_1, \dots, \mathcal{X}_N, \mathcal{Y}_1, \dots, \mathcal{Y}_M)}{P(\mathcal{X}_1, \dots, \mathcal{X}_N)P(\mathcal{Y}_1, \dots, \mathcal{Y}_M)}\end{aligned}$$

Note that $\text{Sem } \mathcal{X}$ will never appear in probability P . Only $\mathcal{X}$ can appear in P :

$$P(\mathcal{X}) = P(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N)$$

Definition 27 (Conditional Co-occurrence Rate). *Let $\mathcal{X}$ and $\mathcal{Y}$ be sets of random variables. The co-occurrence rate of $\mathcal{X}$ conditioned by $\mathcal{Y}$ is defined as:*

$$\text{CR}(\mathcal{X}_1 ; \mathcal{X}_2 ; \dots ; \mathcal{X}_N \mid \mathcal{Y}) = \frac{P(\mathcal{X}_1 ; \mathcal{X}_2 ; \dots ; \mathcal{X}_N \mid \mathcal{Y})}{P(\mathcal{X}_1 \mid \mathcal{Y})P(\mathcal{X}_2 \mid \mathcal{Y})\dots P(\mathcal{X}_N \mid \mathcal{Y})}$$

Definition 28 (Co-occurrence Rate Independence). *Let $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$ be sets of random variables. If*

$$\text{CR}(\mathcal{X} ; \mathcal{Y} \mid \mathcal{Z}) = \text{CR}(\mathcal{X} ; \mathcal{Y}),$$

we say the co-occurrence rate between $\mathcal{X}$ and $\mathcal{Y}$ is independent of $\mathcal{Z}$, which is denoted by $(\mathcal{X} ; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z}$.

3.1.2 Properties of Co-occurrence Rate

Co-occurrence rate has some simple and nice properties. That is operations on a joint CR can be perfectly mapped to operations on the corresponding sub-graph. We describe them as follows.

Theorem 10 (Partition Operation). *Let $\mathcal{X}$ be a set of random variables. $\mathcal{X}_A \subseteq \mathcal{X}$ and $\mathcal{X}_B \subseteq \mathcal{X}$ form a partition of $\mathcal{X}$. That is $\mathcal{X}_A \cap \mathcal{X}_B = \emptyset$ and $\mathcal{X}_A \cup \mathcal{X}_B = \mathcal{X}$. Then,*

$$\text{CR}(\text{Sem } \mathcal{X}) = \text{CR}(\text{Sem } \mathcal{X}_A)\text{CR}(\text{Sem } \mathcal{X}_B)\text{CR}(\mathcal{X}_A ; \mathcal{X}_B)$$

This theorem tells the original CR can be decomposed into three parts: part $\mathcal{A}$, part $\mathcal{B}$ and the cut between part $\mathcal{A}$ and part $\mathcal{B}$. The proof follows directly from the definition of co-occurrence rate (Definition 26).

Proof.

$$\begin{aligned}
 & \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{A}}) \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{B}}) \text{CR}(\mathcal{X}_{\mathcal{A}} ; \mathcal{X}_{\mathcal{B}}) \\
 &= \frac{P(\mathcal{X}_{\mathcal{A}})}{\prod_{\mathcal{X}_a \in \mathcal{X}_{\mathcal{A}}} P(\mathcal{X}_a)} \times \frac{P(\mathcal{X}_{\mathcal{B}})}{\prod_{\mathcal{X}_b \in \mathcal{X}_{\mathcal{B}}} P(\mathcal{X}_b)} \times \frac{P(\mathcal{X})}{P(\mathcal{X}_{\mathcal{A}})P(\mathcal{X}_{\mathcal{B}})} \\
 &= \frac{P(\mathcal{X})}{\prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i)} \\
 &= \text{CR}(\text{Sem } \mathcal{X})
 \end{aligned}$$

□

In a similar way, we can prove that this theorem also holds for conditional CR (Definition 27):

Proof.

$$\begin{aligned}
 & \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{A}} \mid \mathcal{Y}) \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{B}} \mid \mathcal{Y}) \text{CR}(\mathcal{X}_{\mathcal{A}} ; \mathcal{X}_{\mathcal{B}} \mid \mathcal{Y}) \\
 &= \frac{P(\mathcal{X}_{\mathcal{A}} \mid \mathcal{Y})}{\prod_{\mathcal{X}_a \in \mathcal{X}_{\mathcal{A}}} P(\mathcal{X}_a \mid \mathcal{Y})} \times \frac{P(\mathcal{X}_{\mathcal{B}} \mid \mathcal{Y})}{\prod_{\mathcal{X}_b \in \mathcal{X}_{\mathcal{B}}} P(\mathcal{X}_b \mid \mathcal{Y})} \times \frac{P(\mathcal{X} \mid \mathcal{Y})}{P(\mathcal{X}_{\mathcal{A} \mid \mathcal{Y}})P(\mathcal{X}_{\mathcal{B}} \mid \mathcal{Y})} \\
 &= \frac{P(\mathcal{X} \mid \mathcal{Y})}{\prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i \mid \mathcal{Y})} \\
 &= \text{CR}(\text{Sem } \mathcal{X} \mid \mathcal{Y})
 \end{aligned}$$

□

This theorem also holds for singleton CR (Definition 24) and empty CR (Definition 25):

Proof.

$$\begin{aligned}
 \text{CR}(X) &= \text{CR}(X ; \emptyset) = \text{CR}(X) \text{CR}(\emptyset) \text{CR}(X ; \emptyset) = 1 \\
 \text{CR}(\emptyset) &= \text{CR}(\emptyset ; \emptyset) = \text{CR}(\emptyset) \text{CR}(\emptyset) \text{CR}(\emptyset ; \emptyset) = 1
 \end{aligned}$$

□

Remark The partition operation decomposes a joint CR into three factors: $\text{CR}(\text{Sem } \mathcal{X}_A)$, $\text{CR}(\text{Sem } \mathcal{X}_B)$ and $\text{CR}(\mathcal{X}_A ; \mathcal{X}_B)$. But we can see the scope of the last factor $\text{CR}(\mathcal{X}_A ; \mathcal{X}_B)$ is still $\mathcal{X}_A \cup \mathcal{X}_B = \mathcal{X}$. We will see later the scope can be reduced using independence relations (Equation 3.2). The partition operation can be applied repeatedly to further decompose $\text{CR}(\text{Sem } \mathcal{X}_A)$ and $\text{CR}(\text{Sem } \mathcal{X}_B)$.

Theorem 11 (Merge Operation). *Let $\mathcal{X}$ be a set of variables and $\mathcal{X}_A \subseteq \mathcal{X}$. Let $\mathcal{X}_U = \mathcal{X} - \mathcal{X}_A$. Then,*

$$\text{CR}(\text{Sem } \mathcal{X}) = \text{CR}(\text{Sem } \mathcal{X}_U ; \mathcal{X}_A) \text{CR}(\text{Sem } \mathcal{X}_A)$$

In this theorem, the variables in $\mathcal{X}_A$ are merged together to be a joint variable, and a new factor $\text{CR}(\text{Sem } \mathcal{X}_A)$ is generated. The merge operation decomposes a joint CR by merging a graph from down to top. In contrast, the partition operation (Theorem 10) decomposes a joint CR by partitioning a graph from top to down. This operation can be repeatedly used to further merge variables in $\mathcal{U}$. This theorem can be proved as follows.

Proof.

$$\begin{aligned} & \text{CR}(\text{Sem } \mathcal{X}_U ; \mathcal{X}_A) \text{CR}(\text{Sem } \mathcal{X}_A) \\ &= \frac{P(\mathcal{X})}{[\prod_{\mathcal{X}_i \in \mathcal{X}_U} P(\mathcal{X}_i)] P(\mathcal{X}_A)} \times \frac{P(\mathcal{X}_A)}{\prod_{\mathcal{X}_a \in \mathcal{X}_A} P(\mathcal{X}_a)} \\ &= \text{CR}(\text{Sem } \mathcal{X}) \end{aligned}$$

□

This theorem also holds for conditional CR (Definition 27), singleton CR (Definition 24) and empty CR (Definition 25).

Theorem 12 (Duplication Operation). *Let $\mathcal{X}$ be a non-empty set of variables and $\mathcal{X}_a \in \mathcal{X}$. Then,*

$$\text{CR}(\text{Sem } \mathcal{X}) = \text{CR}(\text{Sem } \mathcal{X} ; \mathcal{X}_a) P(\mathcal{X}_a)$$

Proof.

$$\begin{aligned} & \text{CR}(\text{Sem } \mathcal{X} ; \mathcal{X}_a) P(\mathcal{X}_a) \\ &= \frac{P(\mathcal{X}, \mathcal{X}_a)}{[\prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i)] P(\mathcal{X}_a)} \times P(\mathcal{X}_a) \\ &= \frac{P(\mathcal{X})}{[\prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i)] P(\mathcal{X}_a)} \times P(\mathcal{X}_a) \\ &= \text{CR}(\text{Sem } \mathcal{X}) \end{aligned}$$

The second equation is obtained because $P(\mathcal{X}, \mathcal{X}_a) = P(\mathcal{X})$, where $\mathcal{X}_a \in \mathcal{X}$. This equation holds because the logic operation ‘and’ ($\wedge$) is absorptive. $\square$

Remark In this theorem, $\mathcal{X}_a$ is duplicated. This theorem is very useful in dealing with *overlapped* sub-graphs (Section 3.2.2). This theorem also holds for conditional CR (Definition 27) and singleton CR (Definition 24). If we repeatedly use this operation, we can duplicate a set of variables.

Theorem 13 (Elimination Operations). *Let $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$ are disjoint sets of variables, and $(\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z})$. That is $\mathcal{X}$ is independent of $\mathcal{Y}$ given $\mathcal{Z}$. Then,*

$$\text{CR}(\mathcal{X}; \mathcal{Y}, \mathcal{Z}) = \text{CR}(\mathcal{X}; \mathcal{Z}) \quad (3.2)$$

$$\text{CR}(\mathcal{X}, \mathcal{Z}; \mathcal{Y}, \mathcal{Z}) = \text{CR}(\mathcal{Z}; \mathcal{Z}) = \frac{1}{P(\mathcal{Z})} \quad (3.3)$$

$$\text{CR}(\mathcal{X}, \mathcal{Y}; \mathcal{Z}) = \frac{\text{CR}(\mathcal{X}; \mathcal{Z})\text{CR}(\mathcal{Y}; \mathcal{Z})}{\text{CR}(\mathcal{X}; \mathcal{Y})} \quad (3.4)$$

Remark These operations can be used to reduce variables in a CR. We can prove these theorems as follows.

Proof.

$$\begin{aligned} (\mathcal{X} \perp\!\!\!\perp \mathcal{Y} \mid \mathcal{Z}) &\Rightarrow P(\mathcal{X}, \mathcal{Y} \mid \mathcal{Z}) = P(\mathcal{X} \mid \mathcal{Z})P(\mathcal{Y} \mid \mathcal{Z}) \\ &\Rightarrow P(\mathcal{X}, \mathcal{Y}, \mathcal{Z}) = P(\mathcal{X}, \mathcal{Z})P(\mathcal{Y}, \mathcal{Z})/P(\mathcal{Z}) \end{aligned}$$

$$\text{CR}(\mathcal{X}; \mathcal{Y}, \mathcal{Z}) = \frac{P(\mathcal{X}, \mathcal{Y}, \mathcal{Z})}{P(\mathcal{X})P(\mathcal{Y}, \mathcal{Z})} = \text{CR}(\mathcal{X}; \mathcal{Z})$$

$$\text{CR}(\mathcal{X}, \mathcal{Z}; \mathcal{Y}, \mathcal{Z}) = \frac{P(\mathcal{X}, \mathcal{Y}, \mathcal{Z})}{P(\mathcal{X}, \mathcal{Z})P(\mathcal{Y}, \mathcal{Z})} = \frac{1}{P(\mathcal{Z})} = \text{CR}(\mathcal{Z}; \mathcal{Z})$$

$$\text{CR}(\mathcal{X}, \mathcal{Y}; \mathcal{Z}) = \frac{P(\mathcal{X}, \mathcal{Y}, \mathcal{Z})}{P(\mathcal{X}, \mathcal{Y})P(\mathcal{Z})} = \frac{\text{CR}(\mathcal{X}; \mathcal{Z})\text{CR}(\mathcal{Y}; \mathcal{Z})}{\text{CR}(\mathcal{X}; \mathcal{Y})}$$

$\square$

Theorem 14 (Expectation Operation). *Let $\mathcal{X}$ be a set of variables, and $\mathcal{X}_a \in \mathcal{X}$. Let $\mathcal{X}_{\mathcal{U}} = \mathcal{X} - \{\mathcal{X}_a\}$. Then,*

$$\sum_{\mathcal{X}_a} [\text{CR}(\text{Sem } \mathcal{X})P(\mathcal{X}_a)] = \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{U}})$$

In other words, $\text{CR}(\text{Sem } \mathcal{X}_{\mathcal{U}}) = E_{\mathcal{X}_a}[\text{CR}(\text{Sem } \mathcal{X})]$. E is the expectation.

Proof.

$$\begin{aligned} \sum_{\mathcal{X}_a} [\text{CR}(\text{Sem } \mathcal{X}) P(\mathcal{X}_a)] &= \sum_{\mathcal{X}_a} \left[\frac{P(\mathcal{X})}{\prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i)} P(\mathcal{X}_a) \right] \\ &= \frac{P(\mathcal{X}_{\mathcal{U}})}{\prod_{\mathcal{X}_j \in \mathcal{X}_{\mathcal{U}}} P(\mathcal{X}_j)} = \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{U}}) \end{aligned}$$

□

Remark Following this theorem, we can eliminate a variable $\mathcal{X}_a$ in CR by applying the expectation operation $E_{\mathcal{X}_a}$ to the CR.

Remark This theorem can be used to check if two CR factors share information. For example, $\text{CR}(Z ; X, Y)$ and $\text{CR}(X ; Y)$ do not share information. Because $E_Z[\text{CR}(Z ; X, Y)] = \text{CR}(X, Y) = 1$. $\text{CR}(Z ; X, Y)$ does not contain any dependence information between X and Y .

Following this theorem, we can eliminate a variable $\mathcal{X}_a$ in CR by applying the expectation operation $E_{\mathcal{X}_a}$ to the CR.

Theorem 15 (Decomposition of Probability Independence).

$$((\mathcal{X}, \mathcal{Y}) \perp\!\!\!\perp \mathcal{Z}) \Leftrightarrow ((\mathcal{X} ; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z}) \wedge (\mathcal{X} \perp\!\!\!\perp \mathcal{Z}) \wedge (\mathcal{Y} \perp\!\!\!\perp \mathcal{Z})$$

This theorem states that the probability independency $((\mathcal{X}, \mathcal{Y}) \perp\!\!\!\perp \mathcal{Z})$ can be decomposed into three parts: (i) $((\mathcal{X} ; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z})$, (ii) $(\mathcal{X} \perp\!\!\!\perp \mathcal{Z})$ and (iii) $(\mathcal{Y} \perp\!\!\!\perp \mathcal{Z})$. The first one is a co-occurrence rate independency (Definition 28).

Proof. $(\Rightarrow)$. Because $((\mathcal{X}, \mathcal{Y}) \perp\!\!\!\perp \mathcal{Z})$, we have $(\mathcal{X} \perp\!\!\!\perp \mathcal{Z})$ and $(\mathcal{Y} \perp\!\!\!\perp \mathcal{Z})$ ³. Hence,

$$\text{CR}(\mathcal{X} ; \mathcal{Y} | \mathcal{Z}) = \frac{P(\mathcal{X}, \mathcal{Y} | \mathcal{Z})}{P(\mathcal{X} | \mathcal{Z})P(\mathcal{Y} | \mathcal{Z})} = \frac{P(\mathcal{X}, \mathcal{Y})}{P(\mathcal{X})P(\mathcal{Y})} = \text{CR}(\mathcal{X}, \mathcal{Y}).$$

According to Definition 28, we have $((\mathcal{X} ; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z})$.

$(\Leftarrow)$. As $((\mathcal{X} ; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z})$, we have

$$\frac{P(\mathcal{X}, \mathcal{Y} | \mathcal{Z})}{P(\mathcal{X} | \mathcal{Z})P(\mathcal{Y} | \mathcal{Z})} = \frac{P(\mathcal{X}, \mathcal{Y})}{P(\mathcal{X})P(\mathcal{Y})}$$

³By a marginalization operation, we can directly get this.

As $(\mathcal{X} \perp\!\!\!\perp \mathcal{Z})$ and $(\mathcal{Y} \perp\!\!\!\perp \mathcal{Z})$, we have $P(\mathcal{X} | \mathcal{Z}) = P(\mathcal{X})$ and $P(\mathcal{Y} | \mathcal{Z}) = P(\mathcal{Y})$. Hence, $P(\mathcal{X}, \mathcal{Y} | \mathcal{Z}) = P(\mathcal{X}, \mathcal{Y}) \Rightarrow ((\mathcal{X}, \mathcal{Y}) \perp\!\!\!\perp \mathcal{Z})$. $\square$

Theorem 16 (Triangle Decomposition). *If $(\mathcal{X}; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z}$, then:*

$$\text{CR}(\mathcal{X}; \mathcal{Y}; \mathcal{Z}) = \text{CR}(\mathcal{X}; \mathcal{Y})\text{CR}(\mathcal{Y}; \mathcal{Z})\text{CR}(\mathcal{Z}; \mathcal{X})$$

Proof. As $(\mathcal{X}; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z}$, we have

$$\begin{aligned} \text{CR}(\mathcal{X}; \mathcal{Y} | \mathcal{Z}) &= \text{CR}(\mathcal{X}; \mathcal{Y}) \\ \Leftrightarrow \frac{P(\mathcal{X}, \mathcal{Y} | \mathcal{Z})}{P(\mathcal{X} | \mathcal{Z})P(\mathcal{Y} | \mathcal{Z})} &= \frac{P(\mathcal{X}, \mathcal{Y})}{P(\mathcal{X})P(\mathcal{Y})} \\ \Leftrightarrow \frac{P(\mathcal{X}, \mathcal{Y}, \mathcal{Z})}{P(\mathcal{X})P(\mathcal{Y})P(\mathcal{Z})} &= \frac{P(\mathcal{X}, \mathcal{Y})P(\mathcal{X}, \mathcal{Z})P(\mathcal{Y}, \mathcal{Z})}{P^2(\mathcal{X})P^2(\mathcal{Y})P^2(\mathcal{Z})} \\ \Leftrightarrow \text{CR}(\mathcal{X}; \mathcal{Y}; \mathcal{Z}) &= \text{CR}(\mathcal{X}; \mathcal{Y})\text{CR}(\mathcal{Y}; \mathcal{Z})\text{CR}(\mathcal{Z}; \mathcal{X}) \end{aligned}$$

$\square$

Remark There is a common abuse of the Hammersley–Clifford theorem (HCT) in pairwise Markov networks in which each edge is associated with a factor. For example, in fact on a triangle graph $G = \{A, B, C\}$, we cannot obtain $P(A, B, C) = \phi_1(A, B)\phi_2(B, C)\phi_3(C, A)$ by HCT. For a triangle graph, the maximum clique is the triangle. Each edge and each node forms a sub-clique. From the proof of HCT, we can see that the factors over a sub-clique can be absorbed into a maximum clique which contains the sub-clique, but not in the other direction. That is following HCT a correct factorization must contain factors over maximum cliques. Factors over sub-cliques are optional because they can be absorbed by maximum cliques. As examples, the following factorizations are correct according to HCT:

$$\begin{aligned} P(A, B, C) &= \phi(A, B, C)\phi_1(A, B)\phi_2(B, C)\phi_3(C, A)\phi_4(A)\phi_5(B)\phi_6(C) \\ P(A, B, C) &= \phi(A, B, C) \\ P(A, B, C) &= \phi(A, B, C)\phi_4(A)\phi_5(B)\phi_6(C) \\ &\dots \end{aligned}$$

But not

$$\begin{aligned} P(A, B, C) &= \phi_1(A, B)\phi_2(B, C)\phi_3(C, A) \\ P(A, B, C) &= \phi_1(A, B)\phi_2(B, C)\phi_3(C, A)\phi_4(A)\phi_5(B)\phi_6(C) \\ &\dots \end{aligned}$$

Theorem 16 provides the sufficient and necessary condition, $(\mathcal{X} ; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z}$, to further decompose a triangle graph into pairwise factors. Without this condition, we cannot do pairwise decomposition to a triangle. The co-occurrence rate independency $(\mathcal{X} ; \mathcal{Y}) \perp\!\!\!\perp_{\text{CR}} \mathcal{Z}$ does not imply any conditional probability independency. In other words, the triangle is still a triangle. We do not need to change the graph structure.

Theorem 17 (Unconnected Nodes Theorem). *Let $\mathcal{X}$ be a set of variables and $\mathbf{x}$ be an assignment of $\mathcal{X}$. Let $\mathcal{X}_a, \mathcal{X}_b \in \mathcal{X}$, and $\mathcal{X}_{\mathcal{U}} \subseteq \mathcal{X} - \{\mathcal{X}_a, \mathcal{X}_b\}$. Suppose $\mathcal{X}_a$ and $\mathcal{X}_b$ are not directly connected. In other words, $(\mathcal{X}_a \perp\!\!\!\perp \mathcal{X}_b \mid \mathcal{X}_{\mathcal{U}})$. Then,*

$$\begin{aligned} & \text{CR}(\mathcal{X}_a = \mathbf{x}_a ; \mathcal{X}_b = \mathbf{x}_b ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_a ; \mathcal{X}_b ; \mathcal{X}_{\mathcal{U}}) \\ &= \text{CR}(\mathcal{X}_a = \mathbf{x}_a ; \mathcal{X}_b ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_a ; \mathcal{X}_b = \mathbf{x}_b ; \mathcal{X}_{\mathcal{U}}) \end{aligned}$$

Proof. On the left hand side, to both terms, we apply the partition operation (Theorem 10) to partition $\mathcal{X}_a$ out and then apply elimination operation (Equation 3.2). We obtain:

$$\text{CR}(\mathcal{X}_a = \mathbf{x}_a ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_b = \mathbf{x}_b ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_a ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_b ; \mathcal{X}_{\mathcal{U}})$$

On the right hand side, to both terms we also apply the partition operation (Theorem 10) to partition $\mathcal{X}_a$ out and then apply elimination operation (Equation 3.2). We obtain:

$$\text{CR}(\mathcal{X}_a = \mathbf{x}_a ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_b ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_a ; \mathcal{X}_{\mathcal{U}}) \times \text{CR}(\mathcal{X}_b = \mathbf{x}_b ; \mathcal{X}_{\mathcal{U}})$$

Hence, the left hand side is equal to the right hand side. $\square$

Remark This theorem is the key in the proof of Hammersley–Clifford theorem using co-occurrence rate (Appendix A.3). Using this theorem, the factors grouped into non-cliques cancel out.

3.2 Examples

In this section, we give examples to illustrate how to obtain a CRN factorization for different kinds of graphs. Also see Section 4.3 for a discussion of consistency between factors.

Figure 3.1: A Tree-structured Graph

3.2.1 Example 1: Tree-structured Graphs

Figure 3.1 is a tree-structured graph. Tree-structured graphs are the simplest type of graphs for factorization. Co-occurrence rate networks have the same independence semantics with Markov networks. That is if $\mathcal{Y}$ separates $\mathcal{X}$ and $\mathcal{Z}$, then $(\mathcal{X} \perp\!\!\!\perp \mathcal{Z} \mid \mathcal{Y})$. For this graph, we have the following independencies: $(X \perp\!\!\!\perp Z \mid Y)$, $(X \perp\!\!\!\perp U \mid Y)$, $(Z \perp\!\!\!\perp U \mid Y)$, $(U \perp\!\!\!\perp X, Z \mid Y)$, $(X \perp\!\!\!\perp U, Z \mid Y)$ and $(Z \perp\!\!\!\perp X, U \mid Y)$. We can decompose the joint probability over this CRN as follows.

First, according to Definition 26, we decouple the joint probability into a joint CR and singleton probabilities as follows:

$$P(X, Y, Z, U) = \text{CR}(X ; Y ; Z ; U) P(X) P(Y) P(Z) P(U) \quad (3.5)$$

We go on to decompose the joint $\text{CR}(X ; Y ; Z ; U)$ as follows:

$$\begin{aligned} \text{CR}(X ; Y ; Z ; U) \\ = \text{CR}(X) \text{CR}(Y ; Z ; U) \text{CR}(X ; Y, Z, U) \end{aligned} \quad (3.6)$$

$$= \text{CR}(Y ; Z ; U) \text{CR}(X ; Y) \quad (3.7)$$

$$= \text{CR}(U) \text{CR}(Y ; Z) \text{CR}(U ; Y, Z) \text{CR}(X ; Y) \quad (3.8)$$

$$= \text{CR}(Y ; Z) \text{CR}(U ; Y) \text{CR}(X ; Y) \quad (3.9)$$

We obtain Equation 3.6 by partitioning X out (Theorem 10). Equation 3.7 is obtained from Equation 3.6 because the singleton $\text{CR}(X) = 1$ (Definition 24), and as $(X \perp\!\!\!\perp Z, U \mid Y)$, we have $\text{CR}(X ; Y, Z, U) = \text{CR}(X ; Y)$ by Equation 3.2. Equation 3.8 is obtained by partitioning U out from $\text{CR}(Y ; Z ; U)$.

Finally, plug Equation 3.9 into Equation 3.5, we obtain the factorization of the joint probability as follows:

$$P(X, Y, Z, U) = \text{CR}(Y ; Z) \text{CR}(U ; Y) \text{CR}(X ; Y) P(X) P(Y) P(Z) P(U) \quad (3.10)$$

Remark An important property of Equation 3.10 is that the CR factors, i.e., $\text{CR}(Y; Z)$, $\text{CR}(U; Y)$, and $\text{CR}(X; Y)$, do not share information directly. In other words, they are independent of each other conditioned by the singleton probabilities. For example, $\text{CR}(Y; Z)$ and $\text{CR}(U; Y)$ do not share any information. This can be seen as follows. Suppose $\text{CR}(Y; Z)$ and $\text{CR}(U; Y)$ share some information, this can only happen on Y . But if we eliminate Z and U by the expectation operation (Theorem 14), we have $E_Z[\text{CR}(Y; Z)] = \text{CR}(Y) = 1$ and $E_U[\text{CR}(U; Y)] = \text{CR}(Y) = 1$. That is we do not need to calibrate information between $\text{CR}(Y; Z)$ and $\text{CR}(U; Y)$. The independencies between all factors can be represented by a Bayesian network as depicted in Figure 3.2.

Figure 3.2: A Bayesian Network of Factors in Equation 3.10

This is different from the tree reparameterization which is given as follows:

$$P(X, Y, Z, U) = \frac{P(Y, Z)P(U, Y)P(X, Y)}{P^2(Y)}$$

In this factorization, the pairwise marginals share common information. For example, $P(Y, Z)$ and $P(U, Y)$ share the common information of $P(Y)$. Hence we can not train $P(Y, Z)$ and $P(U, Y)$ separately models as separate training can break consistency between them. To avoid the shared information contained in singleton probabilities, we always strip off singleton probabilities from the joint probability as illustrated in Equation 3.5.

Remark On tree graphs, Equation 3.10 is identical to the hypertree factorization⁴ described in Wainwright et al. [6] even though the underlying motivations are different. When moving to more complex graphs, the specificity of the CRN factorization from the hypertree factorization becomes obvious. We will illustrate this in the following examples (Figure 3.6).

⁴Here we consider a normal tree graph as a special hypertree.

Remark Using chain-rule of probability, we can finally obtain the same result for tree-structured graphs [6]. But by chain-rule we use conditional probabilities, which is a asymmetric notion, in intermediate steps. That is we need to select a direction. By CR, we always keep the symmetry throughout and avoid to select a direction.

3.2.2 Example 2: A Acyclic Clique Graph

Figure 3.3: A Acyclic Clique Graph

Figure 3.3 depicts a acyclic clique graph. There are three maximum cliques (complete sub-graphs) in this graph: $\mathcal{C}_1 = \{X, Y, Z\}$, $\mathcal{C}_2 = \{Z, U\}$, and $\mathcal{C}_3 = \{U, V, W\}$. Its clique graph is shown in Figure 3.4.

Figure 3.4: The Clique Graph of Figure 3.3

In the clique graph (Figure 3.4), nodes represent cliques in the original graph, and edges are labeled with overlapped nodes between two cliques. The clique graph in Figure 3.4 is *acyclic*. That is there is no loop in the graph. Tree structured graphs discussed in the previous section are a special case of acyclic clique graphs. Note that even though the original graph contains loops, it may still be a acyclic clique graph. Acyclic or not is judged based on its clique graph rather than the original graph.

The joint probability over this graph can be decomposed as follows. First, we decouple the joint probability into a joint CR and singleton probabilities

according to Definition 26:

$$P(X, Y, Z, U, V, W) = \text{CR}(X; Y; Z; U; V; W)P(X)P(Y)P(Z)P(U)P(V)P(W) \quad (3.11)$$

We go on to decompose the joint CR as follows:

$$\begin{aligned} & \text{CR}(X; Y; Z; U; V; W) \\ &= \text{CR}(X; Y; Z; Z; U; V; W)P(Z) \end{aligned} \quad (3.12)$$

$$= \text{CR}(X; Y; Z)\text{CR}(Z; U; V; W)\text{CR}(X, Y, Z; Z, U, V, W)P(Z) \quad (3.13)$$

$$= \text{CR}(X; Y; Z)\text{CR}(Z; U; V; W)\text{CR}(Z; Z)P(Z) \quad (3.14)$$

$$= \text{CR}(X; Y; Z)\text{CR}(Z; U; V; W) \quad (3.15)$$

$$= \text{CR}(X; Y; Z)\text{CR}(Z; U; U; V; W)P(U) \quad (3.16)$$

$$= \text{CR}(X; Y; Z)\text{CR}(Z; U)\text{CR}(U; V; W)\text{CR}(Z, U; U, V, W)P(U) \quad (3.17)$$

$$= \text{CR}(X; Y; Z)\text{CR}(Z; U)\text{CR}(U; V; W) \quad (3.18)$$

In Equation 3.12, we duplicate the overlapped variable Z between $\mathcal{C}_1$ and $\mathcal{C}_2$ by Theorem 12. Equation 3.13 is obtained from Equation 3.12 by partitioning $\{X, Y, Z\}$ out using Theorem 10. Equation 3.14 is obtained as follows. Since $(X, Y \perp\!\!\!\perp U, V, W \mid Z)$, we have $\text{CR}(X, Y, Z; Z, U, V, W) = \text{CR}(Z; Z) = \frac{1}{P(Z)}$ by Equation 3.3. In Equation 3.16, we duplicate the overlapped variable U between $\mathcal{C}_2$ and $\mathcal{C}_3$. We go on this process to obtain the final result.

Plug Equation 3.18 into Equation 3.11, we have:

$$\begin{aligned} P(X, Y, Z, U, V, W) &= \text{CR}(X; Y; Z)\text{CR}(Z; U)\text{CR}(U; V; W) \\ &\quad \times P(X)P(Y)P(Z)P(U)P(V)P(W) \end{aligned} \quad (3.19)$$

In this factorization, CR factors do not share information conditioned by singleton probabilities. The independencies between all factors can be represented by a Bayesian network as depicted in Figure 3.5.

3.2.3 Example 3: Another Cyclic Clique Graph

Figure 3.6 depicts another acyclic clique graph. The difference between Figure 3.6 and Figure 3.3 is that there are more than one variables in the separator set $\{X, Y\}$ in Figure 3.6. But there is only one variable in the separator sets $\{Z\}$, $\{U\}$ in Figure 3.3. This leads to a slightly different treatment to obtain the CRN factorization as we will see in this example.

Figure 3.5: A Bayesian Network of Factors in Equation 3.19

Figure 3.6: Another Acyclic Clique Graph

Similar to the previous example, we can obtain the following factorization.

$$\begin{aligned} \text{CR}(U ; X ; Y ; Z) \\ = \text{CR}(U ; X ; Y ; X ; Y ; Z)P(X)P(Y) \end{aligned} \quad (3.20)$$

$$= \text{CR}(U ; X ; Y)\text{CR}(X ; Y ; Z)\text{CR}(U, X, Y ; X, Y, Z)P(X)P(Y) \quad (3.21)$$

$$= \text{CR}(U ; X ; Y)\text{CR}(X ; Y ; Z)\text{CR}(X, Y ; X, Y)P(X)P(Y) \quad (3.22)$$

$$= \text{CR}(U ; X ; Y)\text{CR}(X ; Y ; Z) \frac{P(X, Y)}{P(X, Y)P(X, Y)} P(X)P(Y) \quad (3.23)$$

$$= \text{CR}(U ; X ; Y)\text{CR}(X ; Y ; Z)/\text{CR}(X ; Y) \quad (3.24)$$

Equation 3.20 is obtained by duplicating X and Y (Theorem 12). Equation 3.21 is obtained by partitioning U, X, Y out (Theorem 10). Because $(U \perp\!\!\!\perp Z \mid X, Y)$, following Theorem 13 (Equation 3.3) we obtain Equation 3.22. Hence, we obtain the following factorization for the graph shown in Figure 3.6:

$$P(X, Y, Z, U) = \frac{\text{CR}(U ; X ; Y)\text{CR}(Z ; X ; Y)}{\text{CR}(X ; Y)} P(X)P(Y)P(Z)P(U) \quad (3.25)$$

Figure 3.7: The Clique Graph of Figure 3.6

In contrast to the previous example, we cannot stop here, because $\text{CR}(U ; X ; Y)$ and $\text{CR}(Z ; X ; Y)$ share the common information of $\text{CR}(X ; Y)$. This can be seen by applying the expectation operation (Theorem 14): $E_U[\text{CR}(U ; X ; Y)] = E_Z[\text{CR}(Z ; X ; Y)] = \text{CR}(X ; Y)$. This kind of shared information can be stripped out by the merge operation (Theorem 11) as follows:

$$\begin{aligned}
 P(X, Y, Z, U) &= \frac{\text{CR}(U ; X ; Y)\text{CR}(Z ; X ; Y)}{\text{CR}(X ; Y)} P(X)P(Y)P(Z)P(U) \\
 &= \frac{\text{CR}(U ; X, Y)\text{CR}(X ; Y)\text{CR}(Z ; X, Y)\text{CR}(X ; Y)}{\text{CR}(X ; Y)} P(X)P(Y)P(Z)P(U) \\
 &= \text{CR}(U ; X, Y)\text{CR}(Z ; X, Y)\text{CR}(X ; Y)P(X)P(Y)P(Z)P(U)
 \end{aligned} \tag{3.26}$$

Remark We can see that $\text{CR}(Z ; X, Y)$ does not share information with $\text{CR}(X ; Y)$ because by the expectation operation (Theorem 14), we have: $E_Z[\text{CR}(Z ; X, Y)] = \text{CR}(X, Y) = 1$. That is we do not need to do calibration between $\text{CR}(Z ; X, Y)$ and $\text{CR}(X ; Y)$. The independencies of factors in Equation 3.26 can be represented with the Bayesian network depicted in Figure 3.8.

Figure 3.8: A Bayesian Network of Factors in Equation 3.26

Note that for the following constraint:

$$\sum_{XYZ} \text{CR}(Z; X, Y) P(Z) P(X, Y) = 1$$

we only need to guarantee

$$\sum_Z \text{CR}(Z; X, Y) P(Z) = 1$$

because

$$\sum_{XYZ} \text{CR}(Z; X, Y) P(Z) P(X, Y) = \sum_{XY} P(XY) \sum_Z \text{CR}(Z; X, Y) P(Z).$$

Here $\sum_{XY} P(XY) = 1$ has already been guaranteed by the constraint: $\sum_{XY} \text{CR}(X; Y) P(X) P(Y) = 1$. Hence, the factor $\text{CR}(Z; X, Y)$ only depends on $P(Z)$. This also applies to the factors $\text{CR}(U; X, Y)$ and $P(U)$.

3.2.4 Example 4: Cyclic Clique Graphs

Figure 3.9 shows a cyclic clique graph, whose clique graph depicted in Figure 3.10 is *cyclic*.

Figure 3.9: A Cyclic Clique Graph

Unfortunately, for cyclic clique graphs, the factorization process used for acyclic clique graphs does not work. This is because the shared variables between two cliques do not completely separate a clique from the rest part of the graph. For example, in Figure 3.10, X can not separate $\{U, X\}$ and $\{X, Z\}$. Because there is a second path $[U, Y, Z]$ connecting these two cliques. Hence we cannot obtain Equation 3.14 or Equation 3.22. For a cyclic clique graph, we first transform it into an acyclic clique graph as follow.

Figure 3.10: The Clique Graph of Figure 3.9

Triangulation A general graph can be transformed to a junction tree by triangulation. This idea can be implemented within the framework of co-occurrence rate networks as follows. First, we partition X out (Theorem 10), and eliminate Y using $(X \perp\!\!\!\perp Y \mid U, Z)$ (Equation 3.2 of Theorem 13):

$$\text{CR}(X ; U ; Y ; Z) = \text{CR}(X ; U, Z) \text{CR}(U ; Y ; Z)$$

Figure 3.11: The Local Graph $\{U, Y, Z\}$ of Figure 3.9

We continue to examine $\text{CR}(U ; Y ; Z)$. $\{U, Y, Z\}$ forms a local graph as shown in Figure 3.11. In Figure 3.9, there is a second path $[U, X, Z]$ connecting U and Z . Hence, U is *not* independent of Z given Y . Therefore, we need to add a new edge between U and Z in Figure 3.11. This makes $\{U, Y, Z\}$ a triangle. We obtain the factorization as follows:

$$P(X, U, Y, Z) = \text{CR}(X ; U, Z) \text{CR}(U ; Y ; Z) P(X) P(U) P(Y) P(Z) \quad (3.27)$$

Note that in this equation, CR factors are independent of each other given singleton probabilities. This can be seen by the expectation operation (Theorem 14) as follows: $E_X[\text{CR}(X ; U, Z)] = \text{CR}(U, Z) = 1$ and $E_Y[\text{CR}(U ; Y ; Z)] = \text{CR}(U ; Z)$.

Remark In this example, the trianglization factorization is not unique. We can also partition U out, and obtain the following factorization:

$$P(X, U, Y, Z) = \text{CR}(U ; X, Y) \text{CR}(X ; Y ; Z) P(X) P(U) P(Y) P(Z)$$

Remark In this example, to obtain the final factorization, we only used one (conditional) independency ($X \perp\!\!\!\perp Y \mid U, Z$). The other one ($U \perp\!\!\!\perp Z \mid X, Y$) has not been used. Hence, the independencies $\mathcal{I}$ encoded in the graph is not equivalent to the factorization F we obtained. Their relation is $\mathcal{I} \Rightarrow F$, but $\mathcal{I} \not\Leftarrow F$. Because we did not utilize all independencies, the factorization obtained seems not optimal in the sense that the scopes of factors, i.e., the variables involved in a factor, are bigger than maximum cliques.

3.2.5 Example 5: Proof of Hammersley-Clifford Theorem

A more complex example of using CR theorems is to prove the Hammersley-Clifford theorem. In Appendix A.3, we give an elegant proof of this theorem which is the most fundamental to Markov networks. The key of this proof is the unconnected nodes theorem (Theorem 17), which cancels the factors grouped into non-cliques out. See Appendix A.3 for details.

3.3 The Hypertree Representation

Wainwright et al. [6] propose the hypertree factorization which uses the hypertree as the representation. Hypertree factorization specifies a family of factorizations which can be reduced to a junction tree factorization. As we will see in the next section, a CRN factorization can also be reduced to a junction tree factorization. A CRN factorization is a hypertree factorization and can be represented as a hypertree. The specificity of our CRN factorization is derived from the MSIP. Following MSIP, a CRN factorization always strips off singleton probabilities and shared CR factors as illustrated by examples in the last section. But for a general hypertree factorization, it is not necessary to always strip off singleton probabilities. In this section, we explain the important graphical structure hypertree, and illustrate how to represent a CRN factorization with hypertrees with examples. The description mainly follows Wainwright et al. [6], which is also related to region graphs [27, 28].

3.3.1 Hypergraph and Hypertree

A hypergraph $H = (\mathcal{V}, \mathcal{E})$ consists of a set of nodes which is denoted by $\mathcal{V}$, and a set of hyperedges which is denoted by $\mathcal{E}$. A hyperedge is a non-empty subset of nodes. Following this perspective, an edge in an ordinary graph can be considered as a set which consists of a pair of nodes. Hence, an ordinary graph is a hypergraph. Based on the inclusion relation between sets, we can say a hyperedge is contained or included by another hyperedge if its nodes form a subset of another hyperedge's nodes. If a hyperedge is not included by any other hyperedge in a graph, it is a *maximal* hyperedge. In this sense, the inclusion relation defines a partial order between hyperedges. Following Wainwright et al. [6], a hypergraph is graphical represented with a diagram in which nodes represent hyperedges and directed edges represent inclusion relations between hyperedges. Figure 3.12 which is borrowed from Wainwright et al. [6] illustrates a graphical representation of a hypergraph.

Figure 3.12: A Hypergraph from Wainwright et al. [6]

The nodes stand for hyperedges. In this graph, there are three maximal hyperedges: $\{1, 2, 3, 4\}$, $\{2, 3, 5, 6\}$, and $\{3, 4, 6, 7\}$. The edges represent the inclusion relation between hyperedges. Acyclic hypergraphs are known as *hypertrees*.

3.3.2 Hypertree Factorization

A junction tree factorization can be represented with a hypergraph. The maximal edges in a hypergraph represent maximum cliques in a junction tree, and the intersection between two maximal hyperedges represent a separator set between two cliques. The running intersection property of junction trees guarantees such a hypergraph is acyclic. Hence, a junction tree factorization can be represented

as a hypertree. The hypertree factorization defines a family of factorizations which can be reduced to a ordinary junction tree factorization. We use the following example to illustrate this.

Figure 3.13: A Hypertree Factorization

For this graph, the ordinary junction tree factorization is as follows:

$$P(1, 2, 3, 4, 5) = \frac{P(1, 2, 5)P(1, 2, 3)P(2, 3, 4)}{P(1, 2)P(2, 3)}. \quad (3.28)$$

The hypertree factorization can be described as follows. For each hyperedge $e \in \mathcal{E}$, we assign the following factor:

$$\phi(e) = \frac{P(\mathcal{V}(e))}{\prod_{e' \in \mathcal{D}(e)} \phi(e')}$$

where $\mathcal{V}(e)$ are the nodes contained by e , and $\mathcal{D}(e)$ are all descendants of e . For example, in Figure 3.13, the descendants of the hyperedge $\{1, 2, 5\}$ are $\{1, 2\}$ and $\{2\}$. Then the joint probability of a hypertree $H(\mathcal{V}, \mathcal{E})$ can be represented as:

$$P(\mathcal{V}) = \prod_{e \in \mathcal{E}} \phi(e) \quad (3.29)$$

It is easy to check that the equation above can be reduced to a ordinary junction

tree factorization by canceling the common terms in numerators and denominators⁵.

For example, for the hyperedge $\{1, 2, 5\}$ in Figure 3.13, we assign the following factor:

$$\phi(1, 2, 5) = \frac{P(1, 2, 5)}{\phi(1, 2)\phi(2)} = \frac{P(1, 2, 5)}{\frac{P(1,2)}{P(2)}P(2)} \quad (3.30)$$

The complete hypertree factorization of Figure 3.13 is given as follows:

$$\begin{aligned} P(1, 2, 3, 4, 5) &= \frac{P(1, 2, 5)}{\frac{P(1,2)}{P(2)}P(2)} \times \frac{P(1, 2, 3)}{\frac{P(1,2)}{P(2)}\frac{P(2,3)}{P(2)}P(2)} \times \frac{P(2, 3, 4)}{\frac{P(2,3)}{P(2)}P(2)} \times \frac{P(1, 2)}{P(2)} \times \frac{P(2, 3)}{P(2)} \times P(2) \end{aligned}$$

This equation can be reduced to Equation 3.28.

3.3.3 Hypertree Representation for the CRN Factorization

The general hypertree factorization does not emphasize on the minimum shared information principle. The specificity of the CRN factorization stems from this point. As illustrated by examples in Section 3.2, to obtain a CRN factorization, we always strip off singleton probabilities and common CR factors to form new factors. For the hypertree factorization given in Figure 3.13, we obtain the following CRN factorization:

$$P(1, 2, 3, 4, 5) = \text{CR}(1, 2; 5)\text{CR}(2, 3; 4)\text{CR}(1; 2; 3)P(1)P(2)P(3)P(4)P(5). \quad (3.31)$$

This CRN factorization can be represented as the hypertree depicted in Figure 3.14. The specificity lies in that we always add singleton hyperedges which is denoted by red numbers. If we apply the hypertree factorization to Figure 3.14, we obtain the following factorization.

$$\begin{aligned} P(1, 2, 3, 4, 5) &= \text{CR}(1, 2; 5)\text{CR}(1, 2; 2, 3)\text{CR}(2, 3; 4)\text{CR}(1; 2)\text{CR}(2; 3)P(1)P(2)P(3)P(4)P(5). \end{aligned}$$

If we merge $\text{CR}(1, 2; 2, 3)$, $\text{CR}(1; 2)$ and $\text{CR}(2; 3)$, we obtain the CRN factorization in Equation 3.31. It is easy to verify that $\text{CR}(1; 2; 3) = \text{CR}(1, 2; 2, 3)\text{CR}(1; 2)\text{CR}(2; 3)$.

⁵This is related to the Möbius function. The same trick plays an important role in the proof of Hammersley–Clifford theorem.

Figure 3.14: A CRN Factorization

From this example, we see that CRN factorization is a hypertree factorization which specially emphasizes on MSIP.

3.4 Co-occurrence Rate Networks

Co-occurrence rate networks (CRNs) are undirected graphs. Similar to Bayesian networks and Markov networks, we first endow a set of independencies and a factorization to a CRN, and then examine the equivalence between the independencies and the factorization endowed.

Independencies We endow CRNs with the same independence semantics as Markov networks (Section 2.4) which is based on the concept U-separation. For the sake of readability, we repeat the definition as follows.

Definition 29 (U-Separation). *Let G be an undirected graph, and $\mathcal{X}$ be all nodes of G . $\mathcal{V}$, $\mathcal{W}$ and $\mathcal{U}$ are disjoint subsets of $\mathcal{X}$. We say $\mathcal{U}$ separates $\mathcal{V}$ and $\mathcal{W}$, denoted by $\text{sep}(\mathcal{V}; \mathcal{W} \mid \mathcal{U})$, if $\forall \mathcal{V}_i \in \mathcal{V}, \mathcal{W}_j \in \mathcal{W}$, for any path between them, at least one node of the path is in $\mathcal{U}$.*

Remark This definition just formalizes our intuition of separation.

Definition 30 (Co-occurrence Rate Networks Independencies). *Let G be an undirected graph, and $\mathcal{X}$ be all nodes of G .*

$$\mathcal{I}_{\text{CRN}} = \{(\mathcal{V} \perp\!\!\!\perp \mathcal{W} \mid \mathcal{U}) : \forall \text{sep}(\mathcal{V}; \mathcal{W} \mid \mathcal{U})\}.$$

This definition is identical to the independencies endowed to a Markov network (Definition 8). That is $\mathcal{I}_{\text{CRN}} = \mathcal{I}_{\text{MN}}$.

3.4.1 Acyclic Clique Graphs

For an acyclic clique graph, we endow a factorization as follows.

Definition 31 (CRN Factorization for Acyclic Clique Graphs). *Let G be an undirected graph G , and $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_N\}$ be all nodes (variables) of G . Let $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_M\}$ are all the maximum cliques (complete sub-graphs) in G . For a maximum clique $\mathcal{C}_i$, all of its separator sets are given by $\{\mathcal{S}_{ij} = \mathcal{C}_i \cap \mathcal{C}_j, \quad \forall \mathcal{C}_j \in \mathcal{C} - \mathcal{C}_i\}$ ⁶. And we define $\mathcal{S}_{ii} = \emptyset$. Let $\mathcal{S}_i = \cup \mathcal{S}_{i*}$. We define the factor over $\mathcal{C}_i$ as follows:*

$$\phi(\mathcal{C}_i) = \text{CR}(\text{Sem } \mathcal{C}_i - \mathcal{S}_i ; \mathcal{S}_{i1} ; \mathcal{S}_{i2} ; \dots ; \mathcal{S}_{iM}).$$

Then the CRN factorization is given by:

$$F_{\text{CRN}} : \quad P(\mathcal{X}) = \prod_{\mathcal{C}_i \in \mathcal{C}} \phi(\mathcal{C}_i) \prod_{\mathcal{X}_j \in \mathcal{X}} P(\mathcal{X}_j)$$

Theorem 18 (Equivalence between Factorization and Independencies of CRNs). *For acyclic clique graphs, we have*

$$\mathcal{I}_{\text{CRN}} \Leftrightarrow F_{\text{CRN}}$$

Proof. ($\mathcal{I}_{\text{CRN}} \Leftarrow F_{\text{CRN1}}$). This direction follows directly from Hammersley-Clifford Theorem (Theorem 4). We just need to consider CRs as non-negative functions and absorb singleton probabilities in the factorization into some cliques containing the singleton variable.

($\mathcal{I}_{\text{CRN}} \Rightarrow F_{\text{CRN1}}$). We first prove the following intermediate result.

$$P(\mathcal{X}) = \frac{\prod_{\mathcal{C}_i \in \mathcal{C}} \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_i})}{\prod_{\mathcal{S}_k \in \mathcal{S}} \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{S}_k})} \prod_{\mathcal{X}_j \in \mathcal{X}} P(\mathcal{X}_j) \quad (3.32)$$

⁶For the simplicity of presentation, here we assume separator sets are not overlapped. It can be easily extended to overlapped separator sets as illustrated by examples in the last two sections.

This follows the junction tree factorization immediately. We can also prove this by CR theorems as follows. First, we decouple the joint probability into a joint CR and singleton probabilities:

$$P(\mathcal{X}) = \text{CR}(\text{Sem } \mathcal{X}) \prod_{\mathcal{X}_j \in \mathcal{X}} P(\mathcal{X}_j).$$

We continue to decompose $\text{CR}(\text{Sem } \mathcal{X})$ as follows.

Since the clique graph is acyclic, we can always find a leaf in the clique graph. Suppose the leaf found is $\mathcal{C}_i$. And it has an adjacent clique $\mathcal{C}_j$. We duplicate the variables in $\mathcal{C}_i \cap \mathcal{C}_j$ by Theorem 12, and then partition $\mathcal{C}_i$ out by Theorem 10 from the rest part of the clique graph. In this way, we can keep a complete $\mathcal{C}_j$ in the rest part.

$$\begin{aligned} \text{CR}(\text{Sem } \mathcal{X}) &= \text{CR}(\text{Sem } \mathcal{X} ; \text{Sem}(\mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j})) \prod_{\mathcal{X}_k \in \mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j}} P(\mathcal{X}_k) \\ &= \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_i}) \times \text{CR}(\mathcal{X}_{\mathcal{C}_i} ; (\mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j}) \cup (\mathcal{X} - \mathcal{X}_{\mathcal{C}_i})) \\ &\quad \times \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_j} \cup (\mathcal{X} - \mathcal{X}_{\mathcal{C}_i})) \prod_{\mathcal{X}_k \in \mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j}} P(\mathcal{X}_k) \\ &= \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_i}) \times \text{CR}(\mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j} ; \mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j}) \\ &\quad \times \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_j} \cup (\mathcal{X} - \mathcal{X}_{\mathcal{C}_i})) \prod_{\mathcal{X}_k \in \mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j}} P(\mathcal{X}_k) \\ &= \frac{\text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_i})}{\text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j})} \times \text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_j} \cup (\mathcal{X} - \mathcal{X}_{\mathcal{C}_i})) \end{aligned}$$

We obtain the third and fourth equations by Equation 3.3, because $(\mathcal{X}_{\mathcal{C}_i} \perp\!\!\!\perp (\mathcal{X} - \mathcal{X}_{\mathcal{C}_i}) \mid \mathcal{X}_{\mathcal{C}_i} \cap \mathcal{X}_{\mathcal{C}_j})$. Then we repeat this process to factorize $\text{CR}(\text{Sem } \mathcal{X}_{\mathcal{C}_j} \cup (\mathcal{X} - \mathcal{X}_{\mathcal{C}_i}))$. If we continue this process until the last clique, we obtain Equation 3.32.

Based on Equation 3.32 we can obtain the CRN factorization (Definition 31) by applying the merge operations (Theorem 11) to CR factors if it is necessary. We have illustrated this in Equation 3.26. $\square$

3.4.2 Cyclic Clique Graphs

Following the triangulation of junction trees, cyclic clique graphs can be first transformed to acyclic clique graphs. An example of the triangulation process

has been illustrated in Section 3.2.4. And then the acyclic clique graphs can be factorized as described in the last section.

3.5 Inference

We do not see any advantage brought by the CRN factorization for inference. The best way for inference with a CRN factorization is going back to the junction tree factorization. This is because the factors of a CRN factorization are smaller than marginals used in a junction tree factorization. In inference, we normally need to calculate some marginal. But if we use a CRN factorization, we need to calculate marginals from smaller factors which are CR factors and singleton probabilities. This takes some additional time comparing with an equivalent junction tree factorization.

3.6 Learning

In the next chapter, we will discuss a two-step separate training method for CRNs.

Two-step Separate Training for CRNs

The contents of this chapter have been published by Zhu et al. [29].

Outline

In the previous chapters we illustrated that the factors in a CRN factorization hold the following properties: (i) a singleton probability is independent of other singleton probabilities; (ii) a CR factor is independent of other CR factors conditioned by singleton probabilities. Based on these properties, we propose an efficient two-step separate training algorithm for CRNs. In the first step, we train singleton probabilities separately. Separate training is allowed here because a singleton probability does not share information with another singleton probability. Then we fix the learned singleton probabilities, and train CR factors separately. This is allowed because a CR factor does not share information with other CR factors conditioned by singleton probabilities. In each step, we train separate log-linear models to approximate the maximum likelihood estimation (MLE) of singleton probabilities and CR factors.

This chapter is organized as follows. We first discuss the MLE of CRNs in Section 4.1. In Section 4.2, we describe separate models for singleton probabilities and CR factors. In Section 4.3 we discuss issues about the consistency between separate models.

4.1 Maximum Likelihood Estimation of Co-occurrence Rate Networks

Chow and Liu [30] show that for a tree-structured graph, the MLE of marginals corresponds to their empirical relative frequencies. Based on this result, we

obtain that the MLE for co-occurrence rate networks are their empirical relative frequencies. Because the MLE is independent of parametrization. More specifically, the MLE for singleton probabilities and CR factors are given as follows:

$$\tilde{P}(\mathcal{X}_i = \mathbf{x}_i) = \frac{\#(\mathbf{x}_i; \mathcal{D})}{|\mathcal{D}|} \quad (4.1)$$

$$\tilde{\text{CR}}(\mathcal{X}_{\mathcal{C}_j} = \mathbf{x}_{\mathcal{C}_j}) = \frac{\#(\mathbf{x}_{\mathcal{C}_j}; \mathcal{D})/|\mathcal{D}|}{\prod_{\mathcal{X}_i \in \mathcal{X}_{\mathcal{C}_j}} \tilde{P}(\mathbf{x}_i)} \quad (4.2)$$

where $\#(\mathbf{x}_i; \mathcal{D})$ is the frequency of $\mathbf{x}_i$ in the dataset $\mathcal{D}$. And $|\mathcal{D}|$ is the number of instances in $\mathcal{D}$. $\mathcal{X}_i$ is a node and $\mathcal{X}_{\mathcal{C}_j}$ is a maximum clique in the graph.

4.2 Separate Models

In the previous section, we obtained the MLE for CRNs, which are the empirical relative frequencies. But empirical relative frequencies can only be counted from the training data for seen instances. We need to train models for making predictions for new instances. As discussed in Chapter 3, the factors in a CRN factorization hold the following properties: (i) a singleton probability is independent of other singleton probabilities; (ii) a CR factor is independent of other CR factors conditioned by singleton probabilities. Based on these properties, we propose a two-step separate training method. In the first step, we train a separate model for each singleton probability and in the second step we train a separate model for each CR factor. Log-linear models are used for modeling singleton probabilities and CR factors.

4.2.1 Log-linear Models

In the area of graphical models, log-linear functions are widely used as local models¹. In a log-linear model, a linear combination of features forms the exponent of an exponential function:

$$\phi(\mathcal{X}_S) = \exp\{\lambda_1 f_1(\mathcal{X}_S) + \dots + \lambda_N f_N(\mathcal{X}_S)\}$$

where $\mathcal{X}_S$ are variables in a local factor, and $\{f_i, i = 1, \dots, N\}$ are feature functions defined over $\mathcal{X}_S$. These features are local features because they are extracted from a local factor. There are several benefits of using log-linear models.

¹Local models are used for modeling local factors in a factorization.

Firstly, with exponential functions, multiplication and logarithm can be easily calculated. Secondly, exponential functions are non-negative which meets the non-negativity property of factors. In other words, we do not need to worry about the non-negativity constraints in optimization. They are automatically satisfied if we use a log-linear model. Also, to some degree, using log-linear models, we can avoid numerical overflow and underflow.

4.2.2 Log-linear Models for Singleton Probabilities

We use the log-linear model for representing a singleton probability. Suppose for $P(\mathcal{X}_j)$, we extract features by a vector of feature functions $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_M)$. For each component $\mathcal{G}_i$, the input is $\mathcal{X}_j$ and output is a real number:

$$\mathcal{G}_i : \mathcal{X}_j \rightarrow \mathcal{R}.$$

And

$$\mathcal{G} : \mathcal{X}_j \rightarrow \mathcal{R}^M \quad \mathbf{x}_j \mapsto (\mathcal{G}_1(\mathbf{x}_j), \mathcal{G}_2(\mathbf{x}_j), \dots, \mathcal{G}_M(\mathbf{x}_j)).$$

Accordingly, we set a vector of parameters as weights of local features: $\Omega = (\omega_1, \omega_2, \dots, \omega_M)$. Then the log-linear model can be written as follows:

$$P(\mathcal{X}_j = \mathbf{x}) \approx \exp\{\Omega \cdot \mathcal{G}(\mathcal{X}_j = \mathbf{x})\}.$$

We use this model to approximate the MLE of singleton probabilities. In other words, we use the following pairs for training this log-linear model:

$$\{(\mathbf{x}_j, \frac{\#(\mathbf{x}_j; \mathcal{D})}{|\mathcal{D}|}), \forall \mathbf{x}_j \in \text{Val}(\mathcal{X}_j)\}$$

where $\mathcal{D}$ is the training dataset. In experiments, we use the empirical MLE of singleton probabilities for instances have been seen in the training data. For new instances, we use the predictions generated by the log-linear model.

4.2.3 Log-linear Models for CR Factors

As discussed in the previous chapter, in a CRN factorization, a CR factor is independent of other CR factor conditioned by singleton probabilities. After we trained the separate models for singleton probabilities, we can fix the singleton probabilities and train separate models for CR factors. We do not need to worry about the consistency between the predictions generated by separate CR models. But we need to consider the consistency between CR model and

corresponding models for singleton probabilities. Because they are under some local constraints, such as $\sum_{X,Y} \text{CR}(X; Y)P(X)P(Y) = 1$. We will discuss this issue in the next section.

We also use the log-linear models for representing CR. The consideration here is to make the models consistent in form. That is if $\text{CR}(X; Y)$, $P(X)$ and $P(Y)$ are all log-linear models, the joint probability $P(X, Y)$ is also a log-linear model.

Similarly, suppose for $\text{CR}(\text{Sem } \mathcal{X}_{C_i})$, we extract N local features by a vector of feature functions $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_N)$. For each component $\mathcal{F}_i$, the input is $\mathcal{X}_{C_i}$ and output is a real number:

$$\mathcal{F}_i : \mathcal{X}_{C_i} \rightarrow \mathcal{R}.$$

And

$$\mathcal{F} : \mathcal{X}_{C_i} \rightarrow \mathcal{R}^N \quad \mathbf{x}_{C_i} \mapsto (\mathcal{F}_1(\mathbf{x}_{C_i}), \mathcal{F}_2(\mathbf{x}_{C_i}), \dots, \mathcal{F}_N(\mathbf{x}_{C_i})).$$

Accordingly, we set a vector of parameters as weights for features: $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_N)$. Hence, the log-linear model for $\text{CR}(\text{Sem } \mathcal{X}_{C_i})$ can be written as follows:

$$\text{CR}(\text{Sem } \mathcal{X}_{C_i}) \approx \exp\{\Lambda \cdot \mathcal{F}(\mathcal{X}_{C_i})\}.$$

We use this model to approximate the MLE of CR factors. In other words, we use the following pairs for training this log-linear model:

$$\{(\mathbf{x}_{C_i}, \frac{\#(\mathbf{x}_{C_i}; \mathcal{D})/|\mathcal{D}|}{\prod_{\mathbf{x}_i \in \mathcal{X}_{C_i}} \tilde{P}(\mathbf{x}_i)}), \forall (\mathbf{x}_{C_i} \in \text{Val}(\mathcal{X}_{C_i}))\}$$

where $\mathcal{D}$ is the training dataset, and $\tilde{P}(\mathbf{x}_i)$ is the MLE of singleton probabilities (Equation 4.1). In experiments, we use the empirical MLE of co-occurrence rates for instances which have been seen in the training data. For new instances, we use the predictions generated by the log-linear model.

4.3 Consistency

The issues of consistency stem from two aspects: (i) shared information between factors; and (ii) normalization constraints across factors.

4.3.1 Shared Information

As we illustrated by examples in Section 3.2, there can be two kinds of *shared information* between factors. (i) The first kind of shared information is the information contained by singleton probabilities (see Section 3.2.1 and Section 3.2.2). For example, $P(X, Y)$ and $P(Y, Z)$ share the information $P(Y)$. To avoid this kind of shared information, a CRN factorization always strip off singleton probabilities from a joint probability. That is $P(X, Y)P(Y, Z) = \text{CR}(X; Y)\text{CR}(Y; Z)P(X)P(Y)P(Z)$. (ii) The second kind of shared information is the dependence information contained by CR factors (see Section 3.2.3). For example, $\text{CR}(X; Y; Z)$ and $\text{CR}(Y; Z; V)$ share the dependence information $\text{CR}(Y; Z)$. To avoid this kind of shared information, we can strip off $\text{CR}(Y; Z)$ by the merge operation (Theorem 11). That is $\text{CR}(X; Y; Z)\text{CR}(Y; Z; V) = \text{CR}(X; Y; Z; V)\text{CR}(Y; Z)$. This equation is obtained by applying the merge operation to merge Y and Z in the factor $\text{CR}(Y; Z; V)$. Hence, a CRN factorization avoids these two kinds of shared information, and we do not need to pay efforts to calibrate the shared information between separate models.

4.3.2 Normalization Constraints

Even though shared information is avoided in a CRN factorization, the separate models still need to satisfy some *normalization constraints*. We take a linear graph consisting of nodes $\{X, Y, Z\}$ as the example to illustrate these constraints. The CRN factorization for this example can be written as follows.

$$P(X, Y, Z) = \text{CR}(X; Y)\text{CR}(Y; Z)P(X)P(Y)P(Z).$$

Constraints on a singleton probability. The first constraint is on a singleton probability. For example, one axiom of probability requires $\sum_X P(X) = 1$. A separate model for a singleton probability may output a distribution which violates this constraint. But this constraint is *not* a real problem, because this constraint is within a separate model. This can be easily fixed by selecting a separate model which is purposed for predicting probability distributions. For such models, the output predictions meet probability axioms. We may also do a post-train normalization to adjust output predictions to meet this constraint.

Local constraints between $\text{CR}(X ; Y)$, $P(X)$ and $P(Y)$. According to the definitions, there are three constraints to be satisfied:

$$\sum_X \text{CR}(X ; Y)P(X) = 1 \quad (4.3)$$

$$\sum_Y \text{CR}(X ; Y)P(Y) = 1 \quad (4.4)$$

$$\sum_{X,Y} \text{CR}(X ; Y)P(X)P(Y) = 1 \quad (4.5)$$

Because $\sum_X \text{CR}(X ; Y)P(X) = 1$ and $\sum_Y P(Y) = 1$ implies $\sum_{X,Y} \text{CR}(X ; Y)P(X)P(Y) = 1$. Hence, these three local constraints can be reduced to the first two. This kind of local constraints are across different separate models. Hence we cannot guarantee the consistency between them if we insist on training them separately. In our two-step algorithm, $P(X)$ and $P(Y)$ have been trained in the first step. We can treat them as known parameters and plug them into the constraints. In this way, we can satisfy these constraints. Another more efficient way is by doing post-train normalization to CR factors to meet these constraints. In practice, we should consider a trade-off between training speed and consistency. For a MAP task in which we try to find an assignment with a greatest probability, only the relative quantity matters. For such a task, the post-train normalization seems not important.

Global constraints. The global constraint is given as follows.

$$\sum_{X,Y,Z} \text{CR}(X ; Y)\text{CR}(Y ; Z)P(X)P(Y)P(Z) = 1$$

Fortunately, we do not need to pay additional efforts to meet this constraint. Because the three local constraints given in Equation 4.3, 4.4 and 4.5 imply this global constraint. In this example we have the following equation:

$$\begin{aligned} & \sum_{X,Y,Z} \text{CR}(X ; Y)\text{CR}(Y ; Z)P(X)P(Y)P(Z) \\ &= \sum_{X,Y} \text{CR}(X ; Y)P(X)P(Y) \sum_Z \text{CR}(Y ; Z)P(Z) \end{aligned}$$

This results can be easily extended to acyclic clique graphs.

From the discussions above, we can see that the real problems of consistency in CRNs are the first two local constraints. In future, we will explore how to improve the current model to fit these constraints better.

4.3.3 Support Vector Regression

There exist a rich set of regression models which may be used for solving the parameters. Particularly, we adopt the support vector regression (SVR) [33] in our experiments which will be described in the next chapter. SVR is linear in the high dimensional transformed space and tolerant to error points with small residuals. The tolerance to low error point seems to fit the natural language processing applications well. Because in natural language processing applications, the input and final output are normally categorical. A small error will not affect the final results. For example, in the task of text classification, large-margin methods achieve very good results [34]. Another consideration is that there are good out-of-box implementations of SVR which can handle very large number of instances and features efficiently. These considerations lead us to adopt SVR. In future, we will try other regression models. To avoid endowing unwanted metric and order structures to a single categorical variable, we use the dummy coding² as the representation of categorical input variables.

4.4 Summary

In this chapter, we obtained the closed-form MLE for CRNs based on the MLE of tree-structured graphs. The MLE of CRNs are empirical relative frequencies which can be counted from the training data. Based on the independence properties of factors in a CRN factorization, we propose a two-step separate training algorithm to approximate the MLE of singleton probabilities and CR factors. We use a separate model to represent each singleton probability and CR factor. This results in an efficient training method.

²https://en.wikipedia.org/wiki/Categorical_variable

Experiments on Chain-structured CRNs

The contents of this chapter have been published by Zhu et al. [35] and Zhu et al. [36]. The software and datasets used in this chapter are available at <https://github.com/zheminzhu/Co-occurrence-Rate-Networks>.

Outline

In this chapter, we apply (chain-structured) co-occurrence rate networks to two real-world natural language processing (NLP) applications: named entity recognition (NER) and part-of-speech (POS) tagging. We compare CRNs with conditional random fields which are the state-of-the-art models for these two tasks. Conditional random fields can be considered as conditional Markov networks. This chapter is structured as follows. The two tasks, NER and POS tagging, are briefly introduced in Section 5.1 and Section 5.2, respectively. Then we review the related models used for these two tasks in Section 5.3. Experimental results are reported in Section 5.6. Summary follows in the last section.

5.1 Named Entity Recognition

Named entities, e.g. New York, are proper nouns in texts. A named entity may contain multiple words. Multiple words in a named entity are often treated as a whole in NLP applications, such as relation extraction. With sufficient contextual information, a named entity in texts can be uniquely mapped to an object in the physical world.¹

¹But commonly, due to lack of contextual information, the mapping may be ambiguous. The task of named entity disambiguation [37] is to find the most likely mapping between named entity expressions in texts and objects in the physical world.

Named entities are belonging to a set of predefined categories called NE labels, such as person, organization, location, time, etc. The task of name entity recognition is to locate and classify named entities in free texts. In information extraction, which extracts relations between entities, named entity recognition is normally the first step. Named entity recognition can be conducted at different levels, such as sentence level, article level, corpus level or web level. In this thesis, we focus on recognizing named entities at sentence level. Given a sentence, which is considered as a sequence of tokens², NER systems assign each token a label which indicates the category of named entity, such as location (LOC), person (PER), organization (ORG), or out of a named entity (O). Here we give a running example to illustrate the typical scenario of named entity recognition at sentence level.

```
[Victor]PER [de]PER [Graaff]PER [is]O [a]O [member]O [of]O
[the]O [Dutch]ORG [National]ORG [Research]ORG [School]ORG
[for]ORG [Knowledge]ORG [Systems]ORG [.]O
```

In this example, [Victor de Graaff] is a person and [Dutch National Research School for Knowledge Systems] is an organization. Other words and the period do not belong to any named entity. A NER system automatically assigns labels to each token in a sentence. Thus, the output of a NER system is a sequence of NE labels. This is a typical structured prediction task. Structured prediction is quite different from the ordinary classification task, in which the output is normally a single label. From this example, we can observe the following facts. To judge whether a token belongs to a named entity or not, there are two kinds of useful information:

1. **Observation evidence** The first kind of information is the observation evidence. Tokens to be labelled in the sentence provide the observable evidence. Obviously, some tokens are more likely to be in a named entities than others. For example, the words starting with capital letters, such as [Victor], [Graaff] and [Dutch], are more likely to appear in names of people or organizations. In contrast, some tokens, such as the copula verb [is] and the article [a], are of lower possibility to be in a named entity.
2. **Dependence between adjacent labels** The second kind of information is the dependence between adjacent labels. That is a token tends to be in the same category of its neighbours. For example, the token [National]

²Tokens include both words and punctuations.

is more likely to be in a named entity if its direct neighbours are in a named entity. Note that the adjacent labels are not observable before we output the labels. Hence, this kind of information is not observable evidence. How to model this kind of information is the critical issue in structured prediction.

We need to consider both kinds of information in one model. This can be done with a conditional chain-structured CRN as depicted in Figure 5.1.

Figure 5.1: A Chain-structured CRN for Named Entity Recognition

In Figure 5.1, the node O stands for ‘Observation’, i.e. tokens in a sentence. The nodes $[NE_1, NE_2, \dots, NE_N]$ is a sequence of random variables which represent NE labels for tokens in the sentence. In this model, we consider the observation as a condition. The directed edges from the observation to NE labels model the first kind of information, i.e. the observation evidence. The undirected edges between adjacent NE labels model the second kind of information, i.e. the dependence between adjacent labels. Here we use undirected edges, because adjacent labels mutually depend on each other. Undirected edges are suitable for modelling mutual dependence. Also this is a first-order model, which implies a label only directly depends on its immediate neighbours.

Independencies According to the independence semantics of CRNs, conditioned by the observation O , we have the following independencies:

$$NE_i \perp\!\!\!\perp (NE_{i+2}, NE_{i+3}, \dots, NE_N) \mid NE_{i+1}$$

$$NE_i \perp\!\!\!\perp (NE_0, NE_2, \dots, NE_{i-2}) \mid NE_{i-1}$$

Factorization On a chain-structured CRN, we obtain the following factorization which is equivalent to the independencies given above:

$$P(\text{NE}_1, \text{NE}_2, \dots, \text{NE}_N \mid O) = \prod_{i=1}^{N-1} \text{CR}(\text{NE}_i; \text{NE}_{i+1} \mid O) \prod_{j=1}^N P(\text{NE}_j \mid O) \quad (5.1)$$

where $\{\text{NE}_i, i = 1, \dots, N\}$ are discrete random variables whose sample space consists of all NE labels.

Modelling named entity recognition Finally, the task of named entity recognition can be formalized as a task to find an assignment of a sequence of NE labels which maximizes the following conditional joint probability:

$$\text{argmax}_{\text{ne}_1, \text{ne}_2, \dots, \text{ne}_N} P(\text{NE}_1, \text{NE}_2, \dots, \text{NE}_N \mid O)$$

5.2 Part-of-speech Tagging

Part-of-speech (POS) tagging is another important natural language processing task which can be modelled in a similar way as the task of named entity recognition. The part-of-speech of a word, such as noun, verb, etc., reflects grammatical functions of the word. Words with similar grammatical properties are grouped in the same part-of-speech category. The task of POS tagging is to classify each word in a sentence to a POS category. POS tagging is normally conducted at sentence level, and often serves as a important pre-processing step for a wide range of advanced NLP applications. We use the running example to illustrate this task as follows:

```
NNP/Victor NNP/de NNP/Graaff VBZ/is DT/a NN/member
PREP/of DT/the NNP/Dutch ADJ/National NNP/Research
NNP/School PREP/for NNP/Knowledge NNPS/Systems ./.
```

In this example, each word is assigned with a POS tag. There are different part-of-speech tagsets. The Upenn tagset [38] is the most popular one. Similar to the task of named entity recognition, from this example, we observe that there are two kinds of information that can be used in predicting POS tags:

1. **Observation evidence** The evidence we can observe are just the tokens in the sentence. For example, some tokens are more likely to be nouns, such as [School] and [Knowledge], while others, such as [of] and [for], are prepositions most of the time.

2. **Dependence between adjacent labels** But sometimes observation evidence is not enough to make the correct prediction. For instance, [de] ('the' in English) in Dutch language is normally an article. But in our example, [de] is a part of a noun phrase³. In such a difficult case, we resort to the second kind of information, i.e. the dependence between adjacent labels. In other words, [de] is more likely to be in a noun phrase when its neighbours are nouns.

The observation above suggests that we can use the same chain-structured CRN to model the task of POS tagging.

Figure 5.2: A Chain-structured CRN for Part-of-speech Tagging

Independencies According to the independence semantics of CRNs, conditioned by the observation O , we have the following independencies:

$$\begin{aligned} \text{POS}_i &\perp\!\!\!\perp (\text{POS}_{i+2}, \text{POS}_{i+3}, \dots, \text{POS}_N) \mid \text{POS}_{i+1}, O \\ \text{POS}_i &\perp\!\!\!\perp (\text{POS}_0, \text{POS}_2, \dots, \text{POS}_{i-2}) \mid \text{POS}_{i-1}, O \end{aligned}$$

Factorization Similarly, on a chain-structured CRN, we obtain the following factorization which is equivalent to the independencies given above:

$$P(\text{POS}_1, \text{POS}_2, \dots, \text{POS}_N \mid O) = \prod_{i=1}^{N-1} \text{CR}(\text{POS}_i; \text{POS}_{i+1} \mid O) \prod_{j=1}^N P(\text{POS}_j \mid O) \quad (5.2)$$

where $\{\text{POS}_i, i = 1, \dots, N\}$ are discrete random variables whose sample space consists of all POS tags.

³In Dutch, [de] used in this case is called *tussenvoegsel*.

Modelling part-of-speech tagging Similar to the task of named entity recognition, part-of-speech tagging can be formalized as the task of finding an assignment a sequence of POS tags that maximizes the following conditional joint probability:

$$\operatorname{argmax}_{\text{pos}_1, \text{pos}_2, \dots, \text{pos}_N} P(\text{POS}_1, \text{POS}_2, \dots, \text{POS}_N \mid O)$$

In order to make the notations general, in the rest of this chapter, we use $S = (S_1, S_2, \dots, S_N)$ to denote labels, no matter NE labels or POS tags. That is we summarize Equation 5.1 and Equation 5.2 by a single equation as follows:

$$P(S_1, S_2, \dots, S_N \mid O) = \prod_{i=1}^{N-1} \text{CR}(S_i; S_{i+1} \mid O) \prod_{j=1}^N P(S_j \mid O) \quad (5.3)$$

5.3 Related Models

The task of predicting a sequence of labels is normally called sequence labelling. Sequence labelling has been studied for a long history in different areas. There exist several widely used models for sequence labelling, such as hidden Markov models (HMMs) [39], conditional Markov models (CMMs) [40] and conditional random fields (CRFs) [41].⁴ All of them can be put under the umbrella of probabilistic graphical models even although they have been proposed in different areas and for different purposes. Hidden Markov models and conditional Markov models are directed models. In contrast, conditional random fields are undirected models. We give a brief review for these models as follows.

5.3.1 Hidden Markov Models

Figure 5.3 depicts a first order HMM. In this picture, $S = [S_1, S_2, \dots, S_N]$ is a sequence of labels, and $O = [O_1, O_2, \dots, O_N]$ is a sequence of observation. Using the task of named entity recognition as an example, $[S_1, S_2, \dots, S_N]$ is a sequence of NE labels and $[O_1, O_2, \dots, O_N]$ is a sequence of tokens. HMMs are directed models, and can be considered as a type of Bayesian network [43]. The independencies of a HMM can be judged by d-separation (Definition 6).

⁴Another popular model is structured (structural) SVM [42] which essentially applies factorization to kernels. Due to its lack of a obvious probabilistic interpretation, we do not include it in this short review.

Figure 5.3: Hidden Markov Models

According to the definition of the factorization of Bayesian networks (Definition 3), the factorization of a HMM can be written as follows:

$$P(\mathcal{S}, \mathcal{O}) = P(S_1) \prod_{i=1}^N P(\mathcal{O}_i | S_i) \prod_{j=1}^{N-1} P(S_{j+1} | S_j). \quad (5.4)$$

From Equation 5.4, we can see the following facts [44]: (i) HMMs are generative models. In other words, HMMs model a joint probability $P(\mathcal{S}, \mathcal{O})$. This also leads to a mismatch problem. In the training stage, HMMs optimize a joint probability $P(\mathcal{S}, \mathcal{O})$. But in the decoding stage, HMMs search for a sequence of labels which maximizes a conditional probability $P(\mathcal{S} | \mathcal{O})$. Klein et al. [45] show that this mismatch problem can decrease accuracy; (ii) The label transition probabilities $P(S_{j+1} | S_j)$ in Equation 5.4 are not conditioned by observations. This is also different from other models. For example, in CRNs, we model the dependence between labels with a conditioned of the observation, which can be observed from $\text{CR}(S_i; S_{i+1} | O)$ in Equation 5.3. That is, HMMs use universal label transition probabilities $P(S_{j+1} | S_j)$ without respect to observations. Hence it is difficult to incorporate observation evidence into HMMs to help predict label transition probabilities. Transition features extracted from observation evidence may contain valuable information.

To avoid the mismatch problem, we need to model the conditional probability $P(\mathcal{S} | \mathcal{O})$. And in order to utilize the transition features, we can set observation evidence as a condition in transition factors. Conditional Markov models discussed below implement these ideas.

Figure 5.4: A Conditional Markov Model

5.3.2 Conditional Markov Models

Figure 5.4 illustrates a conditional Markov models (CMM) [40]. From the graph, we can see that: (i) CMMs are directed models (Bayesian networks); (ii) CMMs are discriminative models which factorize a conditional joint probability $P(S | \mathcal{O})$, which is different from HMMs. According to Definition 3, we obtain the following factorization for a CMM:

$$P(S | \mathcal{O}) = P(S_1 | \mathcal{O}) \prod_{i=1}^{N-1} P(S_{i+1} | S_i, \mathcal{O}) \quad (5.5)$$

CMMs avoid the mismatch problem of HMMs because they are discriminative models. And the transition probabilities $P(S_{i+1} | S_i, \mathcal{O})$ are conditioned by previous label S_i as well as the observation $\mathcal{O}$. hence observation features can be easily encoded into transition probabilities. This is the second advantage of CMMs over HMMs. But CMMs suffer from another problem called the label bias problem (LBP) [41]. Intuitively, this problem is caused by the label dependence (given by S_i) and observation evidence (given by $\mathcal{O}$) are mixed together in one factor $P(S_{i+1} | S_i, \mathcal{O})$. One of them may dominate the factor when its distribution is of low entropy, while the other is underestimated or even ignored. An extreme case is when S_i has only one possible out-going transition S_{i+1} ⁵, then $P(S_{i+1} | S_i, \mathcal{O})$ is always equal to 1 no matter what the observation $\mathcal{O}$ is. That is the observation evidence $\mathcal{O}$ is ignored and the label dependence dominates the factor.

We use the following example to illustrate the label bias problem more clearly. Suppose the training dataset consists of 21 training instances including

⁵In this extreme case, the entropy of $P(S_{i+1} | S_i)$ is the lowest: 0.

11 of (rib : XIB), 9 of (rob : YOB) and 1 of (rob : XIB), where $\{r, o, i, b\}$ are observations and $\{X, Y, O, I, B\}$ are tags. At test time, we want to predict the tags for the observation sequence (rob). Obviously, the correct tags for (rob) should be (YOB) rather than (XIB). Because there are 9 of (rob : YOB) and only 1 of (rob : XIB). But according to Equation 5.5, $P(YOB | rob) = P(Y | r)P(O | Y, o)P(B | O, b) = \frac{9}{21} \times 1 \times 1 = \frac{9}{21}$, which is smaller than $P(XIB | rob) = P(X | r)P(I | X, o)P(B | I, b) = \frac{12}{21} \times 1 \times 1 = \frac{12}{21}$. So CMMs will mislabel (rob) as (XIB). The reason is that (X) has only one outgoing transition (I). This constrains $P(I | X, o)$ to be 1 even though $P(I | o)$ is very small. That is (o) is not used for prediction its tag. The tag of (o) totally depends on the previous tag. From this example, we can see that the local conditional probabilities $P(\mathcal{S}_{i+1} | \mathcal{S}_i, \mathcal{O})$ cause the label bias problem. See Lafferty et al [41], Hong [46] and Zhu et al. [20] for more discussions. Koller and Friedman [3] also provide examples to explain this problem.

To avoid the label bias problem, we need to guarantee that the observation evidence can always be used in predicting the next label⁶. This can be done by decoupling the label dependence and observation evidence into different factors, such that none of them can dominate the other. Conditional random fields (Section 5.3.4) and co-occurrence rate networks (Section 5.4) implement this idea, and hence they are immune to the label bias problem.

5.3.3 Bi-directed Networks

Figure 5.5: A Bi-directed Network

Besides undirected edges, there is another idea to model mutual dependence: bi-directed edges. As depicted in Figure 5.5, bi-directed networks (BDNs) [47]

⁶HMMs do not suffer from the label bias problem, because the factors $P(\mathcal{O}_i | \mathcal{S}_i)$ in Equation 5.4 guarantee that the observation evidence is always used.

use bi-directed edge to model mutual dependence. The factorization of BDNs can be considered as using pseudolikelihood [48] to approximate the joint conditional probability as follows:

$$P(\mathcal{S} \mid \mathcal{O}) \propto \prod_{i=1}^N P(\mathcal{S}_i \mid \text{Pa}_i, \mathcal{O})$$

where Pa_i are all parents of $\mathcal{S}_i$. For example, $\text{Pa}_2 = \{\mathcal{S}_1, \mathcal{S}_3\}$ in Figure 5.5. Even though BDNs work well in the case of using rich features in practice, there are some difficulties to precisely define the independence semantics and the equivalent factorization for bi-directed networks. If we assign the same independencies of Markov networks to BDNs, the factorization of BDNs is not equivalent to these independencies. Also we can not assign the same independencies as Bayesian networks to BDNs, because the d-separation criterion (Definition 6) does not work for bi-directed edges. To be a well-defined model, these fundamental issues need to be solved first.

5.3.4 Conditional Random Fields

Figure 5.6: A Conditional Random Field

Figure 5.6 shows a chain-structured conditional random field (CRFs) [41]. The graph structure is essentially the same as the graphs in Figure 5.1 and 5.2. CRFs are conditional (discriminative) Markov networks. According to the factorization of Markov networks (Definition 9), the factorization of CRFs can be written as follows:

$$P(\mathcal{S} \mid \mathcal{O}) = \frac{1}{Z_{\mathcal{O}}} \prod_{i=1}^{N-1} \psi_i(\mathcal{S}_i, \mathcal{S}_{i+1}, \mathcal{O}) \prod_{j=1}^N \phi_j(\mathcal{S}_j, \mathcal{O}),$$

where $Z_{\mathcal{O}}$ is a global normalization constant, which is also called partition function. ψ and ϕ are non-negative factors defined over pairwise and singleton cliques.

CRFs avoid the known problems we mentioned above. From the factorization, we can see that CRFs model conditional joint probabilities $P(S \mid \mathcal{O})$. Hence they avoid the mismatch problem of HMMs. Also the pairwise factors $\psi(S_i, S_{i+1}, \mathcal{O})$ models label dependence including the observations. Hence the observation features can be easily encoded into CRFs. Furthermore, CRFs decouple the label dependence (modeled by $\psi(S_i, S_{i+1}, \mathcal{O})$) and observation evidence (modeled by $\phi(S_j, \mathcal{O})$) into different factors. This guarantees that none of them can dominate the other. Singleton factors $\phi(S_j, \mathcal{O})$ guarantee that $\mathcal{O}$ is always used for prediction. Therefore the label bias problem is avoided.

Nevertheless, the obtaining the maximum likelihood estimation (MLE) of CRFs can be very expensive [44, 49]. This is because CRFs have no closed-form MLE, and we need to re-calculate the global partition function $Z_{\mathcal{O}}$ for each instance in each optimization iteration. Co-occurrence rate networks also avoid the problems above, but they can be trained much more efficiently because CRNs have a closed-form MLE.

5.4 CRNs are Immune to the Label Bias Problem

In this section, we use an example to illustrate that CRNs are Immune to the label bias problem (LBP). As we discussed above, the LBP is a model defect of conditional Markov models (CMMs) [41]. The LBP stems from the nature of the CMMs factorization (Equation 5.5). The CMMs factorization includes factors $P(S_{i+1} \mid S_i, \mathcal{O})$ which are *local conditional probabilities*. These local conditional probabilities prefer the S_i with fewer outgoing transitions. The extreme case is when S_i has only one possible outgoing transition S_{i+1} , then $P(S_{i+1} \mid S_i, \mathcal{O})$ is always equal to 1 no matter what $\mathcal{O}$ is. This is because we need to normalize $P(S_{i+1} \mid S_i, \mathcal{O})$ to make its summation equal 1. The ignorance of the observation $\mathcal{O}$ may lead to low accuracy. CRNs (Equation 5.3) are immune to this problem, because it does not include local conditional probabilities. We use the following example to illustrate this. In Section 5.6.3, we also provide experimental evidence.

Suppose the training dataset consists of 21 training instances including 11 of (rib : XIB), 9 of (rob : YOB) and 1 of (rob : XIB), where $\{r, o, i, b\}$ are observations and $\{X, Y, O, I, B\}$ are tags. At test time, we want to predict the tags for the observation sequence (rob). Obviously, the correct tags for (rob)

should be (Y0B) rather than (XIB). Because there are 9 of (rob : Y0B) and only 1 of (rob : XIB). But according to CMM factorization,

$$P(Y0B|rob) = P(Y|r)P(0|Y, o)P(B|0, b) = \frac{9}{21} \times 1 \times 1 = \frac{9}{21},$$

which is smaller than

$$P(XIB|rob) = P(X|r)P(I|X, o)P(B|I, b) = \frac{12}{21} \times 1 \times 1 = \frac{12}{21}$$

So CMMs will mislabel (rob) as (XIB). According to Equation 5.3, we have

$$\begin{aligned} P(Y0B|rob) &= CR(Y; 0|ro)CR(0; B|ob)P(Y|r)P(0|o)P(B|b) \\ &= \frac{9/10}{9/21 \times 9/10} \times \frac{9/10}{9/10 \times 1} \times \frac{9}{21} \times \frac{9}{10} \times 1 \\ &= \frac{9}{10} \end{aligned}$$

which is bigger than

$$\begin{aligned} P(XIB|rob) &= CR(X; I|ro)CR(I; B|ob)P(X|r)P(I|o)P(B|b) \\ &= \frac{1/10}{12/21 \times 1/10} \times \frac{1/10}{1/10 \times 1} \times \frac{12}{21} \times \frac{1}{10} \times 1 \\ &= \frac{1}{10}. \end{aligned}$$

That is, CRNs make the correct prediction (Y0B) for (rob) in our example.

5.5 Training and Decoding

Training for chain-structured CRNs is a special case of the general training of CRNs on acyclic clique graphs which we have developed in Chapter 4.

We use the Viterbi algorithm [50] for finding the assignment which maximizes the joint probability of a trained models. The Viterbi algorithm can be considered as special case of the maximization algorithm described in Section 2.5.1.3. The speciality is that the Viterbi algorithm applies the maximization operation to chain-structured graphs rather than general graphs. Because the maximization operation is commutative and localizable, the Viterbi algorithm over chain-structured graphs can be very efficient, i.e. linear to the size of the chain.

5.6 Experiments

In this section, we use experiments to compare our chain-structured CRN, denoted by L-CRN⁷, with CRFs⁸. We adopt CRF++ version 0.58 [51] as the implementation for CRFs and LIBLINEAR version 1.94 [52] for as the linear regression algorithm for solving parameters in L-CRNs (Section ??). We set the configurations of LIBLINEAR as L2 regularization L2 loss support vector regression (solving dual). For a fair comparison, we always run software with a single thread on the same machine⁹. We apply CRFs and L-CRNs to two important natural language processing applications: named entity recognition (NER) and part-of-speech (POS) tagging.

5.6.1 Named Entity Recognition

The English part of the CoNLL-2003 NER dataset¹⁰ [53] is used for our NER experiment. There are three data files in this dataset: ner.train, ner.testa and ner.testb. The first one is designed for training and the last two are used for testing. The size of the label space is 8. These three files include 14,987, 3,466, 3,684 sentences and 204,567, 51,578, 46,666 words respectively. We use the same orthographic features as those used by Lafferty et al. [41]: “whether a spelling begins with a number or upper case letter, whether it contains a hyphen, and whether it ends in one of the following suffixes: -ing, -ogy, -ed, -s, -ly, -ion, -tion, -ity, -ies”. Additionally, we use the chunk tags and POS tags provided together with the CoNLL dataset.

Table 5.1: Training Time (Seconds) on NER

CRF	L-CRN
1,666	14

Table 5.1 gives the time taken for training by CRF and L-CRN. We can see L-CRN can be trained more than 100 times faster than CRFs.

⁷Here ‘L’ stands for linear.

⁸Lafferty et al. [41] and Hong [46] show superiority of CRFs over other models considered in this chapter. Hence it is reasonable to compare with CRFs which are the state-of-the-art models.

⁹Obviously, L-CRNs can be easily parallelized. Each regression model can be trained parallelly with others.

¹⁰This dataset can be downloaded at <http://www.cnts.ua.ac.be/conll2003/ner/>.

Table 5.2: Metrics on ner.testa

%	All	Known	Unknown	Precision	Recall	F1
CRF	97.00	98.27	85.42	84.66	82.31	83.47
L-CRN	97.44	98.80	85.05	84.21	84.45	84.33

Table 5.3: Metrics on ner.testb

%	All	Known	Unknown	Precision	Recall	F1
CRF	95.00	97.46	80.32	75.61	74.70	75.15
L-CRN	95.55	98.06	80.62	75.78	76.43	76.10

Table 5.2 and Table 5.3 show the quality metrics achieved by CRF and L-CRN on ner.testa and ner.testb, respectively. The first three columns show the per-word accuracies (%) on all, known and unknown words.¹¹ On all and known words, L-CRN consistently outperforms CRF slightly on both test data sets. On unknown words, CRF performs better on ner.testa, but L-CRN performs slightly better on ner.testb. The last three columns give the precision, recall and F1 metrics. These metrics were evaluated using the standard CoNLL evaluation tool¹². CRF obtains better results in precision on ner.testa. But L-CRN obtains better results in recall and F1 on both data sets.

5.6.2 Part-of-speech Tagging

We use the Brown Corpus [54] in our experiments of part-of-speech (POS) tagging. Incomplete sentences not ending with a punctuation are excluded from our experimental dataset. This results in 34,623 sentences. The size of the tag space is 252. Following Lafferty et al. [41], we introduce parameters for each tag-word pair and tag-tag pair. We also use the same spelling features as those used by Lafferty et al. [41]. We split the dataset into two parts: half of sentences, 17,312, for training, and the other half for testing. Table 5.4 reports the experimental results.

From these results, we can see the training time taken by CRNs is more than 100 times less than the time taken by CRFs. This is consistent with the

¹¹Known words are the words that appear in the training data. Unknown words are the words that have not been seen in the training data. All words include both.

¹²This tool can be downloaded at <http://www.cnts.ua.ac.be/conll2000/chunking/conllEval.txt>.

Metric	L-CRN	CRF
Overall	94.2	93.2
Known	96.4	95.3
Unknown	61.0	62.3
Time (s)	1,023	133,048

Table 5.4: POS Tagging Accuracy (%)

results obtained in the NER task. CRNs also achieve better accuracies on overall and known words. This result is also consistent with the results obtained in the NER task. The MLE of CRNs performs better on known words, which is expected because our model can fit the empirical distribution better. But on unknown words, CRFs outperform CRNs on both NER and POS tagging tasks. Maybe this is because the global normalization brings more coherence between factors, which makes CRFs work better on unseen words. Improving coherence between factors of CRNs can be an interesting issue in the future.

5.6.3 Modeling Label Bias

We test LBP on simulated data which are generated following Lafferty et al. [41]. More specifically, we generate the simulated data as follows. There are five elements in the tag space: {R1, R2, I, O, B} and four elements in the observation space: {r, i, o, b}. The designated symbol for both R1 and R2 is r, for I it is i, for O it is o and for B it is b. We generate the paired sequences from two tag sequences: [R1, I, B] and [R2, O, B]. Each tag emits the designated symbol with probability of 29/32 and each of other three symbols with probability 1/32. For training, we generate 1000 pairs for each tag sequence, so totally the size of training dataset is 2000. For testing, we generate 250 pairs for each tag sequence, so totally the size of testing dataset is 500. We run the experiment for 10 rounds and report the average accuracy on tags ($\frac{\#CorrectTags}{\#AllTags}$) in Table 5.5.

L-CRN	CRF	CMM
95.9	95.9	66.6

Table 5.5: Accuracy for Label Bias Problem

Experiment results show that both CRNs and CRFs are unaffected by the label bias problem, but CMMs suffer from this problem. These results verify

our discussion in Section 5.3.

5.7 Summary

In this chapter, we do experiments to test chain-structured CRNs on two important NLP applications, i.e. named entity recognition and part-of-speech tagging. We compared CRNs with conditional random fields, which is a type of conditional Markov network. Experimental results show that CRNs can be trained orders of magnitudes faster than CRFs, and achieve slightly better results on overall and known words. But on unknown words, CRFs obtains slightly worse results than CRNs. We make a conclusion that CRNs can be trained much faster than CRFs, while achieving better or competitive quality on two NLP applications.

Part II

Open Relation Extraction

A Review of Open Relation Extraction

Outline

In this chapter, we first introduce the task of open relation extraction (ORE) in Section 6.1. Then we review ORE systems in two perspectives: (i) quality metrics, and (ii) technical aspects. More specifically, we summarize quality metrics used for measuring the outputs of ORE systems. Quality metrics define the objective of ORE systems. We extract quality metrics used by different ORE systems both implicitly or explicitly, and organize them as a hierarchy multi-dimensional quality model (Section 6.2). ORE systems can also be distinguished in technical aspects (Section 6.3).

6.1 Introduction

Relation extraction plays a central role in transforming free texts to structured data, which enables advanced NLP, IR and web applications. The usefulness of relation extraction techniques is almost axiomatic in the age of big data. Everyday, there are tons of texts posted on the web. Relation extraction techniques can be employed to extract use information from unstructured texts automatically. Relation extraction systems recognize entities in texts, and extract relations between entities. The underlying model which makes relation extraction an important technique is that we believe most part of the world can be represented by entities and relations between entities. Such an entity-relation model is expressive most of the time. In Part II of this thesis, we focus on this interesting and challenging task: extracting relations from texts. The connection between Part II and Part I of this thesis is that we formalize the task of relation extraction with a novel sequence labeling model, which can be considered as an undirected graphical model. In other words, in Part II we provide an application case for the

theory developed in Part I.

There are two paradigms developed for relation extraction: (i) closed relation extraction (CRE), and (ii) open relation extraction (ORE). The difference between them can be concluded as follows. A CRE system extracts relations limited to a pre-specified set of relation types. That is given a type of relation, such as [Married] or [BirthPlace], CRE *recognizes*, i.e. matches or classifies, pairs of entities which hold the given relation. For example, if we input [Married] to a CRE system, it will return a set of pairs of entities, such as ([Bill Clinton], [Hillary Clinton]). A brief review of CRE systems can be found in Section 6.3.9.

In contrast, open relation extraction, which was firstly introduced by [55], does not require a pre-specified set of relation types. It *discovers* relations, i.e., extracts relation expressions, between two entities in free texts. We will give an example in **Example 1**. [56] argues that when the number of relation types are massive or cannot be specified in advance, open relation extraction is needed for exploring unknown relations. For example, in Wikipedia, massive relations exist and they are updated everyday. It is impossible to pre-specify all relation types. In such cases, automatically extracting relation expressions between entities is the key step in exploring unknown relations. ORE systems have become prevalent due to their ability to explore relations of unknown types. This is quite effective if we shift to a new domain. Hence, ORE systems are expected be applied to massive and heterogeneous corpora.

In this thesis, we focus on open relation extraction. Generally, there are two sub-tasks in open relation extraction: (i) entity recognition, and (ii) relation expression extraction. We use the following example to illustrate what a typical ORE system does:

Example 1 [[Lenovo] completes acquisition of [Motorola Mobility] from Google.

1. *Entity recognition.* [Lenovo], [Motorola Mobility] and [Google] are labeled as entities.
2. *Relation expression extraction.* Relation expressions are also called relation phrases, relation descriptors, or surface form relations. In this example, we may extract [completes acquisition of], [acquisition of], or just [acquisition] as the relation expression between the entities [Lenovo] and [Motorola Mobility]. Here we can see the difference between CRE and ORE systems. For an ORE system, we do not need to input [acquisition] as a pre-specified relation type. Instead,

[acquisition] is automatically extracted from texts as a relation expression. Unfortunately, there is no standard answer to which one should be extracted. This is because different ORE systems use different quality metrics which will be described in the next section. According to [56], relation expressions are “salient textual cues that indicate the relation among them”. Salience means relation expressions extracted are informative and simple (concise). In this sense, the relation expression [acquisition] is the best choice. But if we want to build a question answering system or summarization system based on outputs of an ORE system, the coherence is an important metric. With respect to coherence, [completes acquisition of] is the best choice. Hence, the choice of relation expressions is based on the target application.

Most of ORE systems first recognize entities, then extract relation expressions between the recognized entities. The only exception is ReVerb [57], which first detects relation expressions that meet some syntactic constraints, then identifies entities which are close to the relation expression. The entities involved in an extraction are called arguments. According to the number of arguments, extractions can be classified to binary relations, which contain 2 arguments, or n-ary relations which contain more than 2 arguments. In Example 1, we can also extract a n-ary relation: (Lenovo; acquire; Motorola Mobility; from Google), where [acquire] is the relation expression and the others, i.e. [Lenovo], [Motorola Mobility] and [Google], are three arguments of [acquire].

6.2 Quality Metrics

There are two basic questions to be answered for any ORE system: (i) *What* are good relation expressions? and (ii) *How* to extract relation expressions? The answer to the first question is related to the quality metrics for measuring outputs of ORE systems, and the answer to the second question is related to technical aspects of ORE systems. As the first question is more fundamental than the second one, before diving into technical details, which will be described in the next section, we summarize the quality metrics which are used in different ORE systems both explicitly or implicitly. ORE systems may implicitly put multiple dimensions under a general dimension called correctness. This leads to confusions in many cases. An extraction which is deemed correct in one system may be considered incorrect for another system. This is because correctness has

different meanings for different systems. In order to reduce confusions, we need to make metrics more specific. We summarize quality metrics used in difference ORE systems, and put them together to form a hierarchical multi-dimensional quality model as depicted in Figure 6.1 for measuring extractions. We explain each metric as follows.

Figure 6.1: A Hierarchical Multi-dimensional Quality Model for ORE

Faithfulness Extractions should be faithful to facts. This metric applies to all ORE systems because unfaithful extractions are not useful, and may lead to a wrong conclusion. There are two methods to ensure faithfulness: (i) context analysis, and (ii) redundancy-based assessor. OLL-IE [58] proposes context analysis to improve faithfulness of extractions. We borrow the following example to show the importance of context analysis. For the sentence, “Early astronomers believed that the earth is the center of the universe.”, ORE systems, which only focus on local analysis may obtain such an extraction: (the earth, be the center of, the universe), which is not a fact. To alleviate this problem, OLL-IE adds contextual information to its extraction: ((the earth, be the center of, the universe) AttributedTo believe; Early astronomers). CSD-IE [59] is another context aware system. It add contextual information to extractions using references:

#1: (Early astronomers) (believe) (#2)

#2: (the earth, be the center of, the universe)

The reason that CSD-IE uses references to represent context information is that it also considers the minimality metric of extractions, which will be described later. TextRunner [55] assumes that if a tuple is extracted many times from different sentences, it is more likely to be faithful to facts. Hence it adds a

redundancy-based assessor as a post-processing step to filter out extractions of a low frequency.

Informativeness Banko and Etzioni [56] consider relation expressions as “salient textual cues that indicate the relation among them”. We understand ‘salient’ as ‘informative’ and ‘simple’ (or concise). ReVerb [57] further points out the problem that extractions by ORE systems may be uninformative (under-specified). Especially, they consider the phenomenon of light verb construction (LVC). A LVC structure is a compound of a verb and a noun, with the noun carrying the main semantic content. That is the verb is too general (under-specific) to convey useful information. For example, for the sentence, “His hobby gave birth to a successful business”, the extraction (`His hobby; gave; birth`) is not informative. To alleviate this problem, ReVerb applies a simple syntactic constraint to exclude uninformative extractions. This constraint roughly states that: every multi-word relation must begin with a verb, end with a preposition, and be a contiguous sequence of words in the sentence. This constraint is also used to exclude incoherent extractions in ReVerb. Also Banko et al. [55] classifies extractions as either *concrete* or *abstract*. Here *concrete* and *abstract* can be considered as synonyms of informativeness and uninformative, respectively.

Bast and Haussmann [60] improve informativeness via simple semantic inference. For the sentence, “The ICRW is a non-profit organization headquartered in Washington.”, the extraction (`a non-profit organization; headquartered in; Washington`) is uninformative. If we know [`The ICRW`] is a [`a non-profit organization`], a more informative tuple can be obtained by semantic inference: (`The ICRW; headquartered in; Washington`).

Simplicity Texts between two entities may contain non-essential information which leads relation expressions to be over-specific. TextRunner deletes non-essential phrases heuristically (e.g., [`definitely developed`] is reduced to [`developed`]). This metric requires relation expressions to be simple and concise. Also ReVerb [57] excludes over-specific extractions based on redundancy. The assumption underlying this method is that over-specific extractions should not take many distinct arguments in a large corpus.

Coherence We understand coherence as ‘almost grammatical’. Banko et al. [55] first judged whether extractions are *well-formed*, which is a synonym of *coherence*. ReVerb [57] points out that sequence labeling models may output incoherent extractions. For example, for the sentence “The guide contains dead

links and omits sites.”, the extraction (The guide; contains omits; sites) is incoherent. ReVerb [57] excludes incoherent extractions by a simple syntactic constraint as discussed before.

Minimality The quality metric minimality was first proposed by Bast and Haussmann [59]. The minimality metric indicates if the extracted triple can be further decomposed into smaller meaningful triples. The outputs of their ORE system are used to enhance a formal ontology. Minimality is an important metric for such an application.

Recall This is an obvious metric. ORE systems try to extract as many correct tuples as possible from a given corpus. But on the open relation extraction task, determining the absolute recall is usually precluded by the amount of hand labeling necessary and the ambiguity of such a task. Relative recall (also called pseudo-recall) which is computed by taking the union of correct tuples from different systems is commonly used as the alternative [61].

Scalability ORE systems emphasize on the Web scalability. Scalability is an important consideration in the age of big data. Processing speed on a single sentence can affect scalability. Also big data frameworks, e.g. MapReduce, can significantly improve scalability.

Different applications emphasize on different quality metrics. For example, for question answering systems [62] or summarization systems, coherence is an important metric, but minimality is not that important. In contrast, to support a ontology knowledge base or a full-text semantic searcher [59], coherence seems not that important, but minimality is an important metric. We also argue that the general metric *correctness* is not specific enough to measure the quality of extractions. More specific metrics are needed to define what is correctness. By doing this, we can reduce ambiguity. For example, the extraction (Bunsen; invented; a device) from the sentence “Bunsen invented a device called the Spectroscope” is coherent, but not informative comparing with the extraction (Bunsen; invented; Spectroscope). We can not simply say the extraction is correct or incorrect.

6.3 Technical Aspects

In this section, we review different techniques for extracting relation expressions. These techniques can be distinguished by the following dimensions.

Extraction vs Inference Most ORE systems only extract relation expressions which are explicitly expressed in texts. We called such systems explicit ORE systems. Let $r(e_i, e_j)$ be a relation, where r is a relation expression, and $\{e_i, e_j\}$ are two entities. Explicit ORE systems extract $r(e_i, e_j)$ only when r and (e_i, e_j) appear in the same sentence. In contrast, Riedel et al. [63] proposes a model for predicting the probability of $r(e_i, e_j)$ no matter r and $\{e_i, e_j\}$ ¹ appear in the same sentence or not. The sample space of all relation expressions (called universal schemas in the paper), denoted by $\mathcal{R}$, can come from different sources, such as a pre-defined knowledge base (e.g. Freebase [64]), or outputs of an explicit ORE system. A random variable $Y_{r,t}$ is set for each $r(e_i, e_j)$, where t is an abbreviated symbol of (e_i, e_j) . Thus there are $|\mathcal{R}| \times |\mathcal{E}| \times |\mathcal{E}|$ random variables, where $\mathcal{E}$ is the set of all entities under consideration. Here $r(e_i, e_j)$ is distinguished from $r(e_j, e_i)$, because they are set with different random variables, which implies this model is asymmetric. The distribution $P(Y_{r,t})$ is modeled by a logistic function:

$$P(Y_{r,t}) = \frac{1}{1 + \exp(-\theta_{r,t})}$$

where $\theta_{r,t}$ is a parameter which can be further decomposed into three parts: latent feature model, neighborhood model and entity model. In the language of recommendation system, this decomposition can be formalized as a matrix factorization: a matrix with entries filled with $\theta_{r,t}$ is decomposed into a product of three matrices. Training is performed to maximize the difference between positive and corresponding negative instances for all observed instances. This model can be used as a post-processing step to improve the precision and recall of explicit ORE systems. Especially, we expect recall can be improved significantly because many implicit relations can be predicted by this model.

Shallow vs Deep According to Mesquita et al. [65], ORE systems can be grouped according to the level of sophistication of the NLP tools they employ: (i) shallow parsing, e.g. POS tagging; (ii) syntactic dependency parsing; (iii) semantic role labeling. ORE Systems depending on shallow parsing tools can

¹In their model, binary relations can be easily extended to n-ary relations.

extract relations orders of magnitude faster than systems based on deep parsing tools. Hence, ORE systems using shallow parsing tools show their advantage when scalability is a critical consideration. But on the other hand, recall can be lower for such systems.

Manual vs Automatic The training dataset can be obtained manually or automatically. Manually annotating a dataset could be very expensive. Hence, automatic methods such as bootstrapping and distant supervision (also called self-supervision) get popular. WOE [66] compiles a training dataset from Wikipedia by the bootstrapping method. Triples including entities and relation expressions are extracted from Wikipedia infobox. Then for each triple, they find a unique sentence (if exists) in the same article by heuristically matching entities and relation expressions. Distant supervision [67] assumes that: if two entities involved in a relation, then all sentences that mention these two entities express that relation. The difference between the bootstrapping method and distant supervision is that the bootstrapping method also matches the relation expression in a sentence while distant supervision only matches entities. Riedel et al. [68] releases the assumption of distant supervision to: if two entities involved in a relation, at least one sentence which mentions these two entities express that relation.

Rule-based vs Probabilistic For rule-based systems, a set of extraction patterns are first compiled manually. Then these rules are applied to sentences to extract relations. In contrast, for probabilistic systems, extraction patterns are predicted by probabilistic models trained by data.

Binary vs N-ary All ORE systems are able to extract binary relations. Some of them can also extract n-ary relations.

Verbs vs Others Based on the analysis of 500 randomly selected sentences, Banko and Etzioni [56] claim that about 95% of binary relations are expressed using a small set of relation-independent lexico-syntactic patterns. This indicates that ORE systems are technically feasible. By a investigation of the lexico-syntactic patterns given by Banko and Etzioni [56], we can find that most of these patterns are verb-mediated patterns, and others are noun-mediated patterns. Most ORE systems can extract verb-mediated relations. Some of them may also extract relations expressed in other lexico-syntactic patterns.

In the remainder of this section, we review a set of state-of-the-art ORE systems in their technical aspects.

6.3.1 TextRunner

TextRunner [55] is the pioneering system² to demonstrate the ORE paradigm. TextRunner consists of three components: an extractor, a classifier and an assessor.

Extractor The input of the extractor is a sentence and the output is a tuple $t = (e_i, r_{i,j}, e_j)$, where e_i and e_j are two entities and $r_{i,j}$ is a relation expression. A part-of-speech (POS) tagger is employed as a pre-processing step. The entities, e_i and e_j , are recognized by identifying noun phrases with POS tags. The relation expression is formed by examining the text between e_i and e_j and normalizing the text by deleting unimportant words heuristically.

Classifier The tuple $t = (e_i, r_{i,j}, e_j)$ output by the extractor is then input to a classifier. The classifier outputs a boolean value which indicates whether the input relation is trustworthy. If the judgement is positive, the relation will be stored by TextRunner. Otherwise, the relation will be discarded. TextRunner uses a naive Bayes classifier, which is essentially a Bayesian network for classification. The training data is labelled by the distant supervision method. For each sentence in the training data, it employs the Stanford parser to obtain the dependency³ graph representation of the sentence. Then a set of manually compiled rules are applied to judge if the extraction is trustworthy. These rules use the features extracted from the dependency graph and POS tags.

Assessor After the classification, TextRunner collects a set of positive tuples. An assessor further evaluates the confidence for each tuple based on redundancy. $t = (e_i, r_{i,j}, e_j)$ is a correct relation given that it was extracted from k different sentences.

Banko and Etzioni [56] replace the Naive Bayes (NB) model used by Banko et al. [55] with CRFs. Similar shallow syntactic features, such as POS tags, are used for training CRFs. For positive examples, a relation expression is extracted

²Preemptive IE developed independently by Shinyama and Sekine [69] also proposed the idea of avoiding relation-specificity.

³Here, dependency means linguistic dependency which is completely different from the concept of the probabilist dependency described in Part I.

by the CRF model. This model boosts F1-measure by 63% comparing with the naive Bayes model. The source code and datasets of TextRunner are included in the OpenIE project which is available online⁴. The OpenIE project also contains implementations of other systems, such as SRL-IE, ReVerb, R2A2 and OLL-IE, which will be discussed as follows.

6.3.2 SRL-IE

In contrast to TextRunner which utilizes shallow syntactic features, such POS tags, SRL-IE [61] is based on the outputs of a semantic role labeling (SRL) system, i.e. UIUC-SRL. Semantic role labeling system requires expensive deep syntactic parsing. The key observation of SRL-IE is that semantically labelled arguments almost always correspond to the arguments in an ORE extraction, which shows the tight connection between these two NLP tasks. SRL-IE converts outputs of UIUC-SRL to binary or n-ary relations by a set of rules.

6.3.3 WOE

Wikipedia-based open extractor (WOE) [66] has two versions which are denoted by WOE^{parse} and WOE^{pos} , respectively.

The WOE^{pos} system uses the same shallow syntactic features as those used by TextRunner. And WOE^{pos} also employs CRFs. The only significant difference between WOE^{pos} and TextRunner is in how they construct training data. WOE^{pos} automatically builds training examples by the bootstrapping method. This results in 301,962 sentence-triple pairs. These pairs are used as positive examples for training WOE^{pos} . For training WOE^{parse} , they apply the Stanford dependency parser to extract the shortest dependency path between a pair of entities. Then 15,333 distinct extraction patterns are identified automatically based on these shortest paths.

6.3.4 ReVerb and R2A2

ReVerb [57] observes that TextRunner and WOE may output uninformative or incoherent extractions. ReVerb introduces a simple syntactic constraint to excluding uninformative or incoherent extractions. R2A2 [70] further employs machine learning methods to determine the boundaries of arguments in a relation, which alleviates incoherent and uninformative relation expressions.

⁴<http://openie.allenai.org/>

6.3.5 OLL-IE

Mausam et al. [58] present OLL-IE (open language learning) for open relation extraction. OLL-IE collects a data set consisting of (sentence, tuple) pairs with the bootstrapping method. The outputs of ReVerb, which are relation tuples, are used as seeds by OLL-IE. It collects sentences which includes all components in a relation tuple. Then OLL-IE applies dependency parser to obtain the dependency path for each sentence. Extraction patterns are automatically learned from these pairs. The learned extraction patterns can be classified into two types: purely syntactic patterns which do not use lexical features, and semantic/lexical patterns which rely on some lexical specific features. As we mentioned in Section 6.2, OLL-IE also proposes context analysis, which improves faithfulness of extractions.

6.3.6 ClausIE

The claus-based IE system (ClausIE) [71] first decomposes a sentence into coherent clauses. This is done by examining the dependency graph of a sentence. Relative clauses and clauses connected by coordinated conjunctions, such as 'and' or 'or', are also considered. Then the identified clauses are classified into 7 general types. This categorization is mainly based on the properties of the central verb of a clause. For example, transitive verbs (V) require a subject (S) and a direct object (O). A clause with a transitive verb is of the SVO type. In contrast, intransitive verbs (V) only require a subject (S). Objects are not mandatory. Other constituents are optional. A clause containing a intransitive verb is classified as the type of SV. The type of a clause indicates whether a constituent in the clause is essential or optional in a relation extraction. Finally, based on the types of clauses, relation extractions are generated correspondingly.

6.3.7 EXEMPLAR

EXEMPLAR [65] applies a set of rules to the dependency graph of a sentence to extract binary and n-ary relations. These rules are derived from a careful study of all dependency types generated by the Stanford dependency parser. They claim 86% of n-ary relations are covered by six syntactic patterns. Through their analysis, three relation types were found. EXEMPLAR is then developed specifically to recognize these relation types as well as their variations.

6.3.8 CSD-IE

Bast and Haussmann [59] present contextual sentence decomposition (CSD-IE) for open relation extraction. The basic idea of CSD-IE is very similar to ClausIE. CSD-IE decomposes a sentence into several contexts which contain independent information of the sentence. The concept of contexts used within CSD-IE is similar to the concept of clauses used within ClausIE. After the decomposition step, relation triples are extracted from these contexts. This system has been developed independently of ClausIE. Bast and Haussmann [59] also propose the minimality as an important metric for measuring quality of extractions.

6.3.9 Closed Relation Extraction

As mentioned in Section 6.1, closed relation extraction formulates the relation extraction task as a supervised binary classification problem. Both syntactic and semantic features have been used for classification. Particularly, kernel methods [72, 73, 74] have been used to model the similarity between structural objects, such as string sequence, parse tree and dependency path. One important observation by Bunescu and Mooney [74] is that the shortest dependency path between two entities encodes sufficient information to extract the relation between them. The recall of their system, which is based on the dependency path kernel, is significantly better than tree based kernels [72, 73] while obtaining comparable precision. Bach and Badaskar [75] give a more comprehensive review on this topic.

SimpleIE: a Simplification Model for Open Relation Extraction

The contents of this chapter will be submitted to the journal Transactions of the Association for Computational Linguistics. The source code of SimpleIE, data sets and experimental outputs that appear in this chapter are all available upon request¹.

Outline

In this chapter, we develop our model SimpleIE for extracting relation expressions between two entities². SimpleIE targets at providing a general model which can represent diverse extraction patterns. Also SimpleIE is a data-driven model which can automatically learn extraction patterns from data.

This chapter is organized as follows. In Section 7.1, we use real-world examples to show that diverse extraction patterns can be represented with a sequence of simplification operations, including insertion, deletion, splitting, reordering, and substitution. These examples motivate us to represent extraction patterns as a sequence of simplification operations in SimpleIE. Section 7.2 provide details of our model. SimpleIE is essentially a sequence labeling model. The label space consists of simplification operations. In Section 7.3, we manually annotate a data set which is compiled from Wikipedia for open relation extraction. Experiments are reported in Section 7.4. Experimental results show that with proper training data sets, SimpleIE boosts recall and F1-score by at least 15% on all three benchmark data sets meanwhile obtaining very competitive precision.

¹The source code, datasets and outputs can be requested by sending a message to (zhuzhemin@gmail.com).

²Currently, SimpleIE can only work on binary relations. We leave extracting n-ary relations to future work.

High recall scores show that SimpleIE is a general model which can represent and capture diverse extraction patterns in training data. In Section 7.4.5, we compare CRNs with CRFs by embedding them as the sequence labeling model in SimpleIE. Experimental results show that CRNs can be trained orders of magnitude faster than CRFs while obtaining equal quality.

7.1 Motivated by Examples

In this section, we give examples to illustrate that an informative and coherent relation expression can be extracted by applying a sequence of simplification operations to the dependency path between two entities. Simplification operations, including insertion, deletion, splitting, reordering, and substitution, are widely used in sentence simplification. In sentence simplification, a long and complex sentence is re-written into several short and simple sentences by simplification operations (See Section 7.6 for a brief introduction of sentence simplification.). A dependency path is the shortest path³ between two nodes (representing two entities) in the dependency graph of a sentence. A sentence can be represented as a dependency graph. Each node of a dependency graph represents a word in the sentence, and each edge of the graph represents a (linguistic) dependency relation, such as subject, direct object and auxiliary, between the two words attached to the edge. Dependency graphs can be constructed by some off-the-shelf tools, e.g., the Stanford CoreNLP [76].

Previous works [66, 58] have shown dependency paths are valuable for open relation extraction. Even though a dependency path contains valuable information for relation extraction, we together with previous works [56, 57] observe that in most cases, simply extracting words from a dependency path will result in a unsatisfactory relation expression. The reasons can be concluded as follows. on one hand, mandatory or important components of a relation may not appear in the path, which need to be *inserted* (see Example 1). On the other hand, dependency paths may contain unimportant, redundant or irrelative information to a relation, which needs to be *deleted* (see Example 2). Moreover, multiple relations may appear together in one path. In this case, we need to *split* the path to into multiple relations (see Example 5). Also sometimes words in a path do not appear in the correct order, which should be *reordered* (see Example 6). Finally, sometimes to obtain a meaningful relation expression we need to *substitute* words in the path with other words (see Example 3).

³There may be more than one shortest paths between two nodes. In this case, we randomly select one of them.

In the modeling aspect, SimpleIE targets at providing a general model which can be used to represent diverse extraction patterns. Most of previous works focus on one or several specific kinds of extraction patterns. And once these patterns are compiled manually or learned from data, they are fixed. In contrast, SimpleIE decompose an extraction pattern into components, i.e., a sequence of simplification operations. These components can be re-combined to generate new extraction patterns with respect to observed features. The O-CRF model proposed by Banko and Etzioni [56] also represents a extraction pattern as a sequence labeling model. But O-CRF essentially only considers the deletion operation. This can be observed from its label space which consists of three labels: B (begin of the relation expression), I (in the relation expression) and O (out of the relation expression). O-CRF assigns these BIO labels to each word of the text between two nouns in a sentence. The words with label *O* will be deleted and words assigned with other labels form the relation expression. From the examples given in this section, we can see that a wide range of extraction patterns cannot be represented with a sequence of only deletion operations. In other words, deletion operation is not sufficient to model a wide range of extraction patterns. In this sense, SimpleIE, which considers insertion, deletion, splitting, reordering, and substitution operations, extends O-CRF to more rich operations. Another difference between O-CRF and SimpleIE is that O-CRF applies operations to the text between two nouns in the sentence. In contrast, SimpleIE applies operations to the dependency path between two entities in the dependency graph. Words which are out of the text between two nouns in a sentence may have tight dependency relations with the words in the text segment. O-CRF will miss these words. In contrast, benefited from the dependency graph representation, SimpleIE can potentially survive these words.

Example 1 *On November 4, 2008, Obama delivered victory speech in Chicago's Grant Park.*

Figure 7.1 depicts the (Stanford) dependency graph⁴ representation of the sentence given in Example 1. For the sake of readability, we give a brief explanation of the dependency relations which appear in this example. A comprehensive description of Stanford dependencies is provided by Marneffe and

⁴Originally, for each word in the sentence, there will be a corresponding node in the dependency graph. To simplify the dependency graph, we merge the nodes containing words that belong to the same noun phrase to one node. For example, the words in [Chicago's Grant Park] are put in one node.

Figure 7.1: Dependency Graph of Example 1

Manning [77].

- nsubj: nominal subject
- dobj: direct object
- prep: preposition
- nn: noun compound modifier

We examine the shortest path between [Obama] and [Chicago's Grant Park] in Figure 7.1. Directly extracting words on this path results in a uninformative relation expression: ([Obama]; [delivered in]⁵; [Chicago's Grant Park]). The problem is the direct object [speech] does appear in this path. To obtain a complete relation expression, we need to *insert* the direct object to the relation expression. That is to apply an insertion operation (label: *insert_dobj*, see Section 7.2.4) to the node [delivered]. By doing this, we obtain the relation expression as we expect: ([Obama]; [delivered speech in]; [Chicago's Grant Park]), where the order of insertions is judged according to the word order in the original sentence.

Previous works, such as WOE [66] and ReVerb [57], also notice the necessity of the insertion operation. But WOE only focuses on a specific case of direct object. In many other cases, we also need the insertion operation, such as completions and noun modifiers. And also WOE does not consider other kinds of operations which are considered by SimpleIE. ReVerb points out that WOE may miss complements to LVCs (light verb constructions), which can result in

⁵The preposition is added by default.

uninformative extraction. To alleviate this problem, ReVerb imposes a simple syntactic constraint to extractions. But such a hard constraint can filter out a lot of good extractions and results in low recall. Also notice that LVCs are lexically specific. Hence we need lexical level features. That is some verbs are LVCs, but others are not. They should be treated differently. For LVCs, we need to insert their complements. But for non-LVCs, we do not insert completions. We give the following examples to illustrate this.

Obama on Tuesday assigned federal appellate Judge Sonia Sotomayor to the Supreme Court.

The extraction ([Obama]; [assigned]; [Sonia Sotomayor]) is not that informative because the verb [assigned] does not bear the main semantics. In other words, the verb is too light (general) to convey useful information. We prefer the other extraction ([Obama]; [assigned to the Supreme Court]; [Sonia Sotomayor]). But for another sentence:

Obama won the 2004 General Election with 70 percent of the vote.

The extraction ([Obama]; [won]; [2004 General Election]) is informative by itself even without the completion phrase [with 70 percent of the vote]. To distinguish these two cases, i.e. [assigned] and [won], lexical level features are required.

Example 2 *Obama was re-elected president in November 2012, defeating Republican nominee Mitt Romney.*

Figure 7.2: Dependency Graph of Example 2

- vmod: verbal modifier

- aux: auxiliary

In this example, we extract relations from three shortest paths as follows.

1. For the shortest path between [Obama] and [Mitt Romney], we *delete* the node [re-elected] and extract the relation ([Obama]; [defeating]; [Mitt Romney]). In other words, we apply a deletion operation (label: delete) to the node [re-elected]. The necessity of the deletion operation in extracting relations has also been noticed by previous works, such as ReVerb [56].
2. There is a noun-mediated relation existing between [Republican nominee] and [Mitt Romney]: ([Mitt Romney]; [be]; [Republican nominee]). To extract this relation expression, we need to *insert* the copula verb [be] between [Republican nominee] and [Mitt Romney]. That is an insertion operation (label: insert_‘be’) is applied to the node [Mitt Romney]. To our knowledge, there is no previous work which explicitly points out the necessity of inserting additional words into a relation expression. But by checking the output of OpenIE 4.1 (<http://openie.allenai.org/>) and ClausIE [71], we find they do insert some additional words which have not appeared in the sentence to the relation expressions.
3. There is another relation existing between [Obama] and [president]: ([Obama]; [was re-elected]; [president]), which can be obtained by applying an insertion operation (label: insert_aux) to the node [re-elected].

Example 3 *As First Lady, Clinton hosted numerous White House conferences.*

Figure 7.3: Dependency Graph of Example 3

- pobj: object of a preposition

- amod: adjectival modifier

In this example, we examine the shortest path between [Clinton] and [First Lady]. From this path, we expect to extract the following relation: ([Clinton]; [be]; [First Lady]). Two operations are needed to achieve this relation. The first operation is a deletion operation applied to the node [hosted], which has been discussed in the previous example. The second operation is a *substitution* operation applied to the node [As]. This substitution operation (label: substitute_‘be’) replaces the node [As] with the word [be]. To the best of our knowledge, we are the first to use the substitution operation.

Example 4 *Obama’s mother, Stanley Ann Dunham, was born in Wichita, Kansas.*

Figure 7.4: Dependency Graph of Example 4

- auxpass: passive auxiliary

In this example, (i) [Stanley Ann Dunham] is an appositive of [Obama’s mother]. From the path between them, by applying an insertion operation (insert_‘be’) to the node [Obama’s mother], we obtain the following relation between these two nodes: ([Obama’s mother]; [be]; [Stanley Ann Dunham]). (ii) By applying an insertion operation (label: insert_auxpass) to the node [born], we obtain ([Obama’s mother]; [was born in]; [Wichita, Kansas]) from the two nodes. (iii) Additionally, if we apply a deletion operation (label: delete) to the node [Obama’s mother] and apply an insertion operation (label: insert_auxpass) to the node [born], we obtain the third relation: ([Stanley Ann Dunham]; [was born in]; [Wichita, Kansas]) from the path between [Stanley Ann Dunham] and [Wichita, Kansas].

Example 5 *Obama worked as a attorney and taught constitutional law at University of Chicago from 1992 to 2004.*

Figure 7.5: Dependency Graph of Example 5

- conj: conjunction

In this example, we examine the shortest path between [Obama] and [University of Chicago]. This path contains two verbs [worked] and [taught]. They are connected by a conjunction [and], which indicates there are two relations existing together in one path. We need to *split* the path into two parts to obtain two relations. This can be done by applying a splitting operation (label: split) to the node [taught] to obtain two paths. Then from these two paths, we further extract the two relations that we expect: ([Obama]; [worked at]; [University of Chicago]) and ([Obama]; [taught law at]; [University of Chicago]). See Section 7.2.4 for actions defined for a split operation.

Example 6 *Sonia Sotomayor, nominated by Obama on May 26, 2009, was confirmed on August 6, 2009.*

Figure 7.6: Dependency Graph of Example 6

For the path between [Sonia Sotomayor] and [Obama], we expect to extract

an active voice relation expression⁶:([Obama]; [nominated]; [Sonia Sotomayor]). To obtain this relation, we need to completely *reorder* the nodes in the path⁷. That is to apply reorder operations (label: *reorder*) to both nodes [nominated] and [Obama]. See Section 7.2.4 for actions defined for a reorder operation.

Our experience of annotating the Wikipedia data set (described in Section 7.3) shows that almost all of paths in the data can be transformed to informative and coherent relation expressions by applying a sequence of simplification operations to nodes in the path. In other words, almost all extraction patterns can be represented with a sequence of simplification operations including operations of deletion, insertion, splitting, substitution and reordering.

7.2 The Model: SimpleIE

In this section, we describe the details of the model SimpleIE. We have illustrated the idea of SimpleIE by examples in the previous section. The core part of SimpleIE is a sequence labeling model which assigns simplification operations to each node in a dependency path.

Input The inputs of SimpleIE are:

1. The dependency graph of the sentence from which we extract relations. We use the Stanford dependency parser in SimpleIE because this parser performs well and provides good documents with the software. But SimpleIE does not depend on a specific parser. Other parsers can be used to replace Stanford parser in SimpleIE if they show advantages over Stanford parser.
2. Two given entities in the sentence which correspond to two nodes in the dependency graph. Entities can be recognized by the method described in Section 7.5. But entity recognition is not the core part of SimpleIE. SimpleIE focuses more on extracting informative and coherent relation expressions between two given entities. Also currently, SimpleIE does not support co-reference resolution. Hence, the entities are allowed to be pronouns. SimpleIE will find the shortest path between these two given

⁶Normally, active voice is considered simpler than passive voice. This also benefits normalization of expressions.

⁷The source and target of a path is decided by their relative order in the original sentence.

entities in the dependency graph, and extract relation expression from the path.

Output The output of SimpleIE is NULL if there is no relation existing between the two given entities in a sentence. Otherwise, SimpleIE returns a relation expression. We give the following example to illustrate this.

Example 7 *Steve Ballmer was succeeded by Satya Nadella on February 4 , 2014 , who previously led Microsoft 's Cloud and Enterprise division .*

For this sentence, if the two entities are [Satya Nadella] and [Microsoft 's Cloud and Enterprise division], SimpleIE is expected to output ([Satya Nadella]; [led]; [Microsoft 's Cloud and Enterprise division]). If the two given entities are [Steve Ballmer] and [Microsoft 's Cloud and Enterprise division], SimpleIE is expected to output NULL.

Core steps SimpleIE consists of three core steps:

1. A binary classifier judges if there is a relation existing between two entities. The input is the shortest path between two entities. The output is 'True' if there is a relation. Otherwise, it outputs 'False'. SimpleIE uses libSVM [78] as the binary classifier.
2. If the output of the first step is positive, we go on to the second step. In this step, SimpleIE uses a sequence labeling model to assign operation labels, such as insert_ 'be' or delete described in previous examples, to each node in the path. According to these operation labels, relation expressions can be extracted by following the pre-defined actions of operations (Section 7.2.4). We try both CRNs⁸ developed in the Part I of this thesis as well as CRF++⁹ as the sequence labeling model in our experiments.
3. As described in Section 7.2.4, the insertion operation has two kinds of parameters. The first kind is to insert a string. For example, in Example 2, the string [be] is inserted to obtain the relation ([Mitt Romney]; [be]; [Republican nominee]). This kind of insertion operation is denoted by a label: insert_ 'be', where 'be' is a parameter of string type. The other kind is to insert an adjacent dependency edge. For example, in Example

⁸<https://github.com/zheminzhu/Co-occurrence-Rate-Networks>

⁹<http://taku910.github.io/crfpp/>

1, the edge [dobj] is inserted to obtain the relation ([Obama]; [delivered speech in]; [Chicago's Grant Park]). This operation is denoted by the label: insert_dobj, where dobj is a parameter of edge type. SimpleIE treats these two kinds of insertion operations differently. The first kind of insertion operations are treated in the same way as other kinds of operations in the second step. The second kind of insertion operations are treated separately in this step. For each adjacent edge of a node in the path, SimpleIE uses a binary classifier to judge if the edge should be inserted or not. Again, the libSVM is employed for this task.

In the following three sub-sections, we detail these three steps.

7.2.1 Step 1: Is There a Relation?

In this step, SimpleIE judges if there exists a relation between two given entities. This can be formulated as a binary classification task. The input are features extracted from the dependency path and the surrounding information in the graph. The output is 'True' or 'False'.

We adopt the SVM (Support Vector Machine) as the classifier. SVM holds some nice properties. In SVM, the original feature space is implicitly transformed to a high-dimensional (which even can be infinite) feature space using kernels. In this high-dimensional space, SVM finds a linear separator (hyper plane) with a maximum margin, while the sum of errors (wrongly sided nodes) with respect to the margin is bounded to a constant. An attractive property of SVM to our task (also to many other NLP tasks) is that only the points close to the boundary play an important role in the optimization process, while other points are kept away as safe as possible. We extract the following features from the dependency path as the input of the SVM model:

1. The length of the path.
2. Unigram directed dependencies. For example, for the path between [Obama] to [Mitt Romney] in Figure 7.2, we extract the following unigram directed dependencies: (nsubj↑), (vmod↓), and (dobj↓). The symbol ↑ represents the direction from source to target, and ↓ represents the direction from target to source.
3. Bigram directed dependencies. For example, for the path between [Obama] to [Mitt Romney] in Figure 7.2, we extract the following bigram directed dependencies: (nsubj↑ vmod↓) and (vmod↓ dobj↓).

4. Trigram directed dependencies.
5. Bigram POS (part-of-speech) tags.
6. Trigram POS tags.

To avoid endowing unwanted metric and order structures¹⁰ to a categorical feature, we use the dummy coding as the representation of categorical features.

7.2.2 Step 2: Assigning Operation Labels to Nodes

7.2.2.1 Operation Labeling

If the output of Step 1 is positive, we go on to assign operation labels to nodes in the path to extract a relation expression. This is essentially a sequence labeling model. After operation labels are assigned, SimpleIE executes corresponding actions as defined in Section 7.2.4 to extract relation expressions. Using the features extracted from the dependency path and its surrounding information in the graph, a sequence labeling model, such as CRNs or CRFs, is employed to assign operation labels to each node in the path. If there is no simplification operation applied, the node will be labeled with NULL. This means SimpleIE just needs to copy the word of the node to the relation expression.

The label space of operations is described in Section 7.2.4. Especially, for the operations with string parameters, e.g., `insert_‘str’` and `substitute_‘str’`, the size of their label space depends on how many different strings are inserted or substituted in the training data set. We find that the strings inserted or substituted are limited to a small set. On our Wikipedia data set, there are 11 insertion strings: [from], [be], [of], [in], [on], [be from], [during], [have], [,] and [be born], and [‘s]. Also substitution strings are limited to [served as->be], [rather than->not] and [as->be].

We use the following features for training the sequence labeling model:

1. The dependency of the edge before current node. If current node is the first node in a path, we assign NULL to be the value of this feature.
2. The dependency of the edge after current node. If current node is the last node in a path, we assign NULL to be the value of this feature.

¹⁰For example, we do not want there is a distance measurement between `nsubj` and `dobj`. We also do not want there is an order between them. If we do not use dummy coding, the metric and order structures will be implicitly introduced to the feature space.

3. The lemma of the current node.
4. The POS tag of the previous node.
5. The POS tag of the current node.
6. The POS tag of the next node.
7. The trigram POS tags of the previous, current and next nodes.
8. The bigram directed dependencies of the two adjacent edges of the current node on the path.

7.2.3 Step 3: Inserting Dependency Edges

In this step, for each node in the path and for each adjacent edge, which does not appear on the path, of the node, SimpleIE judges if the edge should be inserted or not. This can be formalized as a binary classification task. In Example 1, we insert the edge *dobj* into the path between [Obama] and [Chicago's Grant Park]. This results in the relation: ([Obama]; [delivered speech in]; [Chicago's Grant Park]). Note that inserting an edge means inserting the beginning node of the edge, e.g., [speech] is the beginning node of the edge *dobj*, into the relation.

For each kind of dependency edges, we train a SVM model to decide if an edge should be inserted or not. For Stanford Dependencies (version 3.5.1)¹¹, there are 48 different types of dependencies. Hence, we train 48 separate SVM models for each dependency. We extract the following features for SVM models:

1. The POS tag of the current node, which is also the end node of the edge to be judged.
2. The POS tag of the start node of the edge.
3. The edge before the current node on the path.
4. The edge after the current node on the path.
5. The pair of lemmas of the edge to be judged.

¹¹The latest manual can be found at http://nlp.stanford.edu/software/dependencies_manual.pdf

7.2.4 Labels and Actions of Operations

After we obtain an operation label to a node, pre-defined actions of this operation label will be applied to the node to extract texts and form a relation expression. Table 7.1 shows operations and their corresponding labels.

Operation	Labels
insert	insert_‘str’ or insert_dep
delete	delete
reorder	reorder
split	split
substitute	substitute_‘str’
null	null

Table 7.1: Operations and Their Labels

As we discussed, insert operations have parameters attached to their labels, which tell what should be inserted. There are two kinds of parameters: string (insert_‘str’) and dependency (insert_dep). insert_‘str’ means the string ‘str’ will be inserted, and insert_dep means the node attached to the dep will be inserted. See insert_‘be’ and insert_aux in Example 2. Similarly, a substitution operation also has a string parameter ‘str’, which says the current node should be replaced by ‘str’. The number of elements in their (insertion and substitution) label spaces is decided by the training data. Other operations have no parameters. Hence there is only one element which indicating the type of operation in their label. The action of the deletion operation indicates the text of the current node will not be used in constructing the relation expression. And the null operation indicates the text of the current node will be simply copied to the expression.

If we encounter a split operation, the path will be duplicated into two paths. The split node will only appear in one of them. Its previous node will appear in the other path. Then two relations will be extracted separately from these two paths.

If the current node is assigned with a reordering operation, this node will be put before its previous node in the original path. We call this partial reordering. Figure 7.7 gives an example to illustrate partial reordering. Partial reordering is a subset of full (arbitrary) reordering. We found partial reordering is sufficient for our task. More importantly, partial reordering can be represented as sequential labels.

Figure 7.7: Example of Partial Reordering

7.3 Wikipedia Dataset

We manually annotate a data set for open relation extraction. To this end, we would like to contribute it to the community for fostering research on this interesting and challenging topic. Our dataset consists of sentences compiled from the following English Wikipedia pages: ‘Barack Obama’, ‘Hillary Clinton’, ‘Google’, ‘Microsoft’, ‘London’, and ‘Harvard University’. Wikipedia markups are cleaned by the tool JWPL [79]. Wikipedia articles are tokenized and split into sentences using the Stanford CoreNLP [76]. We select sentences which meet the following two constraints for annotating:

1. There is no duplicate words in the sentence. This is an engineering consideration. The reason is that we need to translate the annotation of relations in text into sequences of simplification operations. These sequences of simplification operations can be used for training the sequence labeling model. To avoid disambiguation in the translation process, we simply impose this constraint. Note that this constraint only required in training step. In the decoding step, any sentences can be processed.
2. The length of the sentence is upto 35 words. Because for sentences which are longer than 35 words, errors happen more frequently in parsing the sentence to construct a dependency graph. These errors will trouble the training according to the trash-in-trash-out principle.

Briefly, the principle to construct our data set is that we want to avoid errors in the dataset as much as possible. We also develop a web interface (Figure 7.8) to

ease the annotation.

He worked as a civil rights attorney and taught constitutional law at the University of Chicago Law School from 1992 to 2004 .

Wikipedia: Barack Obama

Para: 0 Sent: 3 Source: 0 1 Target: 9 11

Source

He (Barack Hussein Obama)

Target

constitutional law

Relation

He; taught ; constitutional law

Temporal

from 1992 to 2004

Skip

Null

Prev

Next

41

/ 4808

Go

Figure 7.8: Annotation Interface

For a given sentence, we first manually select two entities appearing in the sentence. An entity can be a named entity, a pronoun, or a noun phrase. Then we manually extract a relation expression for these two entities. We are keeping the quality metrics described in the previous chapter in mind when we are doing the annotation. More specifically, annotations are expected to be: faithful, informative, coherent, simple and minimal. When we encounter some dilemmas, we just skip the annotation. The principle under our annotation is that we only produce annotations for which we have high confidence.

#Sent.	#Path	Avg.	Min.	Max.	#Pos.	#Neg.
266	925	3.36	1	10	557	368

Table 7.2: Statistics of the Wikipedia Dataset

Table 7.2 lists basic statistics of the annotated data set. The number of sentences in the data set is 266. There are 925 paths or entity pairs to be considered for relation extraction. The average length of these paths is 3.36, i.e., the number of edges in the path, with maximum length of 10, and minimum length of 1. Among these 925 paths, 557 of them are positive examples, and the rest 368 are negative examples which means there is no relation existing for these paths.

7.4 Experiments

Our experiment methodology follows the method used by Mesquita et al. [65]. Mesquita et al. [65] provide five benchmark data sets. Three of them are manually annotated for testing binary relation extractions. These are suitable data sets for testing our SimpleIE. The fourth data set is for testing n-ary relation extractions. As at current stage, SimpleIE does not support n-ary relation extraction, we do not use this data set. The last data set is generated automatically by machine. As the quality of this dataset is not as good as human annotations, we do not use this data set in our experiments. Hence, finally, we use the three human annotated binary relation data sets in our experiments. We give a brief description of these three data sets as follows.

1. PENN-100. This dataset was originally provided by the TreeKernel's authors [80]. PENN-100 contains 100 randomly chosen sentences from Penn Treebank.
2. WEB-500. WEB-500 is a widely used data set which is originally developed by Banko and Etzioni [56] for testing TextRunner. The sentences are collected from web by querying a search engine with pre-defined relations. Sentences from web are often informal. Informality is a big challenge to NLP techniques which were originally developed for well-written texts. This data set contains 500 sentences.
3. NYT-500. This dataset contains 500 randomly collected sentences from the New York Times corpus [81]. Sentences from New York Times are normally long and formal.

We compare SimpleIE with the following 7 ORE systems, which are also compared in experiments by Mesquita et al. [65]: Patty, Lund, OLLIE, SwiRL, ReVerb, Exemplar(S), SONEX. OLLIE, ReVerb, Exemplar(S) and SONEX have already been described in the previous chapter. We briefly describe the other systems here. Similar to SimpleIE, Patty [82] is based on dependency paths. But only paths starting with *nsubj*, *re-mod* and *partmod* are considered. Lund [83] and SwiRL [84] are based on the outputs of a semantic role labeling system. SwiRL is trained on PropBank and only extracts verb-mediated relations. In contrast, Lund is trained on both PropBank and NomBank. Hence it can potentially extract both verb-mediated and noun-mediated relations.

In this work, we focus on the task of extracting relation expressions rather than recognizing entities. To ensure entities are recognized properly for all systems, following the method used by Mesquita et al. [65], we replace the original

entities by two single-word entities ‘Europe’ and ‘Asia’. Mesquita et al. [65] claim that all systems in experiments can recognize these two entities correctly. In these benchmark data sets, a relation expression consists of one trigger word and several optional words. The trigger word is the most informative word which must appear in the extraction. And the optional words may appear in the relation extraction. Except the trigger word and the optional words, other words in the sentence are not allowed to appear in an extraction. Otherwise, the relation expression will be evaluated as incorrect extraction. In our opinion, the relations annotated in these benchmark data sets are informative but not very coherent. They do not contain prepositions and auxiliary verbs to connect words. Hence, for a fair comparison, we also remove the prepositions and auxiliary verbs from outputs of SimpleIE.

7.4.1 PENN

System	Precision	Recall	F1
Patty	0.62	0.27	0.38
Lund	0.92	0.40	0.56
OLLIE	0.80	0.40	0.53
SwiRL	1.00	0.17	0.29
ReVerb	0.83	0.17	0.28
Exemplar(S)	0.95	0.60	0.74
SONEX	0.93	0.47	0.62
<i>SimpleIE(CRF)</i>	0.96	0.83	0.89
<i>SimpleIE(CRN)</i>	0.96	0.83	0.89

Table 7.3: Precision, Recall and F1 on the PENN-50 Dataset

On the PENN-100 dataset, we use our Wikipedia dataset and half of the PENN-100 dataset for training SimpleIE, and test all systems on the other half of PENN-100 dataset. Hence the test data set contains 50 sentences and is denoted by PENN-50. For the 7 comparing systems, we directly use their outputs provided by Mesquita et al. [65]. Table 7.3 reports precision, recall and F1 scores for all systems on PENN-50 test data set, where

$$F1 = 2 \times \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

These results are automatically measured by using the evaluation software

provided by Mesquita et al. [65]. SimpleIE(CRF) is our SimpleIE system using CRF++¹² as the sequence labeling model. In contrast, SimpleIE(CRN) uses CRNs¹³ developed in Part I of this thesis as the sequence labeling model. From Table 7.3, we can see that SimpleIE clearly outperforms other systems on recall and F1. SwiRL achieves the best result on precision, but its recall is very low.

System	#Correct	#System	#GroundTruth
Patty	8	13	30
Lund	12	13	30
OLLIE	12	15	30
SwiRL	5	5	30
ReVerb	5	6	30
Exemplar(S)	18	19	30
SONEX	14	15	30
<i>SimpleIE(CRF)</i>	<u>25</u>	26	30
<i>SimpleIE(CRN)</i>	<u>25</u>	26	30

Table 7.4: Outputs on the PENN-50 Dataset

Table 7.4 lists more specific numbers. On the PENN-50 dataset, there are 30 ground truth extractions, which is given in the column **#GroundTruth**. The left 20 are NULL relations. The numbers of extractions output by systems are given in the column **#System**, and the numbers of correct extractions output by systems are given in the column **#Correct**.

7.4.2 WEB

Similarly, on the WEB-500 data set, we use our Wikipedia dataset and half of the WEB-500 dataset to train SimpleIE. The rest 250 sentences are used as test data set for all systems. Table 7.5 gives results for all systems. SimpleIE significantly outperforms other systems on recall and F1. But on precision, SimpleIE does not work as well as other systems. The reason may be that the informality of web sentences challenges the dependency parser, and reduces the precision of the whole system.

System	Precision	Recall	F1
Patty	0.56	0.35	0.43
Lund	0.91	0.29	0.44
OLLIE	0.80	0.30	0.44
SwiRL	0.97	0.29	0.45
ReVerb	0.91	0.29	0.44
Exemplar(S)	0.94	0.44	0.60
SONEX	<u>0.98</u>	0.27	0.42
<i>SimpleIE(CRF)</i>	0.76	0.64	0.69
<i>SimpleIE(CRN)</i>	0.82	<u>0.67</u>	<u>0.74</u>

Table 7.5: Precision, Recall and F1 on the WEB-250 Dataset

System	#Correct	#System	#GroundTruth
Patty	81	145	234
Lund	68	75	234
OLLIE	70	88	234
SwiRL	69	71	234
ReVerb	69	76	234
Exemplar(S)	104	111	234
SONEX	63	64	234
<i>SimpleIE(CRF)</i>	150	197	234
<i>SimpleIE(CRN)</i>	<u>159</u>	195	234

Table 7.6: Outputs on the WEB-250 Dataset

7.4.3 NYT

Table 7.7 and 7.8 list results obtained by systems on the NYT dataset. Similar to previous two data sets, we use our Wikipedia dataset and half of the NYT-500 data set to train SimpleIE. The rest 250 sentences are used for testing. On this dataset, SimpleIE achieves best results among all systems on all three metrics. As we mentioned, the NYT dataset consists of long and formal sentences from the journal of New York Times. Results show that SimpleIE performs much better than other systems on such data. This may be because the NLP components used by SimpleIE are more suitable for well-formed texts.

¹²<https://code.google.com/p/crfpp/>

¹³<https://github.com/zheminzhu/Co-occurrence-Rate-Networks>

System	Precision	Recall	F1
Patty	0.51	0.20	0.29
Lund	0.79	0.27	0.40
OLLIE	0.62	0.22	0.32
SwiRL	0.54	0.09	0.15
ReVerb	0.62	0.10	0.17
Exemplar(S)	0.70	0.32	0.44
SONEX	0.71	0.19	0.30
<i>SimpleIE(CRF)</i>	0.80	0.59	0.68
<i>SimpleIE(CRN)</i>	0.75	0.59	0.66

Table 7.7: Precision, Recall and F1 on the NYT-250 Dataset

System	#Correct	#System	#GroundTruth
Patty	16	31	81
Lund	22	28	30
OLLIE	18	29	30
SwiRL	7	13	30
ReVerb	8	13	30
Exemplar(S)	26	37	30
SONEX	15	21	30
<i>SimpleIE(CRF)</i>	48	60	30
<i>SimpleIE(CRN)</i>	48	64	30

Table 7.8: Outputs on the NYT-250 Dataset

7.4.4 Speed

Table 7.9 reports results on the speed metric. As we see, the systems based on shallow parsing, i.e., part-of-speech tagging, are the fastest, such as ReVerb and SONEX. Dependency-based systems, such OLLIE, Exemplar, Patty and our SimpleIE, follow at the second position in speed metric. The most expensive systems are SwiRL and LUND, which depend on the most expensive parsing technique, i.e., semantic role labeling. The difference between SimpleIE(CRN) and SimpleIE(CRF) on decoding speed is very small. Hence they are reported as one result under the name of SimpleIE.

System	PENN-50	WEB-250	NYT-250
ReVerb	<u>1.1</u>	<u>2.0</u>	<u>3.3</u>
SONEX	1.5	5.6	14.6
OLLIE	5.3	10.5	18.3
<i>SimpleIE</i>	23.9	52.8	162.4
Exemplar(S)	23.5	123.4	376.9
Patty	24.2	126.0	431.8
SwiRL	82.1	454.1	1083.3
Lund	197.2	706.1	4172.4

Table 7.9: Speed (Seconds)

7.4.5 CRNs vs CRFs

From Table 7.3, 7.5 and 7.7 we can see that SimpleIE(CRN) and SimpleIE(CRF) performs close on quality metrics. More specifically, on the PENN dataset, the outputs of SimpleIE(CRN) and SimpleIE(CRF) are exactly the same. On the WEB data set, SimpleIE(CRN) outperforms SimpleIE(CRF) on all metrics. But on the NYT dataset, the precision and recall of SimpleIE(CRN) are lower than SimpleIE(CRF).

In Part I of this thesis, we expect that CRNs can obtain competitive results on quality metrics to CRFs, while can be trained much faster than CRFs. Table 7.10 shows the training time for CRNs and CRFs on these three datasets. The time is measured separately from other parts of SimpleIE.

	PENN-50	WEB-250	NYT-250
CRF	6.29	5.65	5.12
CRN	<u>0.17</u>	<u>0.54</u>	<u>0.45</u>

Table 7.10: Speed (Seconds)

From Table 7.10, we see CRNs can be trained orders of magnitude faster than CRFs, which is consistent with our expectation.

7.5 Noun Phrase Recognition

We consider the following two sources as candidates of noun phrases in a sentence. The recognized noun phrases will be treated as source or target in a

dependency path.

1. *Atomic noun phrases.* Atomic noun phrases do not contain nested noun phrases. We use phrase tree structures output by the Stanford CoreNLP [76] to recognize atomic noun phrases. Pronouns which can be recognized from their part-of-speech tags are also considered as atomic noun phrases. We also attach original references to pronouns using a co-reference resolution system of the Stanford CoreNLP to ease annotation (see Figure 7.8). But the co-reference information has not been used in SimpleIE for relation extraction.
2. *Named entities.* Named entities are recognized using the named entity recognition (NER) components of the Stanford CoreNLP.

To solve conflicts between these two sources, we always prefer the longer ones. In the original Stanford dependency graph, each node in the dependency graph represents one word in the sentence. To simplify the complexity of the dependency graph, for noun phrases consisting of multiple words, we correspondingly merge their nodes in the dependency graph into one node. The inter-edges between them are removed, which significantly simplifies the original dependency graph.

7.6 Related Work on Sentence Simplification

Sentence simplification is a recently established area, which can be traced back to long ago. In this area, the target is to improve readability of texts, and most work are at sentence level. That is to simplify long and complex sentences to short and simple ones while retaining the information. There are majorly two strategies: (i) One strategy is simplifying by extracting. Heilman and Smith [85] and Barzilay [86] extract facts or phrases from sentences to rebuild new sentences. (ii) The other strategy is simplifying by transforming. In this strategy, operations, typically including deletion, insertion, splitting, substitution and reordering are employed to rewrite complex sentences. This can be data-driven [87, 88, 89, 90, 91, 92, 93] or rule-based [92]. Our work was mostly inspired by data-driven methods. Zhu et al. [88] compile a comparable data set consisting of complex-simple sentence pairs from Wikipedia and Simple Wikipedia. A probabilistic model based on simplification operations is trained with this data set to simplify sentences.

7.7 Summary

In this chapter, we proposed SimpleIE system for open relation extraction. Essentially, the idea of our model is to decompose an extraction pattern into a sequence of different types of simplification operations, including operations of insertion, deletion, splitting, reordering, and substitution. Our model can represent a diversity of extraction patterns and learn patterns from data. We compare SimpleIE with seven state-of-the-art systems on three benchmark data sets. Experimental results show that SimpleIE boosts recall and F1-score by at least 15% on all three data sets meanwhile obtaining very competitive precision.

We also compare co-occurrence rate networks (CRNs) with conditional Markov networks, i.e., conditional random fields (CRFs) by employing them as the sequence labeling model in SimpleIE. Experimental results show that CRNs can be trained orders of magnitude faster than CRFs while obtaining equal quality. These results are consistent with the experimental results reported on named entity recognition and part-of-speech tagging (Section 5). The experimental results on these three tasks, i.e., NER, POS tagging and ORE, together provide experimental evidence for verifying our theory developed in Chapter 3.

Part III

Conclusion

Conclusions and Future Work

Outline

This chapter is organized as follows. Firstly we make general conclusions to summarize this thesis. Then we revisit the research questions proposed in Chapter 1. Finally, we propose potential research topics in future work.

8.1 General Conclusions

At a high level, the work of this thesis can be concluded as follows.

8.1.1 Theoretical Aspects

The two fundamental questions of probabilistic graphical models (PGMs) can be described as follows:

1. Given a set of (conditional) independencies, how to decompose (factorize) a joint probability into a product of local factors?
2. How learning and inference can be efficiently conducted over the factorization representation?

To the first fundamental question of PGMs, the existing answers are Bayesian networks and Markov networks. In the factorization of a Bayesian network, a joint probability is decomposed to a product of conditional probabilities. In the factorization of a Markov network, based on the Hammersley–Clifford theorem, a joint probability is decomposed to a product of positive functions over (maximum) cliques.

In this thesis, we propose the Minimal Shared Information Principle (MSIP) to answer the first question. That is we try to find a factorization in which local

factors share minimum information between each other. The benefit is that we do not need to pay a lot of efforts to calibrate local factors to achieve a consistent status. To achieve the goal of MSIP, we propose a systematic framework called co-occurrence rate networks (CRNs) to find such a factorization. In the factorization of a CRN, the factors consists of all singleton probabilities and CR factors. There are two important properties of a CRN factorization: (i) A singleton probability does not share information with another singleton probability; (ii) A CR factor does not share information with another CR factor conditioned by singleton probabilities. These properties lead to an efficient separate training method. But for inference, we have not seen a CRN factorization brings any advantage. On three real-world NLP applications, we show that chain-structured CRNs can be trained almost two orders of magnitude faster than conditional random fields, while achieving competitive quality (often better on the overall quality metric F1).

8.1.2 Application Aspects

The second part of this thesis focuses on open relation extraction (ORE). We propose a general model called SimpleIE (Chapter 7) to represent diverse extraction patterns. The basic idea of SimpleIE is to *decompose* an extraction pattern into a sequence of simplification operations. Decomposition makes the model general. Because by decomposing a pattern into components, we can re-combine these components in a new way to generate new patterns, which shows the power of the decomposition strategy discussed in Chapter 2. By training a sequence labeling model, SimpleIE can automatically learn extraction patterns from training data. Experimental results show that on three benchmark data sets, SimpleIE boosts recall and F1 at least 15% over seven state-of-the-art ORE systems (Section 7.4).

The backbone of SimpleIE is a sequence labelling model, i.e., a chain-structured undirected graphical model. As we discussed in Part I, undirected graphs can be modeled by Markov networks or our co-occurrence rate networks. Experimental results show that our CRNs can be trained much faster than CRFs (Table 7.10), while achieving equal quality (Section 7.4).

8.2 Research Questions Revisited

To make conclusions more specific, we revisit the research questions proposed in Chapter 1.

Q1. How to obtain a factorization in which factors share minimum information? More specifically, what independence semantics should be endowed? Is the factorization equivalent to the given independencies?

A: This question has been answered in Chapter 3. Briefly, given a joint probability, its CRN factorization is obtained as follows. We first strip off singleton probabilities from the joint probability. The quantity left is called co-occurrence rate (CR). CR is a symmetric quantity which measures mutual dependence among variables involved. Then we further decompose the joint CR into smaller and independent CRs if it is possible. Finally, we obtain a CRN factorization whose factors consist of all singleton probabilities and CR factors. As described in Chapter 3, the (conditional) independencies endowed to a CRN graph are essentially the same as Markov networks. And the CRN factorization can be proved equivalent to the given independencies.

Q2. How to prove the theory of co-occurrence rate networks is sound? Soundness can be verified by proofs or experiments.

A: In Chapter 3, we gave proofs to all theorems. Most of these proofs are simple, and can be easily verified. Also experimental results on named entity recognition (Section 5.6.1), part-of-speech tagging (Section 5.6.2) and open relation extraction (Section 7.4.5) show that chain-structured CRNs can be trained much faster than Markov networks while achieving competitive quality. These experimental results are consistent with our expectation in theory. Proofs and experiments are direct evidences of the soundness of CRNs. In Section A.3, we give a proof of the Hammersley-Clifford theorem based on the theorems of co-occurrence rate. This shows the consistency between CRNs and the existing theory. Hence it can be considered as an indirect evidence of the soundness of CRNs.

Q3. What are the advantages of co-occurrence rate networks? Do CRNs bring any added value?

A: Yes, CRN factorizations lead to an efficient separate training method. See Section 8.1.1 for more details. Another obvious advantage is that just like Bayesian networks, CRNs can be easily understood without mysterious notions like ‘compatibility’ or ‘affinity’. CRNs can be explained

clearly with some simple quantities and formulas. This simplifies the theoretical framework.

Q4. In open relation extraction, how to model diverse extraction patterns with graphical models? Considering the diversity of extraction patterns, the model should be general enough to represent diverse patterns. Also the model is expected to be able to automatically learn extraction patterns from training data.

A: In Chapter 7, we decompose an extraction pattern into a sequence of simplification operations including insertion, deletion, reordering, substitution, and splitting. This is essentially a sequence labeling model, and hence can be considered as a chain-structured undirected graphical model. Decomposition brings the benefit that we can re-combine these components (simplification operations) in a new way to form new extraction patterns. Hence, our model can capture diverse extraction patterns. By training the sequence labeling model in SimpleIE, extraction patterns can be automatically learned from data.

Q5. How to evaluate the system? To evaluate our system, we need to compare it with other systems on benchmark datasets.

A: In Section 7.4, we compare our system with seven state-of-the-art systems on three benchmark data sets.

Q6. How well do CRNs perform on open relation extraction comparing with Markov networks? Are experimental results consistent with our expectations in Part I?

A: The experiments in Section 7.4 show that CRNs can be trained orders of magnitude faster than Markov networks, while achieving equally good quality. These results are consistent with our expectations in Part I, and also consistent with experimental results on NER and POS tagging (Chapter 5).

8.3 Future Work

Due to limited amount of time and the ambition of our research topics, many ideas need to be implemented in future. We list some of them as follows. Several items below have already been mentioned in previous chapters.

- F1. Learning and Inference on Cyclic Clique Graphs.** In this thesis, we mainly focused on acyclic clique graphs. The treatment for cyclic clique graphs is to transform them to acyclic clique graphs by triangulation. An interesting and challenging research topic is to explore if we can obtain a CRN factorization with respect to MSIP for a acyclic clique graph directly. The proof of Hammersley–Clifford theorem may provide some hints, in which a global but fixed configuration is used.
- F2. Bayesian Paradigm.** In this thesis, we only explored maximum likelihood estimation for co-occurrence rate networks. Maximum likelihood estimation is a point estimate which assumes that the true value for a parameter is a unique point. To our experience, most of the time, MLE works well for NLP tasks. Bayesian methods are another popular paradigm. Bayesian methods assume the true value of a parameter is uncertain under a prior distribution. It seems that the advantages of co-occurrence rate networks are more obvious in the Bayesian setting. Firstly, as the factors of a CRN factorization, which are singleton probabilities and CR factors, have a probabilistic definition, it is easier to choose a prior. Secondly, Bayesian inference needs to sum over a distribution, which can be expensive. The local normalization of CRNs may alleviate this problem because no global normalization needs to be calculated. These can be interesting research topics in the future.
- F3. Experiments.** The experiments of CRNs in this thesis are limited to chain-structured graphs. For more complex graphs we need more complex application scenes. But it is not easy to find such a real-world application with in NLP. In natural language processing, the graph structures are commonly chain- or tree-structured. In the future, applications from other areas, such as speech processing, computer vision, biomedicine and information retrieval, can be considered.
- F4. Causal Relations.** In the second part of this thesis, we did open relation extraction from texts. Causal relations are a special type of relations which are very interesting and important. Causal relations can be identified in a

quantitative way, such as causal Bayesian networks. Here we note another approach for identifying causal relations. We call this approach as the linguistic or NLP approach. There are trigger words, such as 'because', 'lead to', 'cause', 'hence' ..., indicating causal relations in texts. In the age of big data, there are tons of texts generated by individuals on the web, such as twitter, forum, facebook, SMS, Wikipedia, etc. Also relation extraction from biomedical texts is a hot topic. It could be very useful if we can develop techniques for extracting causal relations from texts on the web. Imagine the scenario that we will attend an examination of driving license tomorrow, if we read a list of reasons which lead to failure of an driving examination tonight, we may avoid failure and success in the examination. These reasons can be gathered from web by a causal relation extractor. Also temporal information of related events is important in casual relation extraction. Because the cause must happen before the effect. Temporal expressions need to be recognized and attached to corresponding events. Causal relation extraction from web or other electronic texts can be a big arena for techniques in both NLP and graphical models.

F5. Continuous Variables, Hidden Variables, Unknown Graph Structures.

In this thesis, we assume all variables are discrete and observable, data are complete and graph structures are given. If these conditions do not hold, we need more complex treatments.

Co-occurrence rate can be extended to continuous variables by replacing the probability mass function P with the probability density function f :

Definition 32 (Continuous Co-occurrence Rate). *Let $\{X_1, X_2, \dots, X_N\}$ be a set of continuous random variables. The continuous co-occurrence rate is defined as follows.*

$$\text{CR}(X_1 ; X_2 ; \dots ; X_N) = \frac{f_{X_1 \dots X_N}(X_1, X_2, \dots, X_N)}{f_{X_1}(X_1)f_{X_2}(X_2)\dots f_{X_N}(X_N)}, \quad (8.1)$$

where f is the probability density function.

The continuous CR happens to be the density function of Copula [25, 26]. Copula is a statistical framework for describing dependence. The Sklar's theorem [26], which is the most fundamental theorem in Copula, states that for continuous random variables there exists a unique Copula function, which is denoted by C , such that:

$$F_{X_1, X_2, \dots, X_N} = C(F_{X_1}, F_{X_2}, \dots, F_{X_N}).$$

F is the cumulative distribution function (cdf). Sklar's theorem says the joint cdf can be constructed from the univariate cdf's by a Copula function. A joint cdf contains all information of (i) univariate cdf's and (ii) dependence relations among uni-variables. Hence, *the Copula function contains all information of the dependence relations*. Assume the joint cdf and the univariate cdf's are all differentiable, and take partial derivatives with respect to $X_1, X_2, \dots, X_N$ to both sides of the equation above, by the chain rule, we obtain:

$$f_{X_1, X_2, \dots, X_N} = \frac{\partial^N C(F_{X_1}, F_{X_2}, \dots, F_{X_N})}{\partial F_{X_1} \partial F_{X_2} \dots \partial F_{X_N}} f_{X_1} f_{X_2} \dots f_{X_N}.$$

Hence,

$$\frac{f_{X_1, X_2, \dots, X_N}}{f_{X_1} f_{X_2} \dots f_{X_N}} = \frac{\partial^N C(F_{X_1}, F_{X_2}, \dots, F_{X_N})}{\partial F_{X_1} \partial F_{X_2} \dots \partial F_{X_N}}$$

The left hand side is just the continuous CR (Definition 32), and the right hand side is the density function of the Copula. Hence the continuous CR is just the density function of the Copula. Since a Copula contains all information of the dependence relations among variables involved, its density function contains (almost) all information of the dependence relations. This is because the density function is obtained from the Copula by taking derivatives. By taking derivatives, we will not lose information of the Copula except some constant terms. Therefore, we obtain an important result: *the continuous CR contains almost all information of the dependence relations among variables involved*.

The properties and results obtained for discrete CR in Chapter 3 still hold for the continuous CR. Hence, our theoretical framework can be extended to continuous variables.

Hidden variables can be learned by EM algorithms. Graph structures can be constructed by independence testing methods. These are interesting topics to be explored in future.

- F6. The Quality Model.** We propose a hierarchical multidimensional quality model (Figure 6.1) in Chapter 6 for measuring the quality of relation extractions. But these metrics have to be judged by human, which makes the evaluation very expensive. It is interesting and useful if we can develop automatic methods for measuring these metrics. Intuitively, we may use the perplexity between the extractions and original sentences to

roughly measure the coherence metric. The average length of extractions seems a good indicator for the simplicity metric. Faithfulness may be verified with redundancy. Informativeness may be measured by the coverage of key phrases, which can be identified automatically by existing NLP tools.

- F7. Consistency between Separate Models.** In Chapter 4 we discussed the issues of consistency between separate models. The problem of consistency happens between a CR factors and the singleton probabilities involved. In future, we will explore improving this kind of consistency in an economic way.
- F8. Quantitative vs Qualitative.** The shared information between factors discussed in this thesis is more or less treated qualitatively. In future, we can consider to introduce quantitative measurement for shared information. For example, the entropy or relative entropy as the measurement of the information shared between factors. By quantifying the notion of inconsistency, we expect to find the optimal separate models by minimizing inconsistency between models.

Appendices

Appendix

A.1 Axioms of Probability

Definition 33 (Probability). Let Ω be a sample space and let $\mathcal{B}$ be the set of events. Let P be a real-valued function defined on $\mathcal{B}$. Then P is a probability set function if P satisfies the following three axioms:

1. $P(\mathcal{S}) \geq 0, \forall \mathcal{S} \in \mathcal{B}$.
2. $P(\Omega) = 1$.
3. If $\{\mathcal{S}_n\}$ is a sequence of events in $\mathcal{B}$, and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset, \forall i \neq j$, then

$$P(\cup_{n=1}^{\infty} \mathcal{S}_n) = \sum_{n=1}^{\infty} P(\mathcal{S}_n).$$

A.2 Proof of $\mathcal{I}_l(G) \Leftrightarrow F_{BN}(G)$

Proof. (i) First, we prove the forward direction: $\mathcal{I}_l(G) \Rightarrow F_{BN}(G)$. That is given the independencies in $\mathcal{I}_l(G)$, we want to obtain $F_{BN}(G)$.

Let $\mathcal{X}$ be all the nodes of G . Assume $[X_1, X_2, \dots, X_n]$ is a topological ordering of $\mathcal{X}$ over G . In other words, parents are put before their descendants. By the chain rule of a joint probability, we have:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i \mid X_1, \dots, X_{i-1}).$$

Chain rule always holds no matter what ordering is. $\forall X_i \in \mathcal{X}$, the rest nodes of G is denoted by $\mathcal{X} \setminus X_i$. $\mathcal{X} \setminus X_i$ can be partitioned into three parts: Pa_i

(parents of X_i), De_i (descants of X_i), and the rest R_i . Due to the topological ordering, $\{X_1, \dots, X_{i-1}\} = Pa_i \cup R_i$. As $(X_i \perp\!\!\!\perp R_i \mid Pa_i) \in \mathcal{I}_l(G)$, we have $P(X_i \mid X_1, \dots, X_{i-1}) = P(X_i \mid Pa_i)$. Hence, we obtain $F_{BN}(G)$:

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i \mid Pa_i).$$

(ii) Then, we prove the other direction: $\mathcal{I}_l(G) \Leftarrow F_{BN}(G)$. That is given $F_{BN}(G)$, we want to prove $\forall X_i \in \mathcal{X}, P(X_i \mid Pa_i, R_i) = P(X_i \mid Pa_i)$.

$$\begin{aligned} P(X_i \mid Pa_i, R_i) &= \frac{P(X_i, Pa_i, R_i)}{P(Pa_i, R_i)} \\ &= \frac{\sum_{De_i} P(X_1, \dots, X_n)}{\sum_{X_i} \sum_{De_i} P(X_1, \dots, X_n)} \\ &= \frac{\sum_{De_i} \prod_{j=1}^n P(X_j \mid Pa_j)}{\sum_{X_i} \sum_{De_i} \prod_{j=1}^n P(X_j \mid Pa_j)} \\ &= \frac{\prod_{X_j \in \mathcal{X} \setminus De_i} P(X_j \mid Pa_j)}{\prod_{X_k \in \mathcal{X} \setminus (De_i \cup X_i)} P(X_k \mid Pa_k)} \\ &= P(X_i \mid Pa_i) \end{aligned}$$

Due to (i) and (ii), we obtain $\mathcal{I}_l(G) \Leftrightarrow F_{BN}(G)$. □

A.3 Proof of the Hammersley-Clifford Theorem

The Hammersley-Clifford Theorem states $\mathcal{I}_{MN} \Leftrightarrow F_{MN}$ (See Theorem 4).

Proof. ($\Leftarrow$). Our proof in this direction is the same as the traditional proof (See Theorem 4.8 in [3]).

($\Rightarrow$). In principle, our proof in this direction is similar to the traditional proof. The only difference is that we use CR rather than conditional probabilities in the proof. Theorem 17 is the key in this proof. According to the Definition 26, we have:

$$P(\mathcal{X}) = \text{CR}(\text{Sem } \mathcal{X}) \prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i)$$

To obtain F_{CRN} , we need to further decompose $\text{CR}(\text{Sem } \mathcal{X})$ using independencies in $\mathcal{I}_{CRN}$. Let $\mathcal{P}(\mathcal{X})$ be the power set of $\mathcal{X}$. That is $\mathcal{P}(\mathcal{X})$ contains all possible

sub-sets of $\mathcal{X}$. Let $|\mathcal{X}|$ be the cardinality (number of elements) of $\mathcal{X}$, and $\mathbf{x}$ be an arbitrary but fixed assignment to $\mathcal{X}$. The following identity holds obviously because numerators equal denominators:

$$1 = \prod_{S \in \mathcal{P}(\mathcal{X}) - \mathcal{X}} \left[\frac{\text{CR}(\text{Sem } S; \text{Sem } \mathcal{X} - S = \mathbf{x}_{\mathcal{X}-S})}{\text{CR}(\text{Sem } S; \text{Sem } \mathcal{X} - S = \mathbf{x}_{\mathcal{X}-S})} \right]^U, \quad (\text{A.1})$$

where $U = 2^{|\mathcal{X}| - |\mathcal{S}| - 1}$. We denote the right hand side of this identity by Q , then:

$$P(\mathcal{X}) = Q \times \text{CR}(\text{Sem } \mathcal{X}) \times \prod_{\mathcal{X}_i \in \mathcal{X}} P(\mathcal{X}_i) \quad (\text{A.2})$$

Then we group all CRs in Equation A.2, including CRs in Q and $\text{CR}(\text{Sem } \mathcal{X})$, into groups. For each $\mathcal{A} \in \mathcal{P}(\mathcal{X})$, where $\mathcal{P}(\mathcal{X})$ is the power set over $\mathcal{X}$, we group the following CRs into $G_{\mathcal{A}}$:

$$G_{\mathcal{A}} = \{\text{CR}(\text{Sem } S; \text{Sem } \mathcal{X} - S = \mathbf{x}_{\mathcal{X}-S})^{(-1)^{|\mathcal{A}| - |\mathcal{S}|}} : \forall S \in \mathcal{P}(\mathcal{A})\}.$$

If $|\mathcal{A}| - |\mathcal{S}|$ is odd, then the element is

$$\frac{1}{\text{CR}(\text{Sem } S; \text{Sem } \mathcal{X} - S = \mathbf{x}_{\mathcal{X}-S})}$$

Otherwise, the element is

$$\text{CR}(\text{Sem } S; \text{Sem } \mathcal{X} - S = \mathbf{x}_{\mathcal{X}-S})$$

We denote the product of all CRs in $G_{\mathcal{A}}$ by $\prod(G_{\mathcal{A}})$.

The following two binomial identities guarantee that all CRs in Equation A.2 are just associated into groups $\{G_{\mathcal{A}} : \mathcal{A} \in \mathcal{P}(\mathcal{X})\}$:

$$\begin{aligned} 2^N &= (1 + 1)^N = \binom{N}{0} + \binom{N}{1} + \dots + \binom{N}{N} \\ 0^N &= (1 - 1)^N = \binom{N}{0} - \binom{N}{1} + \dots + (-1)^N \binom{N}{N} \end{aligned}$$

where $N = |\mathcal{X}| - |\mathcal{S}|$.

We go on to prove if $\mathcal{A}$ is not a clique (complete sub-graph), then $\prod(G_{\mathcal{A}}) = 1$. As $\mathcal{A}$ is not a clique, there must exist two unconnected nodes $\mathcal{X}_a$ and $\mathcal{X}_b$ in $\mathcal{A}$. For each $\mathcal{W} \in \mathcal{P}(\mathcal{X} - \{\mathcal{X}_a, \mathcal{X}_b\})$, then all CRs selected into $\mathcal{A}$ can be categorized

into four types: $\mathcal{W}$, $\mathcal{W} \cup \{\mathcal{X}_a\}$, $\mathcal{W} \cup \{\mathcal{X}_b\}$ and $\mathcal{W} \cup \{\mathcal{X}_a, \mathcal{X}_b\}$, and they can be written as follows:

$$\begin{aligned}
 \prod(G_{\mathcal{A}}) &= \prod_{\mathcal{S} \in \mathcal{P}(\mathcal{A})} \text{CR}(\text{Sem } \mathcal{S} ; \text{Sem } \mathcal{X} - \mathcal{S} = \mathbf{x}_{\mathcal{X}-\mathcal{S}})^{(-1)^{|\mathcal{A}|-|\mathcal{S}|}} \\
 &= \prod_{\mathcal{W} \in \mathcal{P}(\mathcal{X}-\{\mathcal{X}_a, \mathcal{X}_b\})} \left[\frac{\text{CR}(\text{Sem } \mathcal{W} ; \mathcal{X}_a = \mathbf{x}_a ; \mathcal{X}_b = \mathbf{x}_b ; \text{Sem } \mathcal{X}^* = \mathbf{x}^*)}{\text{CR}(\text{Sem } \mathcal{W} ; \mathcal{X}_a = \mathbf{x}_a ; \mathcal{X}_b ; \text{Sem } \mathcal{X}^* = \mathbf{x}^*)} \right. \\
 &\quad \left. \frac{\text{CR}(\text{Sem } \mathcal{W} ; \mathcal{X}_a ; \mathcal{X}_b ; \text{Sem } \mathcal{X}^* = \mathbf{x}^*)}{\text{CR}(\text{Sem } \mathcal{W} ; \mathcal{X}_a ; \mathcal{X}_b = \mathbf{x}_b ; \text{Sem } \mathcal{X}^* = \mathbf{x}^*)} \right]^{-1^*} \\
 &= 1
 \end{aligned}$$

where $\mathcal{X}^* = \mathcal{X} - \mathcal{W} - \{\mathcal{X}_a, \mathcal{X}_b\}$. Only the relative positions of the four factors are significant. Hence we use -1^* to represent the power. According to Theorem 17, this equation equals 1. Now only the CRs selected into cliques are left, which can be further absorbed into maximum cliques. $\square$

Bibliography

- [1] C. Eickhoff and A. P. de Vries, "Modelling Complex Relevance Spaces With Copulas," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014, pp. 1831 – 1834.
- [2] N. A. Smith, *Linguistic Structure Prediction*, G. Hirst, Ed. Morgan & Claypool Synthesis Lectures on Human Language Technologies, 2011.
- [3] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.
- [4] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., 2006.
- [5] J. M. Hammersley and P. E. Clifford, "Markov random fields on finite graphs and lattices," Unpublished manuscript, 1971.
- [6] M. J. Wainwright, T. Jaakkola, and A. S. Willsky, "Tree-based reparameterization framework for analysis of sum-product and related algorithms," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1120–1146, 2003. [Online]. Available: <http://dblp.uni-trier.de/db/journals/tit/tit49.html#WainwrightJW03>
- [7] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1-2, pp. 1–305, Jan. 2008. [Online]. Available: <http://dx.doi.org/10.1561/22000000001>
- [8] Y. W. Teh and M. Welling, "On improving the efficiency of the iterative proportional fitting procedure," in *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, vol. 9, 2003.
- [9] B. Cseke and T. Heskes, "Properties of bethe free energies and message passing in gaussian models," *Journal of Artificial Intelligence Research*, vol. 41, pp. 1–24, 2011. [Online]. Available: <http://www.jair.org/papers/paper3195.html>
- [10] C. A. Sutton and A. McCallum, "Piecewise training for undirected models," in *UAI 05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, July 26-29 2005, Edinburgh, Scotland*, 2005, pp. 568–575.
- [11] T. Apostol, *Mathematical Analysis*. Pearson, 1974.
- [12] S. Friedberg, A. Insel, and L. Spence, *Linear Algebra*, 4th ed. Pearson, 2002.

- [13] D. Hiemstra, "Using language models for information retrieval," Ph.D. dissertation, University of Twente, Enschede, 2001.
- [14] P. Koehn, *Statistical Machine Translation*, 1st ed. Cambridge University Press, 2010.
- [15] Y. Bengio, "Learning deep architectures for AI," *Foundation and Trends in Machine Learning*, vol. 2, pp. 1–127, 2009.
- [16] M. Fokkinga, "Mapreduce — a two-page explanation for laymen," 2008, unpublished Technical Report.
- [17] T. Heskes, "On the uniqueness of loopy belief propagation fixed points," *Neural Comput.*, vol. 16, pp. 2379–2413, 2004.
- [18] —, "Stable fixed points of loopy belief propagation are local minima of the bethe free energy," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. MIT Press, 2003, pp. 359–366.
- [19] J. M. Mooij and H. J. Kappen, "Sufficient conditions for convergence of loopy belief propagation," in *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI-05)*, 2005, pp. 396–403.
- [20] Z. Zhu, D. Hiemstra, P. M. G. Apers, and A. Wombacher, "Empirical co-occurrence rate networks for sequence labeling," in *The 12th International Conference on Machine Learning and Applications*, 2013, pp. 93–98.
- [21] Z. Zhu, *Factorizing Probabilistic Graphical Models Using Cooccurrence Rate*, *arXiv:1008.1566v1*, August 2010.
- [22] C. Sutton and A. McCallum, "Piecewise training for structured prediction," *Mach. Learn.*, vol. 77, pp. 165–194, 2009.
- [23] K. W. Church and P. Hanks, "Word association norms, mutual information, and lexicography," *Comput. Linguist.*, vol. 16, pp. 22–29, 1990.
- [24] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, pp. 379–423, 1948.
- [25] M. Fréchet, "Sur les tableaux de corrélation dont les marges sont donnés," *Ann. Univ. Lyon, Science*, vol. 4, pp. 13–84, 1951.
- [26] A. Sklar, "Fonctions de répartition à n dimensions et leurs marges," *Publ. Inst. Statist. Univ. Paris*, vol. 8, pp. 229–231, 1959.
- [27] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in Neural Information Processing Systems 13*, T. Leen, T. Dietterich, and V. Tresp, Eds. MIT Press, 2001, pp. 689–695. [Online]. Available: <http://papers.nips.cc/paper/1832-generalized-belief-propagation.pdf>
- [28] M. Welling, "On the choice of regions for generalized belief propagation." in *UAI*, 2004, pp. 585–592.

- [29] Z. Zhu, D. Hiemstra, and P. M. G. Apers, "Linear co-occurrence rate networks (l-crns) for sequence labeling," in *Statistical language and speech processing*, 2014, pp. 185–196.
- [30] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. Inf. Theor.*, vol. 14, no. 3, pp. 462–467, 2006.
- [31] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks*, 1st ed. Springer Publishing Company, Incorporated, 2007.
- [32] S. L. Lauritzen, *Graphical Models*. Oxford University Press, 1996.
- [33] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, pp. 199–222, 2004.
- [34] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, 1998, pp. 137–142.
- [35] Z. Zhu, D. Hiemstra, P. M. G. Apers, and A. Wombacher, "Comparison of local and global undirected graphical models," in *The 22nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2014, pp. 479–484.
- [36] Z. Zhu, D. Hiemstra, P. M. G. Apers, and A. Wombacher, "Closed form maximum likelihood estimator of conditional random fields," <http://eprints.eemcs.utwente.nl/23097/>, Centre for Telematics and Information Technology, University of Twente, Enschede, Technical Report TR-CTIT-13-03, February 2013.
- [37] M. Badieh Habib Morgan, "Named entity extraction and disambiguation for informal text: the missing link," Ph.D. dissertation, University of Twente, 2014.
- [38] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini, "Building a large annotated corpus of english: The penn treebank," *Comput. Linguist.*, pp. 313–330, 1993.
- [39] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of the IEEE*, 1989, pp. 257–286.
- [40] A. McCallum, D. Freitag, and F. C. N. Pereira, "Maximum entropy markov models for information extraction and segmentation," 2000, pp. 591–598.
- [41] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, 2001, pp. 282–289.
- [42] Y. Altun, A. J. Smola, and T. Hofmann, "Exponential families for conditional random fields," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, 2004, pp. 2–9.
- [43] Z. Ghahramani, "Hidden markov models," 2002, pp. 9–42.

- [44] T. A. Cohn, "Scaling conditional random fields for natural language processing," Ph.D. dissertation, University of Melbourne, 2007.
- [45] D. Klein and C. D. Manning, "Conditional structure versus conditional estimation in nlp models," 2002, pp. 9–16.
- [46] P. Le-Hong, X.-H. Phan, and T.-T. Tran, "On the effect of the label bias problem in part-of-speech tagging," in *The 10th IEEE RIVF International Conference on Computing and Communication Technologies*, 2013, pp. 103 – 108.
- [47] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, 2003, pp. 173–180.
- [48] J. Besag, "Statistical analysis of non-lattice data," *The Statistician*, vol. 24, pp. 179–195, 1975.
- [49] C. Sutton and A. McCallum, "An introduction to conditional random fields," *Foundations and Trends in Machine Learning*, pp. 267–373, 2012.
- [50] M. S. Ryan and G. R. Nudd, "The viterbi algorithm," Tech. Rep., 1993.
- [51] T. Kudo, "Crf++: Yet another crf toolkit," free software, March 2012. [Online]. Available: <http://crfpp.googlecode.com/svn/trunk/doc/index.html>
- [52] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.
- [53] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of CoNLL-2003*, 2003.
- [54] W. N. Francis and H. Kucera, "Brown corpus manual," Department of Linguistics, Brown University, Providence, Rhode Island, US, Tech. Rep., 1979.
- [55] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni, "Open information extraction from the web," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 2670–2676.
- [56] M. Banko and O. Etzioni, "The tradeoffs between open and traditional relation extraction," in *Proceedings of ACL-08: HLT*, 2008, pp. 28–36.
- [57] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 1535–1545.
- [58] Mausam, M. Schmitz, R. Bart, S. Soderland, and O. Etzioni, "Open language learning for information extraction," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 523–534.

- [59] H. Bast and E. Haussmann, "Open information extraction via contextual sentence decomposition." in *ICSC*, 2013, pp. 154–159.
- [60] —, "More informative open information extraction via simple inference," in *Advances in Information Retrieval*, 2014, pp. 585–590.
- [61] J. Christensen, Mausam, S. Soderland, and O. Etzioni, "Semantic role labeling for open information extraction," in *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, 2010, pp. 52–60.
- [62] A. Fader, L. Zettlemoyer, and O. Etzioni, "Paraphrase-driven learning for open question answering," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 1608–1618.
- [63] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 74–84.
- [64] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: A collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 2008, pp. 1247–1250.
- [65] F. Mesquita, J. Schmidek, and D. Barbosa, "Effectiveness and efficiency of open relation extraction," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 447–457.
- [66] F. Wu and D. S. Weld, "Open information extraction using Wikipedia," in *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010, pp. 118–127.
- [67] M. Koch, J. Gilmer, S. Soderland, and D. S. Weld, "Type-aware distantly supervised relation extraction with linked arguments," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1891–1901.
- [68] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part III*, 2010, pp. 148–163.
- [69] Y. Shinyama and S. Sekine, "Preemptive information extraction using unrestricted relation discovery," in *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2006, pp. 304–311.
- [70] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and M. Mausam, "Open information extraction: The second generation," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, 2011, pp. 3–10.

- [71] L. Del Corro and R. Gemulla, "Clausie: Clause-based open information extraction," in *Proceedings of the 22Nd International Conference on World Wide Web*, 2013, pp. 355–366.
- [72] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *J. Mach. Learn. Res.*, vol. 3, pp. 1083–1106, 2003.
- [73] A. Culotta and J. Sorensen, "Dependency tree kernels for relation extraction," in *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, 2004, pp. 423–429.
- [74] R. C. Bunescu and R. J. Mooney, "A shortest path dependency kernel for relation extraction," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 724–731.
- [75] N. Bach and S. Badaskar, "A review of relation extraction," *Literature review for language and statistics II*, Carnegie Mellon University, 2007.
- [76] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.
- [77] M.-C. de Marneffe and C. D. Manning. (2008) Stanford typed dependencies manual. [Online]. Available: http://nlp.stanford.edu/software/dependencies_manual.pdf
- [78] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, pp. 27:1–27:27, 2011.
- [79] T. Zesch, C. Müller, and I. Gurevych, "Extracting lexical semantic knowledge from Wikipedia and Wiktionary," in *Proceedings of the Conference on Language Resources and Evaluation, electronic proceedings*, 2008.
- [80] Y. Xu, M.-Y. Kim, K. Quinn, R. Goebel, and D. Barbosa, "Open information extraction with tree kernels," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 868–877.
- [81] V. Punyakanok, D. Roth, and W.-t. Yih, "The new york times annotated corpus ldc2008t19," 2008. [Online]. Available: <https://catalog.ldc.upenn.edu/LDC2008T19>
- [82] N. Nakashole, G. Weikum, and F. Suchanek, "Patty: A taxonomy of relational patterns with semantic types," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 1135–1145.
- [83] R. Johansson and P. Nugues, "Dependency-based semantic role labeling of propbank," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 69–78.

- [84] M. Surdeanu, S. Harabagiu, J. Williams, and P. Aarseth, "Using predicate-argument structures for information extraction," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, 2003, pp. 8–15.
- [85] M. Heilman and N. Smith, "Extracting simplified statements for factual question generation," in *Proceedings of the Third Workshop on Question Generation*, 2010, pp. 11–20.
- [86] R. Barzilay, "Information fusion for multi-document summarization: Paraphrasing and generation," Ph.D. dissertation, Columbia University, 2003.
- [87] S. M. Aluísio, L. Specia, T. A. Pardo, E. G. Maziero, and R. P. Fortes, "Towards brazilian portuguese automatic text simplification systems," in *Proceedings of the Eighth ACM Symposium on Document Engineering*, 2008, pp. 240–248.
- [88] Z. Zhu, D. Bernhard, and I. Gurevych, "A monolingual tree-based translation model for sentence simplification," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 1353–1361.
- [89] K. Woodsend and M. Lapata, "Learning to simplify sentences with quasi-synchronous grammar and integer programming," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 409–420.
- [90] S. Wubben, A. V. D. Bosch, and E. Krahmer, "Sentence simplification by monolingual machine translation," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012, pp. 1015–1024.
- [91] D. Kauchak, "Improving text simplification language modeling using unsimplified text data," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013, pp. 1537–1546.
- [92] M. Angrosh, T. Nomoto, and A. Siddharthan, "Lexico-syntactic text simplification and compression with typed dependencies," in *Proceedings of the 25th International Conference on Computational Linguistics*, 2014, pp. 1996–2006.
- [93] S. Narayan and C. Gardent, "Hybrid simplification using deep semantics and machine translation," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014, pp. 435–445.

Publications by the Author

Publications

- **Zhemin Zhu**, Djoerd Hiemstra, Peter Apers, *Closed Form Maximum Likelihood Estimator of Conditional Random Fields*, University of Twente, Technical Report TR-CTIT-13-03, Centre for Telematics and Information Technology, University of Twente, 2013.
- **Zhemin Zhu**, Djoerd Hiemstra, Peter Apers, Andreas Wombacher, *Empirical Co-occurrence Rate Networks for Sequence Labeling*, in Proceedings of the 12th International Conference on Machine Learning and Applications (ICMLA), pp. 93-98, 2013.
- **Zhemin Zhu**, Djoerd Hiemstra, Peter Apers, Andreas Wombacher, *Comparison of Local and Global Undirected Graphical Models*, In Proceedings of the 22nd European Symposium on Artificial Neural Networks (ESANN), pp. 479-484, 2014.
- **Zhemin Zhu**, Djoerd Hiemstra, Peter Apers, Andreas Wombacher, *UT-DB: an Experimental Study on Sentiment Analysis in Twitter*, In Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval), pp. 384-389, 2013.
- Mena B Habib, Maurice van Keulen, **Zhemin Zhu**, *Named Entity Extraction and Linking Challenge: University of Twente at # microposts2014*, In Proceedings of the the 4th Workshop on Making Sense of Microposts co-located with the 23rd International World Wide Web Conference (WWW), pp. 64-65, 2014.
- **Zhemin Zhu**, Djoerd Hiemstra, Peter Apers, *Linear Co-occurrence Rate Networks (L-CRNs) for Sequence Labeling*, In Proceedings of Statistical Language and Speech Processing (SLSP), pp. 185-196, 2014.
- **Zhemin Zhu**, Djoerd Hiemstra, Peter Apers, Andreas Wombacher, *Empirical Training for Conditional Random Fields*, In Proceedings of the 22nd Belgian-Dutch Conference on Machine Learning (BENELEARN), pp. 105-105, 2013.
- **Zhemin Zhu**, Delphine Bernhard, Iryna Gurevych, *A Monolingual Tree-based Translation Model for Sentence Simplification*, in Proceedings of the 23rd international conference on computational linguistics (COLING), pp. 1353-1361, 2010.
- **Zhemin Zhu**, Delphine Bernhard, Iryna Gurevych, *A Multi-dimensional Model for Assessing the Quality of Answers in Social Q& A Sites*, TU Darmstadt, Technical Report TUD-CS-2009-0158, 2009.

- **Zhemín Zhu**, Chen Wang, Li Ma, Yue Pan, Zhiming Ding, *Scalable Community Discovery of Large Networks*, in Proceedings of The Ninth International Conference on Web-Age Information Management (WAIM), pp. 381-388, 2009.
- **Zhemín Zhu**, *Factorizing Probabilistic Graphical Models Using Co-occurrence Rate*, arXiv:1008.1566v1, August 9th, 2010.

Patent application

- Li Ma, Yue Pan, Chen Wang, **Zhemín Zhu** (alphabet order), *System and Method for Graph Coarsening*, US20080313251 A1, Owner: International Business Machines Corporation (IBM), 15-07-2007.

SIKS Dissertation Series

- 2015-21** Sibren Fetter (OUN), *Using Peer-Support to Expand and Stabilize Online Learning*.
2015-20 Loïs Vanhée (UU), *Using Culture and Values to Support Flexible Coordination*.
2015-19 Bernardo Tabuenca (OUN), *Ubiquitous Technology for Lifelong Learners*.
2015-18 Holger Pirk (CWI/UVA), *Waste Not, Want Not! - Managing Relational Data in Asymmetric Memories*.
2015-17 André van Cleeff (UT), *Physical and Digital Security Mechanisms: Properties, Combinations and Trade-offs*.
2015-16 Changyun Wei (TUD), *Cognitive Coordination for Cooperative Multi-Robot Teamwork*.
2015-15 Klaas Andries de Graaf (VU), *Ontology-based Software Architecture Documentation*.
2015-14 Bart van Straalen (UT), *A cognitive approach to modeling bad news conversations*.
2015-13 Giuseppe Procaccianti (VU), *Energy-Efficient Software*.
2015-12 Julie M. Birkholz (VU), *Modi Operandi of Social Network Dynamics: The Effect of Context on Scientific Collaboration Networks*.
2015-11 Yongming Luo (TUE), *Designing algorithms for big graph datasets: A study of computing bisimulation and joins*.
2015-10 Henry Hermans (OUN), *OpenUI: design of an integrated system to support lifelong learning*.
2015-09 Randy Klaassen (UT), *HCI Perspectives on Behavior Change Support Systems*.
2015-08 Jie Jiang (TUD), *Organizational Compliance: An agent-based model for designing and evaluating organizational interactions*.
2015-07 Maria-Hendrike Peetz (UvA), *Time-Aware Online Reputation Analysis*.
2015-06 Farideh Heidari (TUD), *Business Process Quality Computation - Computing Non-Functional Requirements to Improve Business Processes*.
2015-05 Christoph Bösch (UT), *Cryptographically Enforced Search Pattern Hiding*.
2015-04 Howard Spoelstra (OUN), *Collaborations in Open Learning Environments*.
2015-03 Twan van Laarhoven (RUN), *Machine learning for network data*.
2015-02 Faiza Bukhsh (UvT), *Smart auditing: Innovative Compliance Checking in Customs Controls*.
2015-01 Niels Netten (UvA), *Machine Learning for Relevance of Information in Crisis Response*.
2014-47 Shangsong Liang (UVA), *Fusion and Diversification in Information Retrieval*.
2014-46 Ke Tao (TUD), *Social Web Data Analytics: Relevance, Redundancy, Diversity*.

- 2014-45** Birgit Schmitz (OUN), *Mobile Games for Learning: A Pattern-Based Approach*.
- 2014-44** Paulien Meesters (UvT), *Intelligent Blauw. Met als ondertitel: Intelligence-gestuurde politiezorg in gebiedsgebonden eenheden..*
- 2014-43** Kevin Vlaanderen (UU), *Supporting Process Improvement using Method Increments*.
- 2014-42** Carsten Eijckhof (CWI/TUD), *Contextual Multidimensional Relevance Models*.
- 2014-41** Frederic Hogenboom (EUR), *Automated Detection of Financial Events in News Text*.
- 2014-40** Walter Omona (RUN), *A Framework for Knowledge Management Using ICT in Higher Education*.
- 2014-39** Jasmina Maric (UvT), *Web Communities, Immigration, and Social Capital*.
- 2014-38** Danny Plass-Oude Bos (UT), *Making brain-computer interfaces better: improving usability through post-processing..*
- 2014-37** Maral Dadvar (UT), *Experts and Machines United Against Cyberbullying*.
- 2014-36** Joos Buijs (TUE), *Flexible Evolutionary Algorithms for Mining Structured Process Models*.
- 2014-35** Joost van Ooijen (UU), *Cognitive Agents in Virtual Worlds: A Middleware Design Approach*.
- 2014-34** Christina Manteli (VU), *The Effect of Governance in Global Software Development: Analyzing Transactive Memory Systems..*
- 2014-33** Tesfa Tegegne (RUN), *Service Discovery in eHealth*.
- 2014-32** Naser Ayat (UvA), *On Entity Resolution in Probabilistic Data*.
- 2014-31** Leo van Moergestel (UU), *Agent Technology in Agile Multiparallel Manufacturing and Product Support*.
- 2014-30** Peter de Cock (UvT), *Anticipating Criminal Behaviour*.
- 2014-29** Jaap Kabbedijk (UU), *Variability in Multi-Tenant Enterprise Software*.
- 2014-28** Anna Chmielowiec (VU), *Decentralized k-Clique Matching*.
- 2014-27** Rui Jorge Almeida (EUR), *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*.
- 2014-26** Tim Baarslag (TUD), *What to Bid and When to Stop*.
- 2014-25** Martijn Lappenschaar (RUN), *New network models for the analysis of disease interaction*.
- 2014-24** Davide Ceolin (VU), *Trusting Semi-structured Web Data*.
- 2014-23** Eleftherios Sidirourgos (UvA/CWI), *Space Efficient Indexes for the Big Data Era*.
- 2014-22** Marieke Peeters (UU), *Personalized Educational Games - Developing agent-supported scenario-based training*.
- 2014-21** Cassidy Clark (TUD), *Negotiation and Monitoring in Open Environments*.
- 2014-20** Mena Habib (UT), *Named Entity Extraction and Disambiguation for Informal Text: The Missing Link*.
- 2014-19** Vinicius Ramos (TUE), *Adaptive Hypermedia Courses: Qualitative and Quantitative Evaluation and Tool Support*.
- 2014-18** Mattijs Ghijsen (VU), *Methods and Models for the Design and Study of Dynamic Agent Organizations*.
- 2014-17** Kathrin Dentler (VU), *Computing healthcare quality indicators automatically: Sec-*

ondary Use of Patient Data and Semantic Interoperability.

2014-16 Krystyna Milian (VU), *Supporting trial recruitment and design by automatically interpreting eligibility criteria.*

2014-15 Natalya Mogles (VU), *Agent-Based Analysis and Support of Human Functioning in Complex Socio-Technical Systems: Applications in Safety and Healthcare.*

2014-14 Yangyang Shi (TUD), *Language Models With Meta-information.*

2014-13 Arlette van Wissen (VU), *Agent-Based Support for Behavior Change: Models and Applications in Health and Safety Domains.*

2014-12 Willem van Willigen (VU), *Look Ma, No Hands: Aspects of Autonomous Vehicle Control.*

2014-11 Janneke van der Zwaan (TUD), *An Empathic Virtual Buddy for Social Support.*

2014-10 Ivan Salvador Razo Zapata (VU), *Service Value Networks.*

2014-09 Philip Jackson (UvT), *Toward Human-Level Artificial Intelligence: Representation and Computation of Meaning in Natural Language.*

2014-08 Samur Araujo (TUD), *Data Integration over Distributed and Heterogeneous Data Endpoints.*

2014-07 Arya Adriansyah (TUE), *Aligning Observed and Modeled Behavior.*

2014-06 Damian Tamburri (VU), *Supporting Networked Software Development.*

2014-05 Jurriaan van Reijssen (UU), *Knowledge Perspectives on Advancing Dynamic Capability.*

2014-04 Hanna Jochmann-Mannak (UT), *Websites for children: search strategies and interface design - Three studies on children's search performance and evaluation.*

2014-03 Sergio Raul Duarte Torres (UT), *Information Retrieval for Children: Search Behavior and Solutions.*

2014-02 Fiona Tuliayano (RUN), *Combining System Dynamics with a Domain Modeling Method.*

2014-01 Nicola Barile (UU), *Studies in Learning Monotone Models from Data.*

2013-43 Marc Bron (UVA), *Exploration and Contextualization through Interaction and Concepts.*

2013-42 Léon Planken (TUD), *Algorithms for Simple Temporal Reasoning.*

2013-41 Jochem Liem (UVA), *Supporting the Conceptual Modelling of Dynamic Systems: A Knowledge Engineering Perspective on Qualitative Reasoning.*

2013-40 Pim Nijssen (UM), *Monte-Carlo Tree Search for Multi-Player Games.*

2013-39 Joop de Jong (TUD), *A Method for Enterprise Ontology based Design of Enterprise Information Systems.*

2013-38 Eelco den Heijer (VU), *Autonomous Evolutionary Art.*

2013-37 Dirk Börner (OUN), *Ambient Learning Displays.*

2013-36 Than Lam Hoang (TUE), *Pattern Mining in Data Streams.*

2013-35 Abdallah El Ali (UvA), *Minimal Mobile Human Computer Interaction Promotor: Prof. dr. L. Hardman (CWI/UVA).*

2013-34 Kien Tjin-Kam-Jet (UT), *Distributed Deep Web Search.*

2013-33 Qi Gao (TUD), *User Modeling and Personalization in the Microblogging Sphere.*

- 2013-32** Kamakshi Rajagopal (OUN), *Networking For Learning; The role of Networking in a Lifelong Learner's Professional Development*.
- 2013-31** Dinh Khoa Nguyen (UvT), *Blueprint Model and Language for Engineering Cloud Applications*.
- 2013-30** Joyce Nakatumba (TUE), *Resource-Aware Business Process Management: Analysis and Support*.
- 2013-29** Iwan de Kok (UT), *Listening Heads*.
- 2013-28** Frans van der Sluis (UT), *When Complexity becomes Interesting: An Inquiry into the Information eXperience*.
- 2013-27** Mohammad Huq (UT), *Inference-based Framework Managing Data Provenance*.
- 2013-26** Alireza Zarghami (UT), *Architectural Support for Dynamic Homecare Service Provisioning*.
- 2013-25** Agnieszka Anna Latoszek-Berendsen (UM), *Intention-based Decision Support. A new way of representing and implementing clinical guidelines in a Decision Support System*.
- 2013-24** Haitham Bou Ammar (UM), *Automated Transfer in Reinforcement Learning*.
- 2013-23** Patricio de Alencar Silva (UvT), *Value Activity Monitoring*.
- 2013-22** Tom Claassen (RUN), *Causal Discovery and Logic*.
- 2013-21** Sander Wubben (UvT), *Text-to-text generation by monolingual machine translation*.
- 2013-20** Katja Hofmann (UvA), *Fast and Reliable Online Learning to Rank for Information Retrieval*.
- 2013-19** Renze Steenhuizen (TUD), *Coordinated Multi-Agent Planning and Scheduling*.
- 2013-18** Jeroen Janssens (UvT), *Outlier Selection and One-Class Classification*.
- 2013-17** Koen Kok (VU), *The PowerMatcher: Smart Coordination for the Smart Electricity Grid*.
- 2013-16** Eric Kok (UU), *Exploring the practical benefits of argumentation in multi-agent deliberation*.
- 2013-15** Daniel Hennes (UM), *Multiagent Learning - Dynamic Games and Applications*.
- 2013-14** Jafar Tanha (UVA), *Ensemble Approaches to Semi-Supervised Learning Learning*.
- 2013-13** Mohammad Safiri (UT), *Service Tailoring: User-centric creation of integrated IT-based homecare services to support independent living of elderly*.
- 2013-12** Marian Razavian (VU), *Knowledge-driven Migration to Services*.
- 2013-11** Evangelos Pournaras (TUD), *Multi-level Reconfigurable Self-organization in Overlay Services*.
- 2013-10** Jeewanee Jayasinghe Arachchige (UvT), *A Unified Modeling Framework for Service Design..*
- 2013-09** Fabio Gori (RUN), *Metagenomic Data Analysis: Computational Methods and Applications*.
- 2013-08** Robbert-Jan Merk (VU), *Making enemies: cognitive modeling for opponent agents in fighter pilot simulators*.
- 2013-07** Giel van Lankveld (UvT), *Quantifying Individual Player Differences*.
- 2013-06** Romulo Goncalves (CWI), *The Data Cyclotron: Juggling Data and Queries for a Data Warehouse Audience*.

- 2013-05** Dulce Pumareja (UT), *Groupware Requirements Evolutions Patterns*.
- 2013-04** Chetan Yadati (TUD), *Coordinating autonomous planning and scheduling*.
- 2013-03** Szymon Klarman (VU), *Reasoning with Contexts in Description Logics*.
- 2013-02** Erietta Liarou (CWI), *MonetDB/DataCell: Leveraging the Column-store Database Technology for Efficient and Scalable Stream Processing*.
- 2013-01** Viorel Milea (EUR), *News Analytics for Financial Decision Support*.
- 2012-51** Jeroen de Jong (TUD), *Heuristics in Dynamic Scheduling; a practical framework with a case study in elevator dispatching*.
- 2012-50** Steven van Kervel (TUD), *Ontologogy driven Enterprise Information Systems Engineering*.
- 2012-49** Michael Kaisers (UM), *Learning against Learning - Evolutionary dynamics of reinforcement learning algorithms in strategic interactions*.
- 2012-48** Jorn Bakker (TUE), *Handling Abrupt Changes in Evolving Time-series Data*.
- 2012-47** Manos Tsagkias (UVA), *Mining Social Media: Tracking Content and Predicting Behavior*.
- 2012-46** Simon Carter (UVA), *Exploration and Exploitation of Multilingual Data for Statistical Machine Translation*.
- 2012-45** Benedikt Kratz (UvT), *A Model and Language for Business-aware Transactions*.
- 2012-44** Anna Tordai (VU), *On Combining Alignment Techniques*.
- 2012-43** Withdrawn.
- 2012-42** Dominique Verpoorten (OU), *Reflection Amplifiers in self-regulated Learning*.
- 2012-41** Sebastian Kelle (OU), *Game Design Patterns for Learning*.
- 2012-40** Agus Gunawan (UvT), *Information Access for SMEs in Indonesia*.
- 2012-39** Hassan Fatemi (UT), *Risk-aware design of value and coordination networks*.
- 2012-38** Selmar Smit (VU), *Parameter Tuning and Scientific Testing in Evolutionary Algorithms*.
- 2012-37** Agnes Nakakawa (RUN), *A Collaboration Process for Enterprise Architecture Creation*.
- 2012-36** Denis Ssebugwawo (RUN), *Analysis and Evaluation of Collaborative Modeling Processes*.
- 2012-35** Evert Haasdijk (VU), *Never Too Old To Learn – On-line Evolution of Controllers in Swarm- and Modular Robotics*.
- 2012-34** Pavol Jancura (RUN), *Evolutionary analysis in PPI networks and applications*.
- 2012-33** Rory Sie (OUN), *Coalitions in Cooperation Networks (COCOON)*.
- 2012-32** Wietske Visser (TUD), *Qualitative multi-criteria preference representation and reasoning*.
- 2012-31** Emily Bagarukayo (RUN), *A Learning by Construction Approach for Higher Order Cognitive Skills Improvement, Building Capacity and Infrastructure*.
- 2012-30** Alina Pommeranz (TUD), *Designing Human-Centered Systems for Reflective Decision Making*.
- 2012-29** Almer Tigelaar (UT), *Peer-to-Peer Information Retrieval*.
- 2012-28** Nancy Pascall (UvT), *Engendering Technology Empowering Women*.

- 2012-27** Hayrettin Gurkok (UT), *Mind the Sheep! User Experience Evaluation & Brain-Computer Interface Games.*
- 2012-26** Emile de Maat (UVA), *Making Sense of Legal Text.*
- 2012-25** Silja Eckartz (UT), *Managing the Business Case Development in Inter-Organizational IT Projects: A Methodology and its Application.*
- 2012-24** Laurens van der Werff (UT), *Evaluation of Noisy Transcripts for Spoken Document Retrieval.*
- 2012-23** Christian Muehl (UT), *Toward Affective Brain-Computer Interfaces: Exploring the Neurophysiology of Affect during Human Media Interaction.*
- 2012-22** Thijs Vis (UvT), *Intelligence, politie en veiligheidsdienst: verenigbare grootheden?.*
- 2012-21** Roberto Cornacchia (TUD), *Querying Sparse Matrices for Information Retrieval.*
- 2012-20** Ali Bahramisharif (RUN), *Covert Visual Spatial Attention, a Robust Paradigm for Brain-Computer Interfacing.*
- 2012-19** Helen Schonenberg (TUE), *What's Next? Operational Support for Business Process Execution.*
- 2012-18** Eltjo Poort (VU), *Improving Solution Architecting Practices.*
- 2012-17** Amal Elgammal (UvT), *Towards a Comprehensive Framework for Business Process Compliance.*
- 2012-16** Fiemke Both (VU), *Helping people by understanding them - Ambient Agents supporting task execution and depression treatment.*
- 2012-15** Natalie van der Wal (VU), *Social Agents. Agent-Based Modelling of Integrated Internal and Social Dynamics of Cognitive and Affective Processes..*
- 2012-14** Evgeny Knutov (TUE), *Generic Adaptation Framework for Unifying Adaptive Web-based Systems.*
- 2012-13** Suleman Shahid (UvT), *Fun and Face: Exploring non-verbal expressions of emotion during playful interactions.*
- 2012-12** Kees van der Sluijs (TUE), *Model Driven Design and Data Integration in Semantic Web Information Systems.*
- 2012-11** J.C.B. Rantham Prabhakara (TUE), *Process Mining in the Large: Preprocessing, Discovery, and Diagnostics.*
- 2012-10** David Smits (TUE), *Towards a Generic Distributed Adaptive Hypermedia Environment.*
- 2012-09** Ricardo Neisse (UT), *Trust and Privacy Management Support for Context-Aware Service Platforms.*
- 2012-08** Gerben de Vries (UVA), *Kernel Methods for Vessel Trajectories.*
- 2012-07** Rianne van Lambalgen (VU), *When the Going Gets Tough: Exploring Agent-based Models of Human Performance under Demanding Conditions.*
- 2012-06** Wolfgang Reinhardt (OU), *Awareness Support for Knowledge Workers in Research Networks.*
- 2012-05** Marijn Plomp (UU), *Maturing Interorganisational Information Systems.*
- 2012-04** Jurriaan Souer (UU), *Development of Content Management System-based Web Applications.*

- 2012-03** Adam Vanya (VU), *Supporting Architecture Evolution by Mining Software Repositories*.
- 2012-02** Muhammad Umair (VU), *Adaptivity, emotion, and Rationality in Human and Ambient Agent Models*.
- 2012-01** Terry Kakeeto (UvT), *Relationship Marketing for SMEs in Uganda*.
- 2011-49** Andreea Niculescu (UT), *Conversational interfaces for task-oriented spoken dialogues: design aspects influencing interaction quality*.
- 2011-48** Mark Ter Maat (UT), *Response Selection and Turn-taking for a Sensitive Artificial Listening Agent*.
- 2011-47** Azizi Bin Ab Aziz (VU), *Exploring Computational Models for Intelligent Support of Persons with Depression*.
- 2011-46** Beibei Hu (TUD), *Towards Contextualized Information Delivery: A Rule-based Architecture for the Domain of Mobile Police Work*.
- 2011-45** Herman Stehouwer (UvT), *Statistical Language Models for Alternative Sequence Selection*.
- 2011-44** Boris Reuderink (UT), *Robust Brain-Computer Interfaces*.
- 2011-43** Henk van der Schuur (UU), *Process Improvement through Software Operation Knowledge*.
- 2011-42** Michal Sindlar (UU), *Explaining Behavior through Mental State Attribution*.
- 2011-41** Luan Ibraimi (UT), *Cryptographically Enforced Distributed Data Access Control*.
- 2011-40** Viktor Clerc (VU), *Architectural Knowledge Management in Global Software Development*.
- 2011-39** Joost Westra (UU), *Organizing Adaptation using Agents in Serious Games*.
- 2011-38** Nyree Lemmens (UM), *Bee-inspired Distributed Optimization*.
- 2011-37** Adriana Burlutiu (RUN), *Machine Learning for Pairwise Data, Applications for Preference Learning and Supervised Network Inference*.
- 2011-36** Erik van der Spek (UU), *Experiments in serious game design: a cognitive approach*.
- 2011-35** Maaïke Harbers (UU), *Explaining Agent Behavior in Virtual Training*.
- 2011-34** Paolo Turrini (UU), *Strategic Reasoning in Interdependence: Logical and Game-theoretical Investigations*.
- 2011-33** Tom van der Weide (UU), *Arguing to Motivate Decisions*.
- 2011-32** Nees-Jan van Eck (EUR), *Methodological Advances in Bibliometric Mapping of Science*.
- 2011-31** Ludo Waltman (EUR), *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*.
- 2011-30** Egon van den Broek (UT), *Affective Signal Processing (ASP): Unraveling the mystery of emotions*.
- 2011-29** Faisal Kamiran (TUE), *Discrimination-aware Classification*.
- 2011-28** Rianne Kaptein (UVA), *Effective Focused Retrieval by Exploiting Query Context and Document Structure*.
- 2011-27** Aniel Bhulai (VU), *Dynamic website optimization through autonomous management of design patterns*.

- 2011-26** Matthijs Aart Pontier (VU), *Virtual Agents for Human Communication - Emotion Regulation and Involvement-Distance Trade-Offs in Embodied Conversational Agents and Robots.*
- 2011-25** Syed Waqar ul Qounain Jaffry (VU), *Analysis and Validation of Models for Trust Dynamics.*
- 2011-24** Herwin van Welbergen (UT), *Behavior Generation for Interpersonal Coordination with Virtual Humans On Specifying, Scheduling and Realizing Multimodal Virtual Human Behavior.*
- 2011-23** Wouter Weerkamp (UVA), *Finding People and their Utterances in Social Media.*
- 2011-22** Junte Zhang (UVA), *System Evaluation of Archival Description and Access.*
- 2011-21** Linda Terlouw (TUD), *Modularization and Specification of Service-Oriented Systems.*
- 2011-20** Qing Gu (VU), *Guiding service-oriented software engineering - A view-based approach.*
- 2011-19** Ellen Rusman (OU), *The Mind 's Eye on Personal Profiles.*
- 2011-18** Mark Ponsen (UM), *Strategic Decision-Making in complex games.*
- 2011-17** Jiyin He (UVA), *Exploring Topic Structure: Coherence, Diversity and Relatedness.*
- 2011-16** Maarten Schadd (UM), *Selective Search in Games of Different Complexity.*
- 2011-15** Marijn Koolen (UvA), *The Meaning of Structure: the Value of Link Evidence for Information Retrieval.*
- 2011-14** Milan Lovric (EUR), *Behavioral Finance and Agent-Based Artificial Markets.*
- 2011-13** Xiaoyu Mao (UvT), *Airport under Control. Multiagent Scheduling for Airport Ground Handling.*
- 2011-12** Carmen Bratosin (TUE), *Grid Architecture for Distributed Process Mining.*
- 2011-11** Dhaval Vyas (UT), *Designing for Awareness: An Experience-focused HCI Perspective.*
- 2011-10** Bart Bogaert (UvT), *Cloud Content Contention.*
- 2011-09** Tim de Jong (OU), *Contextualised Mobile Media for Learning.*
- 2011-08** Nieske Vergunst (UU), *BDI-based Generation of Robust Task-Oriented Dialogues.*
- 2011-07** Yujia Cao (UT), *Multimodal Information Presentation for High Load Human Computer Interaction.*
- 2011-06** Yiwen Wang (TUE), *Semantically-Enhanced Recommendations in Cultural Heritage.*
- 2011-05** Base van der Raadt (VU), *Enterprise Architecture Coming of Age - Increasing the Performance of an Emerging Discipline..*
- 2011-04** Hado van Hasselt (UU), *Insights in Reinforcement Learning; Formal analysis and empirical evaluation of temporal-difference learning algorithms.*
- 2011-03** Jan Martijn van der Werf (TUE), *Compositional Design and Verification of Component-Based Information Systems.*
- 2011-02** Nick Tinnemeier (UU), *Organizing Agent Organizations. Syntax and Operational Semantics of an Organization-Oriented Programming Language.*
- 2011-01** Botond Cseke (RUN), *Variational Algorithms for Bayesian Inference in Latent Gaussian Models.*
- 2010-53** Edgar Meij (UVA), *Combining Concepts and Language Models for Information Access.*
- 2010-52** Peter-Paul van Maanen (VU), *Adaptive Support for Human-Computer Teams: Exploring the Use of Cognitive Models of Trust and Attention.*

- 2010-51** Alia Khairia Amin (CWI), *Understanding and supporting information seeking tasks in multiple sources.*
- 2010-50** Bouke Huurnink (UVA), *Search in Audiovisual Broadcast Archives.*
- 2010-49** Jahn-Takeshi Saito (UM), *Solving difficult game positions.*
- 2010-48** Withdrawn.
- 2010-47** Chen Li (UT), *Mining Process Model Variants: Challenges, Techniques, Examples.*
- 2010-46** Vincent Pijpers (VU), *e3alignment: Exploring Inter-Organizational Business-ICT Alignment.*
- 2010-45** Vasilios Andrikopoulos (UvT), *A theory and model for the evolution of software services.*
- 2010-44** Pieter Bellekens (TUE), *An Approach towards Context-sensitive and User-adapted Access to Heterogeneous Data Sources, Illustrated in the Television Domain.*
- 2010-43** Peter van Kranenburg (UU), *A Computational Approach to Content-Based Retrieval of Folk Song Melodies.*
- 2010-42** Sybren de Kinderen (VU), *Needs-driven service bundling in a multi-supplier setting - the computational e3-service approach.*
- 2010-41** Guillaume Chaslot (UM), *Monte-Carlo Tree Search.*
- 2010-40** Mark van Assem (VU), *Converting and Integrating Vocabularies for the Semantic Web.*
- 2010-39** Ghazanfar Farooq Siddiqui (VU), *Integrative modeling of emotions in virtual agents.*
- 2010-38** Dirk Fahland (TUE), *From Scenarios to components.*
- 2010-37** Niels Lohmann (TUE), *Correctness of services and their composition.*
- 2010-36** Jose Janssen (OU), *Paving the Way for Lifelong Learning; Facilitating competence development through a learning path specification.*
- 2010-35** Dolf Trieschnigg (UT), *Proof of Concept: Concept-based Biomedical Information Retrieval.*
- 2010-34** Teduh Dirgahayu (UT), *Interaction Design in Service Compositions.*
- 2010-33** Robin Aly (UT), *Modeling Representation Uncertainty in Concept-Based Multimedia Retrieval.*
- 2010-32** Marcel Hiel (UvT), *An Adaptive Service Oriented Architecture: Automatically solving Interoperability Problems.*
- 2010-31** Victor de Boer (UVA), *Ontology Enrichment from Heterogeneous Sources on the Web.*
- 2010-30** Marieke van Erp (UvT), *Accessing Natural History - Discoveries in data cleaning, structuring, and retrieval.*
- 2010-29** Stratos Idreos (CWI), *Database Cracking: Towards Auto-tuning Database Kernels.*
- 2010-28** Arne Koopman (UU), *Characteristic Relational Patterns.*
- 2010-27** Marten Voulon (UL), *Automatisch contracteren.*
- 2010-26** Ying Zhang (CWI), *XRPC: Efficient Distributed Query Processing on Heterogeneous XQuery Engines.*
- 2010-25** Zulfiqar Ali Memon (VU), *Modelling Human-Awareness for Ambient Agents: A Human Mindreading Perspective.*
- 2010-24** Dmytro Tykhonov, *Designing Generic and Efficient Negotiation Strategies.*

- 2010-23** Bas Steunebrink (UU), *The Logical Structure of Emotions*.
- 2010-22** Michiel Hildebrand (CWI), *End-user Support for Access to Heterogeneous Linked Data*.
- 2010-21** Harold van Heerde (UT), *Privacy-aware data management by means of data degradation*.
- 2010-20** Ivo Swartjes (UT), *Whose Story Is It Anyway? How Improv Informs Agency and Authorship of Emergent Narrative*.
- 2010-19** Henriette Cramer (UvA), *People's Responses to Autonomous and Adaptive Systems*.
- 2010-18** Charlotte Gerritsen (VU), *Caught in the Act: Investigating Crime by Agent-Based Simulation*.
- 2010-17** Spyros Kotoulas (VU), *Scalable Discovery of Networked Resources: Algorithms, Infrastructure, Applications*.
- 2010-16** Sicco Verwer (TUD), *Efficient Identification of Timed Automata, theory and practice*.
- 2010-15** Lianne Bodestaff (UT), *Managing Dependency Relations in Inter-Organizational Models*.
- 2010-14** Sander van Splunter (VU), *Automated Web Service Reconfiguration*.
- 2010-13** Gianluigi Folino (RUN), *High Performance Data Mining using Bio-inspired techniques*.
- 2010-12** Susan van den Braak (UU), *Sensemaking software for crime analysis*.
- 2010-11** Adriaan Ter Mors (TUD), *The world according to MARP: Multi-Agent Route Planning*.
- 2010-10** Rebecca Ong (UL), *Mobile Communication and Protection of Children*.
- 2010-09** Hugo Kielman (UL), *A Politiele gegevensverwerking en Privacy, Naar een effectieve waarborging*.
- 2010-08** Krzysztof Siewicz (UL), *Towards an Improved Regulatory Framework of Free Software. Protecting user freedoms in a world of software communities and eGovernments*.
- 2010-07** Wim Fikkert (UT), *Gesture interaction at a Distance*.
- 2010-06** Sander Bakkes (UvT), *Rapid Adaptation of Video Game AI*.
- 2010-05** Claudia Hauff (UT), *Predicting the Effectiveness of Queries and Retrieval Systems*.
- 2010-04** Olga Kulyk (UT), *Do You Know What I Know? Situational Awareness of Co-located Teams in Multidisplay Environments*.
- 2010-03** Joost Geurts (CWI), *A Document Engineering Model and Processing Framework for Multimedia documents*.
- 2010-02** Ingo Wassink (UT), *Work flows in Life Science*.
- 2010-01** Matthijs van Leeuwen (UU), *Patterns that Matter*.
- 2009-46** Loredana Afanasiev (UvA), *Querying XML: Benchmarks and Recursion*.
- 2009-45** Jilles Vreeken (UU), *Making Pattern Mining Useful*.
- 2009-44** Roberto Santana Tapia (UT), *Assessing Business-IT Alignment in Networked Organizations*.
- 2009-43** Virginia Nunes Leal Franqueira (UT), *Finding Multi-step Attacks in Computer Networks using Heuristic Search and Mobile Ambients*.
- 2009-42** Toine Bogers (UvT), *Recommender Systems for Social Bookmarking*.

- 2009-41** Igor Berezhnny (UvT), *Digital Analysis of Paintings*.
- 2009-40** Stephan Raaijmakers (UvT), *Multinomial Language Learning: Investigations into the Geometry of Language*.
- 2009-39** Christian Stahl (TUE, Humboldt-Universitaet zu Berlin), *Service Substitution – A Behavioral Approach Based on Petri Nets*.
- 2009-38** Riina Vuorikari (OU), *Tags and self-organisation: a metadata ecology for learning resources in a multilingual context*.
- 2009-37** Hendrik Drachsler (OUN), *Navigation Support for Learners in Informal Learning Networks*.
- 2009-36** Marco Kalz (OUN), *Placement Support for Learners in Learning Networks*.
- 2009-35** Wouter Koelewijn (UL), *Privacy en Politiegegevens; Over geautomatiseerde normatieve informatie-uitwisseling*.
- 2009-34** Inge van de Weerd (UU), *Advancing in Software Product Management: An Incremental Method Engineering Approach*.
- 2009-33** Khiet Truong (UT), *How Does Real Affect Affect Affect Recognition In Speech?*.
- 2009-32** Rik Farenhorst (VU) and Remco de Boer (VU), *Architectural Knowledge Management: Supporting Architects and Auditors*.
- 2009-31** Sofiya Katrenko (UVA), *A Closer Look at Learning Relations from Text*.
- 2009-30** Marcin Zukowski (CWI), *Balancing vectorized query execution with bandwidth-optimized storage*.
- 2009-29** Stanislav Pokraev (UT), *Model-Driven Semantic Integration of Service-Oriented Applications*.
- 2009-28** Sander Evers (UT), *Sensor Data Management with Probabilistic Models*.
- 2009-27** Christian Glahn (OU), *Contextual Support of social Engagement and Reflection on the Web*.
- 2009-26** Fernando Koch (UU), *An Agent-Based Model for the Development of Intelligent Mobile Services*.
- 2009-25** Alex van Ballegooij (CWI), *"RAM: Array Database Management through Relational Mapping"*.
- 2009-24** Annerieke Heuvelink (VUA), *Cognitive Models for Training Simulations*.
- 2009-23** Peter Hofgesang (VU), *Modelling Web Usage in a Changing Environment*.
- 2009-22** Pavel Serdyukov (UT), *Search For Expertise: Going beyond direct evidence*.
- 2009-21** Stijn Vanderlooy (UM), *Ranking and Reliable Classification*.
- 2009-20** Bob van der Vecht (UU), *Adjustable Autonomy: Controlling Influences on Decision Making*.
- 2009-19** Valentin Robu (CWI), *Modeling Preferences, Strategic Reasoning and Collaboration in Agent-Mediated Electronic Markets*.
- 2009-18** Fabian Groffen (CWI), *Armada, An Evolving Database System*.
- 2009-17** Laurens van der Maaten (UvT), *Feature Extraction from Visual Data*.
- 2009-16** Fritz Reul (UvT), *New Architectures in Computer Chess*.
- 2009-15** Rinke Hoekstra (UVA), *Ontology Representation - Design Patterns and Ontologies that Make Sense*.

- 2009-14** Maksym Korotkiy (VU), *From ontology-enabled services to service-enabled ontologies (making ontologies work in e-science with ONTO-SOA)*.
- 2009-13** Steven de Jong (UM), *Fairness in Multi-Agent Systems*.
- 2009-12** Peter Massuthe (TUE, Humboldt-Universitaet zu Berlin), *Operating Guidelines for Services*.
- 2009-11** Alexander Boer (UVA), *Legal Theory, Sources of Law & the Semantic Web*.
- 2009-10** Jan Wielemaker (UVA), *Logic programming for knowledge-intensive interactive applications*.
- 2009-09** Benjamin Kanagwa (RUN), *Design, Discovery and Construction of Service-oriented Systems*.
- 2009-08** Volker Nannen (VU), *Evolutionary Agent-Based Policy Analysis in Dynamic Environments*.
- 2009-07** Ronald Poppe (UT), *Discriminative Vision-Based Recovery and Recognition of Human Motion*.
- 2009-06** Muhammad Subianto (UU), *Understanding Classification*.
- 2009-05** Sietse Overbeek (RUN), *Bridging Supply and Demand for Knowledge Intensive Tasks - Based on Knowledge, Cognition, and Quality*.
- 2009-04** Josephine Nabukenya (RUN), *Improving the Quality of Organisational Policy Making using Collaboration Engineering*.
- 2009-03** Hans Stol (UvT), *A Framework for Evidence-based Policy Making Using IT*.
- 2009-02** Willem Robert van Hage (VU), *Evaluating Ontology-Alignment Techniques*.
- 2009-01** Rasa Jurgelenaite (RUN), *Symmetric Causal Independence Models*.
- 2008-35** Ben Torben Nielsen (UvT), *Dendritic morphologies: function shapes structure*.
- 2008-34** Jeroen de Knijf (UU), *Studies in Frequent Tree Mining*.
- 2008-33** Frank Terpstra (UVA), *Scientific Workflow Design; theoretical and practical issues*.
- 2008-32** Trung H. Bui (UT), *Toward Affective Dialogue Management using Partially Observable Markov Decision Processes*.
- 2008-31** Loes Braun (UM), *Pro-Active Medical Information Retrieval*.
- 2008-30** Wouter van Atteveldt (VU), *Semantic Network Analysis: Techniques for Extracting, Representing and Querying Media Content*.
- 2008-29** Dennis Reidsma (UT), *Annotations and Subjective Machines - Of Annotators, Embodied Agents, Users, and Other Humans*.
- 2008-28** Ildiko Flesch (RUN), *On the Use of Independence Relations in Bayesian Networks*.
- 2008-27** Hubert Vogten (OU), *Design and Implementation Strategies for IMS Learning Design*.
- 2008-26** Marijn Huijbregts (UT), *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled*.
- 2008-25** Geert Jonker (UU), *Efficient and Equitable Exchange in Air Traffic Management Plan Repair using Spender-signed Currency*.
- 2008-24** Zharko Aleksovski (VU), *Using background knowledge in ontology matching*.
- 2008-23** Stefan Visscher (UU), *Bayesian network models for the management of ventilator-associated pneumonia*.

- 2008-22** Henk Koning (UU), *Communication of IT-Architecture*.
- 2008-21** Krisztian Balog (UVA), *People Search in the Enterprise*.
- 2008-20** Rex Arendsen (UVA), *Geen bericht, goed bericht. Een onderzoek naar de effecten van de introductie van elektronisch berichtenverkeer met de overheid op de administratieve lasten van bedrijven..*
- 2008-19** Henning Rode (UT), *From Document to Entity Retrieval: Improving Precision and Performance of Focused Text Search*.
- 2008-18** Guido de Croon (UM), *Adaptive Active Vision*.
- 2008-17** Martin Op 't Land (TUD), *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*.
- 2008-16** Henriette van Vugt (VU), *Embodied agents from a user's perspective*.
- 2008-15** Martijn van Otterlo (UT), *The Logic of Adaptive Behavior: Knowledge Representation and Algorithms for the Markov Decision Process Framework in First-Order Domains..*
- 2008-14** Arthur van Bunningen (UT), *Context-Aware Querying; Better Answers with Less Effort*.
- 2008-13** Caterina Carraciolo (UVA), *Topic Driven Access to Scientific Handbooks*.
- 2008-12** Jozsef Farkas (RUN), *A Semiotically Oriented Cognitive Model of Knowledge Representation*.
- 2008-11** Vera Kartseva (VU), *Designing Controls for Network Organizations: A Value-Based Approach*.
- 2008-10** Wauter Bosma (UT), *Discourse oriented summarization*.
- 2008-09** Christof van Nimwegen (UU), *The paradox of the guided user: assistance can be counter-effective*.
- 2008-08** Janneke Bolt (UU), *Bayesian Networks: Aspects of Approximate Inference*.
- 2008-07** Peter van Rosmalen (OU), *Supporting the tutor in the design and support of adaptive e-learning*.
- 2008-06** Arjen Hommersom (RUN), *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*.
- 2008-05** Bela Mutschler (UT), *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*.
- 2008-04** Ander de Keijzer (UT), *Management of Uncertain Data - towards unattended integration*.
- 2008-03** Vera Hollink (UVA), *Optimizing hierarchical menus: a usage-based approach*.
- 2008-02** Alexei Sharpanskykh (VU), *On Computer-Aided Methods for Modeling and Analysis of Organizations*.
- 2008-01** Katalin Boer-Sorbán (EUR), *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*.
- 2007-25** Joost Schalken (VU), *Empirical Investigations in Software Process Improvement*.
- 2007-24** Georgina Ramírez Camps (CWI), *Structural Features in XML Retrieval*.
- 2007-23** Peter Barna (TUE), *Specification of Application Logic in Web Information Systems*.
- 2007-22** Zlatko Zlatev (UT), *Goal-oriented design of value and process models from patterns*.
- 2007-21** Karianne Vermaas (UU), *Fast diffusion and broadening use: A research on residential*

adoption and usage of broadband internet in the Netherlands between 2001 and 2005.

2007-20 Slinger Jansen (UU), *Customer Configuration Updating in a Software Supply Network.*

2007-19 David Levy (UM), *Intimate relationships with artificial partners.*

2007-18 Bart Orriens (UvT), *On the development an management of adaptive business collaborations.*

2007-17 Theodore Charitos (UU), *Reasoning with Dynamic Networks in Practice.*

2007-16 Davide Grossi (UU), *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems.*

2007-15 Joyca Lacroix (UM), *NIM: a Situated Computational Memory Model.*

2007-14 Niek Bergboer (UM), *Context-Based Image Analysis.*

2007-13 Rutger Rienks (UT), *Meetings in Smart Environments; Implications of Progressing Technology.*

2007-12 Marcel van Gerven (RUN), *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty.*

2007-11 Natalia Stash (TUE), *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System.*

2007-10 Huib Aldewereld (UU), *Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols.*

2007-09 David Mobach (VU), *Agent-Based Mediated Service Negotiation.*

2007-08 Mark Hoogendoorn (VU), *Modeling of Change in Multi-Agent Organizations.*

2007-07 Natasa Jovanovic (UT), *To Whom It May Concern - Addressee Identification in Face-to-Face Meetings.*

2007-06 Gilad Mishne (UVA), *Applied Text Analytics for Blogs.*

2007-05 Bart Schermer (UL), *Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance.*

2007-04 Jurriaan van Diggelen (UU), *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach.*

2007-03 Peter Mika (VU), *Social Networks and the Semantic Web.*

2007-02 Wouter Teepe (RUG), *Reconciling Information Exchange and Confidentiality: A Formal Approach.*

2007-01 Kees Leune (UvT), *Access Control and Service-Oriented Architectures.*

2006-28 Borkur Sigurbjornsson (UVA), *Focused Information Access using XML Element Retrieval.*

2006-27 Stefano Bocconi (CWI), *Vox Populi: generating video documentaries from semantically annotated media repositories.*

2006-26 Vojkan Mihajlovic (UT), *Score Region Algebra: A Flexible Framework for Structured Information Retrieval.*

2006-25 Madalina Drugan (UU), *Conditional log-likelihood MDL and Evolutionary MCMC.*

2006-24 Laura Hollink (VU), *Semantic Annotation for Retrieval of Visual Resources.*

2006-23 Ion Juvina (UU), *Development of Cognitive Model for Navigating on the Web.*

2006-22 Paul de Vrieze (RUN), *Fundaments of Adaptive Personalisation.*

- 2006-21** Bas van Gils (RUN), *Aptness on the Web*.
- 2006-20** Marina Velikova (UvT), *Monotone models for prediction in data mining*.
- 2006-19** Birna van Riemsdijk (UU), *Cognitive Agent Programming: A Semantic Approach*.
- 2006-18** Valentin Zhizhkun (UVA), *Graph transformation for Natural Language Processing*.
- 2006-17** Stacey Nagata (UU), *User Assistance for Multitasking with Interruptions on a Mobile Device*.
- 2006-16** Carsten Riggelsen (UU), *Approximation Methods for Efficient Learning of Bayesian Networks*.
- 2006-15** Rainer Malik (UU), *CONAN: Text Mining in the Biomedical Domain*.
- 2006-14** Johan Hoorn (VU), *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change*.
- 2006-13** Henk-Jan Lebbink (UU), *Dialogue and Decision Games for Information Exchanging Agents*.
- 2006-12** Bert Bongers (VU), *Interactivation - Towards an e-cology of people, our technological environment, and the arts*.
- 2006-11** Joeri van Ruth (UT), *Flattening Queries over Nested Data Types*.
- 2006-10** Ronny Siebes (VU), *Semantic Routing in Peer-to-Peer Systems*.
- 2006-09** Mohamed Wahdan (UM), *Automatic Formulation of the Auditor's Opinion*.
- 2006-08** Eelco Herder (UT), *Forward, Back and Home Again - Analyzing User Behavior on the Web*.
- 2006-07** Marko Smiljanic (UT), *XML schema matching – balancing efficiency and effectiveness by means of clustering*.
- 2006-06** Ziv Baida (VU), *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*.
- 2006-05** Cees Pierik (UU), *Validation Techniques for Object-Oriented Proof Outlines*.
- 2006-04** Marta Sabou (VU), *Building Web Service Ontologies*.
- 2006-03** Noor Christoph (UVA), *The role of metacognitive skills in learning to solve problems*.
- 2006-02** Cristina Chisalita (VU), *Contextual issues in the design and use of information technology in organizations*.
- 2006-01** Samuil Angelov (TUE), *Foundations of B2B Electronic Contracting*.
- 2005-21** Wijnand Derks (UT), *Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics*.
- 2005-20** Cristina Coteanu (UL), *Cyber Consumer Law, State of the Art and Perspectives*.
- 2005-19** Michel van Dartel (UM), *Situated Representation*.
- 2005-18** Danielle Sent (UU), *Test-selection strategies for probabilistic networks*.
- 2005-17** Boris Shishkov (TUD), *Software Specification Based on Re-usable Business Components*.
- 2005-16** Joris Graaumans (UU), *Usability of XML Query Languages*.
- 2005-15** Tibor Bosse (VU), *Analysis of the Dynamics of Cognitive Processes*.
- 2005-14** Borys Omelayenko (VU), *Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics*.
- 2005-13** Fred Hamburg (UL), *Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen*.

gen.

2005-12 Csaba Boer (EUR), *Distributed Simulation in Industry*.

2005-11 Elth Ogston (VU), *Agent Based Matchmaking and Clustering - A Decentralized Approach to Search*.

2005-10 Anders Bouwer (UVA), *Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments*.

2005-09 Jeen Broekstra (VU), *Storage, Querying and Inferencing for Semantic Web Languages*.

2005-08 Richard Vdovjak (TUE), *A Model-driven Approach for Building Distributed Ontology-based Web Applications*.

2005-07 Flavius Frasincar (TUE), *Hypermedia Presentation Generation for Semantic Web Information Systems*.

2005-06 Pieter Spronck (UM), *Adaptive Game AI*.

2005-05 Gabriel Infante-Lopez (UVA), *Two-Level Probabilistic Grammars for Natural Language Parsing*.

2005-04 Nirvana Meratnia (UT), *Towards Database Support for Moving Object data*.

2005-03 Franc Grootjen (RUN), *A Pragmatic Approach to the Conceptualisation of Language*.

2005-02 Erik van der Werf (UM)), *AI techniques for the game of Go*.

2005-01 Floor Verdenius (UVA), *Methodological Aspects of Designing Induction-Based Applications*.

2004-20 Madelon Evers (Nyenrode), *Learning from Design: facilitating multidisciplinary design teams*.

2004-19 Thijs Westerveld (UT), *Using generative probabilistic models for multimedia retrieval*.

2004-18 Vania Bessa Machado (UvA), *Supporting the Construction of Qualitative Knowledge Models*.

2004-17 Mark Winands (UM), *Informed Search in Complex Games*.

2004-16 Federico Divina (VU), *Hybrid Genetic Relational Search for Inductive Learning*.

2004-15 Arno Knobbe (UU), *Multi-Relational Data Mining*.

2004-14 Paul Harrenstein (UU), *Logic in Conflict. Logical Explorations in Strategic Equilibrium*.

2004-13 Wojciech Jamroga (UT), *Using Multiple Models of Reality: On Agents who Know how to Play*.

2004-12 The Duy Bui (UT), *Creating emotions and facial expressions for embodied agents*.

2004-11 Michel Klein (VU), *Change Management for Distributed Ontologies*.

2004-10 Suzanne Kabel (UVA), *Knowledge-rich indexing of learning-objects*.

2004-09 Martin Caminada (VU), *For the Sake of the Argument; explorations into argument-based reasoning*.

2004-08 Joop Verbeek (UM), *Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politile gegevensuitwisseling en digitale expertise*.

2004-07 Elise Boltjes (UM), *Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes*.

2004-06 Bart-Jan Hommes (TUD), *The Evaluation of Business Process Modeling Techniques*.

2004-05 Viara Popova (EUR), *Knowledge discovery and monotonicity*.

- 2004-04** Chris van Aart (UVA), *Organizational Principles for Multi-Agent Architectures*.
- 2004-03** Perry Groot (VU), *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*.
- 2004-02** Lai Xu (UvT), *Monitoring Multi-party Contracts for E-business*.
- 2004-01** Virginia Dignum (UU), *A Model for Organizational Interaction: Based on Agents, Founded in Logic*.
- 2003-18** Levente Kocsis (UM), *Learning Search Decisions*.
- 2003-17** David Jansen (UT), *Extensions of Statecharts with Probability, Time, and Stochastic Timing*.
- 2003-16** Menzo Windhouwer (CWI), *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses*.
- 2003-15** Mathijs de Weerd (TUD), *Plan Merging in Multi-Agent Systems*.
- 2003-14** Stijn Hoppenbrouwers (KUN), *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*.
- 2003-13** Jeroen Donkers (UM), *Nosce Hostem - Searching with Opponent Models*.
- 2003-12** Roeland Ordelman (UT), *Dutch speech recognition in multimedia information retrieval*.
- 2003-11** Simon Keizer (UT), *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*.
- 2003-10** Andreas Lincke (UvT), *Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture*.
- 2003-09** Rens Kortmann (UM), *The resolution of visually guided behaviour*.
- 2003-08** Yongping Ran (UM), *Repair Based Scheduling*.
- 2003-07** Machiel Jansen (UvA), *Formal Explorations of Knowledge Intensive Tasks*.
- 2003-06** Boris van Schooten (UT), *Development and specification of virtual environments*.
- 2003-05** Jos Lehmann (UVA), *Causation in Artificial Intelligence and Law - A modelling approach*.
- 2003-04** Milan Petkovic (UT), *Content-Based Video Retrieval Supported by Database Technology*.
- 2003-03** Martijn Schuemie (TUD), *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*.
- 2003-02** Jan Broersen (VU), *Modal Action Logics for Reasoning About Reactive Systems*.
- 2003-01** Heiner Stuckenschmidt (VU), *Ontology-Based Information Sharing in Weakly Structured Environments*.
- 2002-17** Stefan Manegold (UVA), *Understanding, Modeling, and Improving Main-Memory Database Performance*.
- 2002-16** Pieter van Langen (VU), *The Anatomy of Design: Foundations, Models and Applications*.
- 2002-15** Rik Eshuis (UT), *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*.
- 2002-14** Wieke de Vries (UU), *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems*.

- 2002-13** Hongjing Wu (TUE), *A Reference Architecture for Adaptive Hypermedia Applications*.
- 2002-12** Albrecht Schmidt (Uva), *Processing XML in Database Systems*.
- 2002-11** Wouter C.A. Wijngaards (VU), *Agent Based Modelling of Dynamics: Biological and Organisational Applications*.
- 2002-10** Brian Sheppard (UM), *Towards Perfect Play of Scrabble*.
- 2002-09** Willem-Jan van den Heuvel (KUB), *Integrating Modern Business Applications with Objectified Legacy Systems*.
- 2002-08** Jaap Gordijn (VU), *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas*.
- 2002-07** Peter Boncz (CWI), *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications*.
- 2002-06** Laurens Mommers (UL), *Applied legal epistemology; Building a knowledge-based ontology of the legal domain*.
- 2002-05** Radu Serban (VU), *The Private Cyberspace Modeling Electronic Environments inhabited by Privacy-concerned Agents*.
- 2002-04** Juan Roberto Castelo Valdueza (UU), *The Discrete Acyclic Digraph Markov Model in Data Mining*.
- 2002-03** Henk Ernst Blok (UT), *Database Optimization Aspects for Information Retrieval*.
- 2002-02** Roelof van Zwol (UT), *Modelling and searching web-based document collections*.
- 2002-01** Nico Lassing (VU), *Architecture-Level Modifiability Analysis*.
- 2001-11** Tom M. van Engers (VUA), *Knowledge Management: The Role of Mental Models in Business Systems Design*.
- 2001-10** Maarten Sierhuis (UvA), *Modeling and Simulating Work Practice BRAHMS: a multiagent modeling and simulation language for work practice analysis and design*.
- 2001-9** Pieter Jan 't Hoen (RUL), *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes*.
- 2001-8** Pascal van Eck (VU), *A Compositional Semantic Structure for Multi-Agent Systems Dynamics*.
- 2001-7** Bastiaan Schonhage (VU), *Diva: Architectural Perspectives on Information Visualization*.
- 2001-6** Martijn van Welie (VU), *Task-based User Interface Design*.
- 2001-5** Jacco van Ossenbruggen (VU), *Processing Structured Hypermedia: A Matter of Style*.
- 2001-4** Evgueni Smirnov (UM), *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets*.
- 2001-3** Maarten van Someren (UvA), *Learning as problem solving*.
- 2001-2** Koen Hindriks (UU), *Agent Programming Languages: Programming with Mental Models*.
- 2001-1** Silja Renooij (UU), *Qualitative Approaches to Quantifying Probabilistic Networks*.
- 2000-11** Jonas Karlsson (CWI), *Scalable Distributed Data Structures for Database Management*.
- 2000-10** Niels Nes (CWI), *Image Database Management System Design Considerations, Algorithms and Architecture*.

- 2000-9 Florian Waas (CWI), *Principles of Probabilistic Query Optimization*.
- 2000-8 Veerle Coupe (EUR), *Sensitivity Analysis of Decision-Theoretic Networks*.
- 2000-7 Niels Peek (UU), *Decision-theoretic Planning of Clinical Patient Management*.
- 2000-6 Rogier van Eijk (UU), *Programming Languages for Agent Communication*.
- 2000-5 Ruud van der Pol (UM), *Knowledge-based Query Formulation in Information Retrieval*.
- 2000-4 Geert de Haan (VU), *ETAG, A Formal Model of Competence Knowledge for User Interface Design*.
- 2000-3 Carolien M.T. Metselaar (UVA), *Sociaal-organisatorische gevolgen van kennistechnologie; een procesbenadering en actorperspectief*.
- 2000-2 Koen Holtman (TUE), *Prototyping of CMS Storage Management*.
- 2000-1 Frank Niessink (VU), *Perspectives on Improving Software Maintenance*.
- 1999-8 Jacques H.J. Lenting (UM), *Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism for Discrete Reallocation*.
- 1999-7 David Spelt (UT), *Verification support for object database design*.
- 1999-6 Niek J.E. Wijngaards (VU), *Re-design of compositional systems*.
- 1999-5 Aldo de Moor (KUB), *Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems*.
- 1999-4 Jacques Penders (UM), *The practical Art of Moving Physical Objects*.
- 1999-3 Don Beal (UM), *The Nature of Minimax Search*.
- 1999-2 Rob Potharst (EUR), *Classification using decision trees and neural nets*.
- 1999-1 Mark Sloof (VU), *Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products*.
- 1998-5 E.W.Oskamp (RUL), *Computerondersteuning bij Straftoemeting*.
- 1998-4 Dennis Breuker (UM), *Memory versus Search in Games*.
- 1998-3 Ans Steuten (TUD), *A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective*.
- 1998-2 Floris Wiesman (UM), *Information Retrieval by Graphically Browsing Meta-Information*.
- 1998-1 Johan van den Akker (CWI), *DEGAS - An Active, Temporal Database of Autonomous Objects*.