COMBINATORIAL MARKOV RANDOM FIELDS AND THEIR APPLICATIONS TO INFORMATION ORGANIZATION

A Dissertation Presented

by

RON BEKKERMAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2008

Computer Science

COMBINATORIAL MARKOV RANDOM FIELDS AND THEIR APPLICATIONS TO INFORMATION ORGANIZATION

A Dissertation Presented

by

RON BEKKERMAN

Approved as to style and content by:

James Allan, Chair

W. Bruce Croft, Member

Erik Learned-Miller, Member

Andrew Cohen, Member

Andrew Barto, Department Chair
Computer Science

To my grandmother, who is my soul,
to my mother, who is my mind,
to my wife, who is my heart,
and to my daughter, who is my life.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval, in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903, in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023, and in part by Google Inc. Any opinions, findings and conclusions or recommendations expressed in this material are the author's and do not necessarily reflect those of the sponsor.

I would like to thank my advisor, Prof. James Allan, for his positive attitude toward my work, for his patience and unfailing optimism in difficult situations, for his brilliant ideas and inspiring vision of the research field. I am grateful to Prof. W. Bruce Croft for being an honest, fair opponent in research topics and a valuable aid in non-research issues. I thank Prof. Erik Learned-Miller for many hours of fruitful discussions and collaborative work. I thank Prof. Andrew Cohen for useful feedback and viewing my work from a different angle. I thank Prof. Andrew McCallum for introducing me to the area of graphical models. I thank Prof. Mehran Sahami for providing intellectual and financial support to my research. I am deeply thankful to Prof. Shlomo Zilberstein for his generous help prior to and throughout my PhD studies. I thank Prof. Sridhar Mahadevan and Prof. Micah Adler for being great teachers. I am especially grateful to Prof. Ran El-Yaniv for introducing me to science, teaching me research principles, guiding me throughout my academic career, being a mentor and a friend.

I would like to thank people who helped me in my thesis work. First, I am thankful to Dr. Victor Lavrenko, who put my preliminary ideas into perspective and made

an important contribution to my work in its initial stages. I thank Prof. Leslie Pack-Kaelbling, Prof. Polina Golland, Prof. Chris Pal, Prof. Rina Dechter, and Dr. Uri Lerner for interesting discussions and notational clarifications. I thank Prof. Yoram Singer, Prof. Nir Friedman, and Dr. Noam Slonim for valuable suggestions for improving my methods. I thank Dr. Melinda Gervasio for providing priceless data. I am very grateful to my co-authors Dr. Koby Crammer, Dr. Hema Raghavan, Dr. Jeon Jiwoon, Prof. Koji Eguchi, Aron Culotta, and Gary Huang for investing their time and effort in our mutual projects. I thank fellow lab members Dr. Fernando Diaz, Ao Feng, David Mimno, Dr. Vanessa Murdock, Mark Smucker, Dr. Trevor Strohman, and Dr. Charles Sutton for their generous technical assistance. Finally, I thank the (current and former) departmental staff Kate Moruzzi, Sharon Mallory, Leeanne Leclerc, Pauline Hollister, and Andre Gauthier for their prompt responses to numerous inquiries from my side.

I would like to thank people who made my family's and my stay in the Pioneer Valley enjoyable and comfortable. First, I thank Dr. Vanessa Murdock and her family for being our guides to the Valley, for making us feel at home far from our home. I thank my colleagues Dr. Hema Raghavan, Dr. Jeon Jiwoon, Dr. Ramesh Nallapati, Mark Smucker, and Ben Wellner for their precious friendship. I thank Lena Bloch for bringing music to our life. I am sincerely grateful to our Israeli friends in the Valley: Prof. Hava Siegelmann, the Katz family, the Shenhar family, the Ofir family, the Avishay family, Susan Moser, Yariv Levy, Nati Lenchner, Yariv Hofstein, Dan Mason, and especially Inbar Bluzer for their moral support and warmth.

I would like to thank our personal friends for always being around, even if they live thousands of miles away from us. Their incomplete list includes: the Spirt family, the Averbouch family, the Akselrod family, the Lederberg family, the Zarzhevsky family, the Zacharias famaly, the Lezhak family, the Etinberg family, the Tsitrin family, the Rubinov family, the Gabrilovich family, the Malik family, Yael Weisberger, Tali Stern,

Yuval Scharf, Eyal Gordon, Evgeny Panman, Prof. Wendy Wang, Dmitri Shtilman, and, of course, Dasha Olshanetskaya.

Finally, I would like to thank my family for their unconditional love and tremendous support. I thank all the Bekkermans in Israel, Russia, and Canada for being my *real* family. I thank my parents-in-law, Tatyana and Alexander Nikitin, for accepting me as their own son. I thank my father, Vladimir, for always thinking about me. I am deeply grateful to my daughter, Naomi, for bringing light to my life. I do not find appropriate words to express my gratitude to my mother, Faina, who devoted her entire life to me. *Every* word in the rest of this thesis is a word of appreciation to my wife, Anna, my other self.

ABSTRACT

COMBINATORIAL MARKOV RANDOM FIELDS AND THEIR APPLICATIONS TO INFORMATION ORGANIZATION

FEBRUARY 2008

RON BEKKERMAN

B.Sc., TECHNION—ISRAEL INSTITUTE OF TECHNOLOGY

M.Sc., TECHNION—ISRAEL INSTITUTE OF TECHNOLOGY

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

We propose a new type of undirected graphical models called a Combinatorial Markov Random Field (Comraf) and discuss its advantages over existing graphical models. We develop an efficient inference methodology for Comrafs based on combinatorial optimization of information-theoretic objective functions; both global and local optimization schema are discussed. We apply Comrafs to multi-modal clustering tasks: standard (unsupervised) clustering, semi-supervised clustering, interactive clustering, and one-class clustering. For the one-class clustering task, we analytically show that the proposed optimization method is optimal under certain simplifying assumptions. We empirically demonstrate the power of Comraf models by comparing them to other state-of-the-art machine learning techniques, both in text clustering and image clustering domains. For unsupervised clustering, we show that Comrafs consistently and significantly outperform three previous state-of-the-art clustering

techniques on six real-world textual datasets. For semi-supervised clustering, we show that the Comraf model is superior to a well-known constrained optimization method. For interactive clustering, Comraf obtains higher accuracy than a Support Vector Machine, trained on a large amount of labeled data. For one-class clustering, Comrafs demonstrate superior performance over two previously proposed methods. We summarize our thesis by giving a comprehensive recipe for machine learning modeling with Comrafs.

TABLE OF CONTENTS

Page

ACKNOWLEDGMENTS . . . . . **v**

ABSTRACT . . . . . **viii**

LIST OF TABLES . . . . . **xiv**

LIST OF FIGURES . . . . . **xvi**

CHAPTER

1. INTRODUCTION **1**

2. PRELIMINARIES **7**

2.1 Markov Random Fields 7
2.2 Three major learning paradigms 9
2.3 Clustering 11

3. COMBINATORIAL MARKOV RANDOM FIELDS **12**

3.1 Combinatorial random variables 12
3.2 Combinatorial Markov Random Fields 13
3.3 Algorithmic aspects of inference in Comrafs 15
3.4 Summary 18

4. COMRAFS FOR MULTI-MODAL CLUSTERING **20**

4.1 Choosing an objective function 21
4.2 Clustering as combinatorial optimization 24
4.3 Multi-way Distributional Clustering (MDC) 25

4.3.1 Computational complexity of MDC 30

4.4 Clique-wise MDC 31
4.5 Related work 32

4.6 Experimentation: email clustering . . . 33
4.6.1 Evaluation measure . . . 34
4.6.2 Datasets . . . 35
4.6.2.1 20 Newsgroups . . . 36
4.6.2.2 Enron Email Dataset . . . 37
4.6.2.3 CALO Email Dataset . . . 37
4.6.3 Baseline algorithms . . . 38
4.6.4 Implementation details . . . 39
4.6.5 Comparative results . . . 40
4.6.5.1 Experimentation with clustering schedule . . . 43
4.6.5.2 Experimentation with the length of local search . . . 44
4.6.6 Model analysis . . . 46
4.6.7 Multi-modal clustering for social network analysis . . . 48
4.7 Experimentation: Web appearance disambiguation . . . 48
4.7.1 Related work . . . 51
4.7.2 Evaluation criterion . . . 51
4.7.3 WAD dataset . . . 52
4.7.4 Baseline: link structure model . . . 54
4.7.5 Comparative results . . . 56
4.8 Experimentation: clustering scientific papers . . . 59
4.9 Experimentation: clustering documents by genre . . . 61
4.9.1 Dataset . . . 64
4.9.2 Comparative results . . . 64
4.10 Summary . . . 68
5. COMRAFS FOR SEMI-SUPERVISED LEARNING . . . 70
5.1 Semi-supervised clustering with Comrafs . . . 70
5.1.1 Experimentation . . . 72
5.2 Transfer learning with Comrafs . . . 73
5.3 Interactive clustering with Comrafs . . . 75
5.3.1 Related work . . . 76
5.3.2 Interactive clustering scenario . . . 77
5.3.3 Clustering by sentiment . . . 79

5.3.4 Dataset . . . 80
5.3.5 Experimental setup . . . 81
5.3.6 Comparative results . . . 83

5.4 Summary . . . 84

6. COMRAFS FOR ONE-CLASS CLUSTERING . . . 86

6.1 Related work . . . 89
6.2 One-class clustering of words . . . 90
6.3 Min-Entropy algorithm for one-class clustering in text . . . 93

6.3.1 Relaxation of the uniformity assumption . . . 95

6.4 One-class co-clustering (OCCC) . . . 96

6.4.1 Heuristic for choosing the size of word cluster . . . 98

6.5 The Latent Topic/Background (LTB) model . . . 98
6.6 Experimentation with OCCC and LTB . . . 100

6.6.1 Web appearance disambiguation . . . 101
6.6.2 Re-ranking Web retrieval results . . . 102
6.6.3 Detecting the topic of the week . . . 105

6.7 Summary . . . 108

7. IMAGE CLUSTERING WITH COMRAFS . . . 110

7.1 Related work . . . 111
7.2 Multi-modal clustering objective, revisited . . . 112
7.3 Comraf*: a lightweight version of the Comraf model . . . 113

7.3.1 Inference in Comraf* . . . 115

7.4 Modalities of an image collection . . . 116

7.4.1 Rectangular blobs . . . 116
7.4.2 Blobs constructed by Comraf models . . . 117

7.5 Experimentation . . . 119

7.5.1 Datasets . . . 120
7.5.2 Comparative results . . . 121

7.6 Summary . . . 123

8. CONCLUSION **126**

BIBLIOGRAPHY **131**

APPENDICES

A. PROOF OF THEOREM 6.2.1 . . . . . **140**
B. DETAILS OF EM ALGORITHM FOR ONE-CLASS CLUSTERING **142**

LIST OF TABLES

Table **Page**

4.1 **Statistics of email datasets.** Number of distinct words and number of correspondents are after preprocessing. 36

4.2 **Micro-averaged accuracy** (± standard error of the mean, when applicable) on the six datasets. The SVM *supervised* classification accuracies are obtained with 4-fold cross validation. "OOM" means "out of memory": WEKA was unable to cluster 20NG, on a 4GB RAM machine. Bold numbers are the best results over all. 41

4.3 **Macro-averaged accuracy** (± standard error of the mean) on CALO and Enron datasets. Each number is an average over ten independent runs. Bold numbers are the best results over all. 42

4.4 **Micro-averaged accuracy** (± standard error of the mean) on CALO and Enron datasets. Each number is an average over ten independent runs. Comrafs models are 2-modal, 3-modal and 4-modal, with the *sequential* optimization applied at each node. Bold numbers are the best results over all. 43

4.5 **Statistics of the WAD dataset.** Categories are different namesakes or *other* in case if the page does not refer to any of the namesakes. The last column shows the number of pages that actually mention the person of our interest. 53

4.6 **Web appearance disambiguation results.** Bi-modal Comraf results are averaged over 4 independent runs, with the standard error of the mean reported after the ± sign. 57

4.7 **Clustering scientific papers.** Comraf models for clustering: (a) documents and title words; (b) documents and citations; (c) documents, title words and citations in a tree-structured model; (d) documents, title words and citations in a loopy model; (e) documents and abstract words. The bottom line is the micro-averaged clustering accuracy obtained by those models. 60

4.8 **Clustering by genre.** Micro-averaged clustering accuracy on the BNC corpus, averaged over four independent runs. Standard error of the mean is shown after the ± sign. Comraf results with other POS tuples, besides bigrams, are in Figure 4.9(left). The BOW+POS hybrid setup is only applicable in Comrafs. 64

4.9 **Performance of various methods per genre.** For each genre we show a list of sizes (in number of documents) of this genre's representation in various clusters. We sort this list by the size of the representation from the largest to the smallest. An asterisk after the number of documents means that this genre is dominant in the corresponding cluster. 66

5.1 **Clustering by sentiment.** Clustering accuracy of Comraf models (both interactive and non-interactive) is compared with clustering accuracy of k-means and LDA, as well as with classification accuracy of SVM. All results are averaged over four independent runs. Standard error of the mean is shown after the ± sign. 82

6.1 **Most highly ranked words** by OCCC and LTB, on the WAD dataset. 102

6.2 **Re-ranking Web retrieval results**: We compare one-class clustering accuracy of our OCCC (with heuristic from Section 6.4.1) and LTB (initialized with $\pi_i = 0.5$) models with the accuracy of the original Google rank lists, of one-class SVM (OC-SVM) and of one-class Information Bottleneck (OC-IB) [28] with l^2-norm. 103

6.3 **One-class clustering accuracy on the "topic of the week" detection task.** The accuracies are macro-averaged over the 26 weekly data chunks. Standard error of the mean is presented after the ± sign. 108

7.1 **Categories (and their sizes) of the IsraelImages dataset.** 120

7.2 **Micro-averaged clustering accuracy on IsraelImages.** All IB/Comraf results are averaged over 10 independent runs with the standard error of the mean reported after the '±' sign. 121

7.3 **Micro-averaged clustering accuracy on Corel.** All IB/Comraf results are averaged over 10 independent runs with the standard error of the mean reported after the '±' sign. 122

LIST OF FIGURES

Figure **Page**

2.1 An example of a Markov Random Field. 9

4.1 A Comraf graphs for: (a) hard version of Information Bottleneck; (b) information-theoretic co-clustering; (c) one of the possible 4-modal Comrafs. 23

4.2 A schematic view of bi-modal MDC with a simple, non-weighted round-robin schedule. At each iteration black clusters are split and then white clusters are merged. 28

4.3 Comraf graphs for 2-modal, 3-modal and 4-modal Comrafs used in our experiments. We consider interactions between combinatorial random variables that correspond to documents D^c, words W^c, email correspondents C^c and email *Subject* lines S^c. Note that we use only tree-structured models, as they are simpler than loopy models and on the email foldering task they show comparable results to those obtained with loopy models (see Section 4.6.6 for a discussion). In Section 4.8 we present a result when a loopy model is significantly superior to a tree-structured one. 35

4.4 Clustering accuracies as a function of the length of local search in sequential MDC: '0.5' on the x-axis means that the MDC's optimization routine was executed over one half of the data points (chosen uniformly at random), while '3' means that the optimization routine was executed over every data point 3 times. All our results are averaged over 10 independent runs. 44

4.5 Experimenting with various Comraf graphs on MGERVASIO. 46

4.6 Relevant and irrelevant Web pages according to the Link Structure model. Relevant pages are within the δ-radius from the *Core Connected Component*. White, gray and black colors indicate that the pages are retrieved by three different queries. 55

4.7 Precision/recall curve of the MDC algorithm. Points correspond to consequent iterations of the algorithm (merges of Web page clusters). 58

4.8 Comraf graphs for: (a) 1-way document clustering with POS unigrams as an *observed* r.v. (shaded node); (b) 2-way clustering of documents and POS bigrams (same as for POS 3-grams or 4-grams); (c) 2-way clustering with BOW; (d) 3-way clustering with POS bigrams and BOW. 63

4.9 Clustering by genre. Micro-averaged clustering accuracy of Comraf models as a function of: (left) size of POS n-gram (1-grams, 2-grams, 3-grams and 4-grams); (right) threshold on low frequency words—a point i on the X axis means that in this experiment words that appear in less than i documents are removed. 68

5.1 Comraf graphs for: (left) semi-supervised clustering; (right) clustering with transfer learning. 72

5.2 Plots (a)-(e): comparing accuracies of the semi-supervised Comraf and the constrained optimization method on five email datasets. Plot (f): the semi-supervised Comraf's resistance to noise in labeled data. 74

5.3 Interactive clustering by sentiment. Micro-averaged clustering accuracy over various users: (left) over interactive learning iterations (with original seed words only, after one correction step and after two correction steps). The horizontal line is SVM performance (after feature extraction using a given list of sentimental words, and after training on over 20K documents); (right) over categories of the dataset after two correction steps. 83

6.1 (left) The simplest generative model; (right) Latent Topic/Background model (Section 6.5). 91

6.2 An illustration of possible distributions of word counts in one-class clustering: (left) uniform case; (right) multinomial case. Words whose counts are above the threshold are considered relevant. Note that in the multinomial case counts of some relevant words can be lower than counts of non-relevant words. 92

6.3 The accuracy (as defined in Section 6.6, averaged over 100 independent runs) of identifying $\mathcal{R}$ in a simulation of the generative process, over various values of the constant from Equation (6.1) for the sampling size N. In Equation (6.1), the value of this constant is set to 16. Here we show that the value of 2 is enough in practice. 93

6.4 Web appearance disambiguation. (left) OCCC accuracy as a function of the word cluster size; (right) accuracy of LTB (with the underlying EM algorithm) over various initializations of π_i parameters: LTB shows a more robust behavior than OCCC, however LTB's maximal result (80.2%) is slightly inferior to the OCCC's (82.4%). 101

6.5 One-class clustering results on the "topic of the week" detection task. 107

7.1 Comraf* models: (a) for images G^c and words W^c from their captions; (b) for images, words and colors C^c; (c) for images, words, colors and blobs B^c; (d) straightforward generalization to any number of modalities. 114

7.2 (left) A Comraf model for simultaneously clustering images G^c and their rectangular regions R^c, while taking into account words W^c from image captions, colors C^c and texture data T^c; (right) a translation of this model into a two-step Comraf*: the first Comraf* is for clustering regions into blobs, whereas the second Comraf* is for clustering images based on these blobs. 118

7.3 Experimentation with various numbers of: (left) colors on IsraelImages in a 3-node images/words/colors Comraf*; (center) colors for clustering regions in the 2-step Comraf* on IsraelImages; (right) blobs on Corel in a 3-node images/words/blobs Comraf*. Our baseline is the 2-node images/words clustering result. Left and right graphs show the same trend: after reaching a certain number of colors (256) or blobs (2000), the results vary only insignificantly. The central graph, however, shows that too many colors for clustering regions can hurt. 123

7.4 Corel dataset. The first row shows clustering results using only words. Swimmers and swimming tigers are clustered together because they share common terms like "water" and "swim". The second and the third rows show clustering results using both words and blobs. The swimmers and the swimming tigers are now in two different clusters with other similar images. 125

7.5 IsraelImages dataset. People portraits and pictures of the menorah monument are clustered together using caption words because they have a word 'Knesset' (the Israeli parliament) in common: the individuals are Knesset members, while the menorah monument is placed in front of the Knesset building. The problem is resolved after the color modality is added. 125

CHAPTER 1

INTRODUCTION

Graphical models have proven themselves to be a useful tool in machine learning, showing excellent results in information retrieval [81], natural language processing [93], computer vision [45], and a variety of other fields [57]. A striking benefit of using graphical models is the availability of black-box inference algorithms; once a model is designed, it is usually straightforward to apply an existing optimization procedure to make inferences in the model. Nonetheless, existing graphical models have certain limitations, both within supervised and unsupervised frameworks (that is, when training data is available or unavailable, respectively).

Supervised learning problems are usually solved using either *generative* graphical models (i.e. Bayesian networks [87]) or *discriminative* graphical models (such as conditional random fields [66]). While the goal of inference in generative models is to estimate model parameters represented jointly with the data, the goal of inference in discriminative graphical models is to estimate model parameters *given* the data, in a conditional manner. The major problem of the discriminative approach is that in order to construct a useful model, a large amount of labeled data is required. If the amount of available labeled data is not enough for training a model, it often *overfits*: i.e. it performs well on data similar to the training data, but shows significantly worse results on "unexpected" data instances. Unfortunately, it is usually impossible to decide whether the amount of available training data is enough for constructing an effective model. Also, a supervised model can perform poorly if trained on low-quality, noisy data.

Unsupervised learning tasks are often performed using generative graphical models (discriminative models are inapplicable to these tasks). The structure of a generative model describes a hypothetical procedure according to which the data was presumably generated. To design a generative model, practitioners traditionally make assumptions about the model's structure, based on domain knowledge, the need for computational tractability, or both. Such assumptions may be inappropriate and thus introduce undesired bias into the model. Another potentially problematic issue is that modern generative models consist of thousands or even millions of nodes—such models are difficult to fit, analyze and learn from data (model learning can easily become infeasible if no significant restrictions on the class of models are made).

Since both generative and discriminative graphical models have significant drawbacks, other types of graphical models are emerging, which now becomes an active topic in machine learning. Recently, McCallum et al. [79] proposed a model that combines generative and discriminative training. LeCun and Huang [69] proposed *energy-based models* which allow optimization of non-normalized objective functions factorized over a graphical model. However, both models are proposed only within the framework of *supervised* learning.

In this thesis, we develop a new type of graphical model that has the following characteristics:

- **Unsupervised or semi-unsupervised flavor.** The model is not overly dependent on the quantity and quality of training data, but rather is applicable to the cases when no or little labeled data is available. Even if the amount of labeled data is sufficiently large, the model does not assume the data's purity, but takes advantage of this data by maximizing agreement of unlabeled and labeled data in a semi-unsupervised setup.

- **Minimal bias; minimal prior knowledge to be incorporated.** The number of assumptions made on the model structure is as small as possible. In

particular, no generative assumptions are made, which minimizes the risk of making assumptions that are misleading or unnecessary.

- **Compactness, ability for model learning and comprehensive analysis of the model behavior.** Graphical models with millions of nodes are difficult to comprehend and analyze. We take into account the fact that learning the model structure can be optimal only for small models [27].

To meet the criteria above, we construct a graphical model which is intrinsically different from existing graphical models. The most important difference is that in our model, a certain portion of the model complexity is transferred from its graph topology into its nodes, such that a resulting model consists of a small number of "rich" nodes. It turns out that such a model is straightforwardly applicable to *multi-modal learning* problems.

Multi-modal learning is a learning framework in the environment where multiple views (or *modalities*) of the data are available. For example, in the text domain, a set of documents is one modality of the data, while a set of words within those documents is another modality. In fact, most real-world datasets are multi-modal. Multi-modality of the data can be observed in a variety of research fields, such as:

- **Text processing:** documents, words, authors, titles, part-of-speech tags;
- **Image processing:** images, colors, texture, blobs, interest points, caption words;
- **Video processing:** video signal, audio signal, frames, subtitles, transcripts;
- **Bioinformatics:** patients, tissues, samples, genes, proteins, compounds;
- **Web information retrieval:** Web pages, words, hyperlinks, markup primitives;

- **Data mining:** movies, actors, directors, production companies;

and many others.

Three decades ago McGurk and MacDonald published their pioneering work [80] that revealed the multi-modal nature of speech perception: sound and moving lips compose one system, so to better process audio signals, an audio/video interaction should be modeled. Since then, machine learning researchers have widely exploited data multi-modality, using a variety of approaches, such as multi-modal neural networks [32], multivariate information bottleneck [46], and multi-view expectation maximization [21].

We propose a graphical model for multi-modal learning, only *one* node of which is assigned for each modality, while edges represent statistical interactions between the modalities. Since such interactions are symmetric, the resulting model is *undirected*, i.e. they adopt the *Markov Random Field (MRF)* formalism. All the applications that we consider in this thesis will be of the multi-modal nature, however, in our future work, we will explore other types of possible applications.

The model we propose has the desired characteristics listed above:

- Multi-modality discloses the high-level *structure* of data, being therefore a cheap and easily available form of supervision. Indeed, while obtaining labeled examples is expensive, deciding which data views are relevant to a particular task in hand is usually straightforward. Taking advantage of this additional, *structural* knowledge allows us to successfully solve unsupervised and semi-supervised problems.

- The *only* domain knowledge incorporated into the model is availability / usability of multiple modalities and their interaction patterns. No assumptions about prior distributions, latent variables and data point-wise interactions are made which minimizes the model's bias.

- Meaningful models can consist of just a handful of nodes, allowing easy analysis. For example, the problem of choosing the most influential interactions between nodes can be straightforwardly solved by testing a number of potentially good combinations (or even all possible combinations, in case of models with only few nodes).

In this thesis we explore a range of multi-modal clustering tasks, including *one-class* clustering. We consider only discrete tasks over finite datasets, and we note that those tasks have a *combinatorial* nature: given a dataset of n instances, the standard (hard) clustering is the problem of partitioning these instances into k groups, whereas one-class clustering is the problem of selecting k instances—both are well-known combinatorial problems. Therefore, we represent these learning tasks as *combinatorial optimization*. In multi-modal cases, our goal is to simultaneously solve multiple combinatorial optimization problems, one for each data modality.

To summarize, the contributions of this thesis are as follows:

1. We propose a new type of graphical model called a *Combinatorial Markov Random Field (Comraf)* that has beneficial properties (as discussed above): it models a high-level structure of the data, represented as a handful of "rich" nodes (that correspond to data modalities) and interactions between them. The inner structure of Comraf's nodes is apparent and therefore does not require an explicit graphical representation in the model, which results in a light and elegant layout.

2. We show that Comrafs are a natural modeling framework for multi-modal problems, able to obtain excellent results on real-world tasks. For each task, a particular objective function is designed that best fits the task. Therefore, Comrafs are more flexible than most graphical models which are mainly limited to using maximum likelihood (ML) or maximum a posteriori (MAP) objectives.

3. We apply Comrafs to unsupervised, semi-supervised, interactive and one-class clustering tasks. We represent each task as a combinatorial optimization problem of the multi-modal nature. To our knowledge, most of the proposed tasks are novel: we are not aware of previous work on semi-supervised multi-modal clustering, interactive multi-modal clustering, or one-class multi-modal clustering.

4. We design *information-theoretic* objective functions for our models. In the case of one-class clustering, we show that optimizing our objective function leads to an optimal solution, under some simplifying assumptions. Also, in the case of multi-modal clustering, we show that incorporating our objective function into the Comraf model nicely generalizes previous successful clustering models.

5. We propose combinatorial optimization methods for solving our learning problems, for each of which we design efficient combinatorial algorithms and analyze their computational complexity

6. Overall, we present a formal framework for multi-modal learning that brings together two research areas: graphical models and combinatorial optimization.

The rest of this thesis is organized as follows: in Chapter 2 we provide some necessary background; in Chapter 3 we describe the Comraf model; after which we discuss each Comraf application in turn: (unsupervised) clustering in Chapter 4, semi-supervised and interactive clustering in Chapter 5, and one-class clustering in Chapter 6. In Chapter 7, we summarize previous chapters by exploring a variety of Comraf modeling possibilities on an example of image clustering. In Chapter 8, we conclude and discuss advanced Comraf problems that are not described in depth in this thesis, such as *multi-modal ranking*.

CHAPTER 2

PRELIMINARIES

In this chapter, we first provide background information on graphical models, and in particular on Markov Random Fields. We then present three major machine learning paradigms: supervised, semi-supervised, and unsupervised learning. Finally, we concentrate on *data clustering*—the most important application of unsupervised learning—for which we give some necessary definitions and insights.

2.1 Markov Random Fields

A *graphical model* is a tuple (G, P), where G is a graph whose nodes correspond to random variables $\mathbf{X} = \{X_1, \ldots, X_m\}$ and whose edges $\mathbf{E}$ denote interactions between these variables; P is a joint probability distribution defined over $\mathbf{X}$. Let us use a short notation $P(\mathbf{x}) = P(X_1 = x_1, \ldots, X_n = x_n)$, where each x_i is a value from X_i's domain.

Definition 2.1.1 *A graphical model* (G, P) *is called a* Markov Random Field (MRF) *if the following two conditions hold:*

- *(Positivity)* $\forall x : \ P(\mathbf{x}) > 0$
- *(Markovianity) Let* $\mathbf{X}_1$, $\mathbf{X}_2$ *and* $\mathbf{X}_3$ *be three disjoint subsets of random variables in* $\mathbf{X}$. *We have that*

$$P(\mathbf{X}_1, \mathbf{X}_2 | \mathbf{X}_3) = P(\mathbf{X}_1 | \mathbf{X}_3) P(\mathbf{X}_2 | \mathbf{X}_3)$$

(i.e. $\mathbf{X}_1$ and $\mathbf{X}_2$ are conditionally independent given $\mathbf{X}_3$) iff every path between a node from $\mathbf{X}_1$ and a node from $\mathbf{X}_2$ contains a node from $\mathbf{X}_3$.

The *Markov blanket* of a node X_i is defined as the set of all the immediate neighbors of X_i in G. The Markovianity can then be restated as having each variable X_i conditionally independent of the rest of the model, given its Markov blanket. Note symmetric dependencies between nodes in an MRF—those dependencies are represented in G by *undirected* edges. Consequently, an MRF is often referred to as an *undirected graphical model*.

An important observation of an MRF is that the joint distribution P is given but (in most cases) not fully observed. The goal of an *inference* procedure in an MRF is then to answer questions about the distribution P, such as what is the most likely assignment $\mathbf{x}^* = \{x_1^*, \ldots, x_m^*\}$ to variables $\{X_1, \ldots, X_m\}$ (this task is called the *Most Probable Explanation—MPE*, see, e.g., [75]). Naturally, answering most such questions is NP-hard since it potentially requires considering every possible assignment. Thus, most inference techniques fall into the category of approximation methods.

Definition 2.1.2 *A distribution P is called a* Gibbs distribution *if it can be written in the form*

$$P(x) = \frac{1}{Z} \exp\left(\sum_{c} f_{\mathcal{C}}(x_{\mathcal{C}})\right),$$

where

- *$\mathcal{C}$ is a clique in the graph G;*
- *$f_{\mathcal{C}}$ is a real-valued function defined over values of random variables from $\mathcal{C}$;*
- *Z is a normalization factor.*

We refer to functions $f_{\mathcal{C}}$ as *log-potential functions* (this term reflects the fact that their exponents are traditionally referred to as *potential functions*). The normalization factor Z is called a *partition function*.

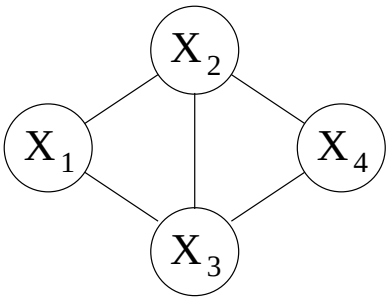


Figure 2.1. An example of a Markov Random Field.

First proven by Julian Besag [19], the Hammersley-Clifford theorem states that

Theorem 2.1.3 *The tuple* $(\mathcal{X}, G)$ *is an MRF if and only if* P *is a Gibbs distribution.*

Note that log-potential functions can be defined on cliques of any size, however, smaller cliques are usually preferred from the computational point of view. For example, consider an MRF from Figure 2.1 where X_1, X_2, X_3, X_4 are multinomial random variables, each with 10,000 possible values. We can consider two cliques of size 3, i.e. $\mathbf{X}_1 = \{X_1, X_2, X_3\}$ and $\mathbf{X}_2 = \{X_2, X_3, X_4\}$ and then the joint distribution $P(\mathbf{x})$ can be factorized over those cliques as:

$$P(\mathbf{x}) = \frac{1}{Z_{\mathbf{f}}} \exp \sum_i f_i(\mathbf{x}_1) \exp \sum_i f_i(\mathbf{x}_2),$$

such that each log-potential function f_i will have to have 10^{12} values. Inference in a model like that can be infeasible in practice. We can also consider five cliques of size 2 (i.e. the edges $\{X_1, X_2\}$, $\{X_1, X_3\}$, $\{X_2, X_3\}$, $\{X_2, X_4\}$, and $\{X_3, X_4\}$), and factorize the joint distribution accordingly. In this case, the log-potentials f_i will have to have only 10^8 values which is, in many cases, feasible.

2.2 Three major learning paradigms

In this thesis, we employ MRFs for solving *unsupervised* and *semi-supervised* learning problems. When possible, we compare our results with the ones of *supervised*

learning methods, such as a Support Vector Machine (SVM) [109]. The most widely studied type of supervised learning problems is *classification*: a model is trained on (a large number of) data instances, each of which was a priori associated with one or more target classes (we say that it was *labeled*). The model is then applied to associate other, unlabeled data instances with the target classes. Obviously, it is burdensome to collect and label the training data.

In contrast, in unsupervised learning, the model is built to fit the data as it is, where no labeled instances are necessary. Data *clustering* is the main example of unsupervised learning problems. There are two versions of data clustering: *hard* and *soft*. In hard clustering, we partition the set of data instances into groups (*clusters*) such that these groups are as homogeneous as possible (according to a given criterion). Soft clustering is applied when data instances can belong to more than one cluster: each data instance is associated with all the clusters according to a certain probability distribution. In this thesis, we will consider only the hard clustering task, leaving the soft clustering case for future work.

Unsupervised learning problems are usually solved in graphical models using the *maximum likelihood (ML)* framework (see, e.g., [22]), where model parameters that best explain the data are estimated. Most ML methods deal with approximating $Z_{\mathbf{f}}$, which is generally a hard task, because $Z_{\mathbf{f}}$ depends on the particular choice of f_i's and is a sum over all the possible configurations. In this thesis, we will apply the MPE framework instead, for the reasons that will be clear later.

Semi-supervised learning is usually viewed from two difference perspectives: (a) as training a supervised model while taking advantage of available unlabeled instances; (b) as building an unsupervised model that takes advantage of some labeled data, whose amount is not enough to train a supervised model. In this thesis, we focus on the latter type, which is often called *semi-supervised clustering* [116].

2.3 Clustering

Most existing data clustering algorithms belong to one of two categories: *hierarchical* (top-down or bottom-up) and *flat*. A flat algorithm starts with data instances distributed over k clusters (where k is the desired number of clusters) and reorganizes / updates the clusters until convergence. A top-down hierarchical algorithm is initialized with one cluster containing all data instances, which is then iteratively split into portions until the desired number of clusters k is achieved. A bottom-up hierarchical algorithm starts with singleton clusters (one data instance per cluster) and merges clusters iteratively until, again, k is reached. An obvious drawback of flat algorithms as compared to hierarchical ones is in the fact that flat procedures are often heavily dependent on their initialization: most of them perform poorly when initialized at random. Many heuristics have been proposed that come up with meaningful initial clusters (see, e.g., [40]), however, most of these heuristics are domain specific. Therefore, in this thesis we concentrate on hierarchical clustering schema, although we occasionally mention flat methods as well (see e.g. Section 4.4).

CHAPTER 3

COMBINATORIAL MARKOV RANDOM FIELDS

In this chapter, we first introduce the notion of a combinatorial random variable, then propose Combinatorial Markov Random Fields (Comrafs), and develop an inference technique for Comrafs based on combinatorial optimization.

3.1 Combinatorial random variables

Definition 3.1.1 *A* combinatorial random variable *(or* combinatorial r.v.*)* X^c *is a discrete random variable defined over a combinatorial set.*

A *combinatorial set* in mathematical parlance means a set of all subsets, partitionings, permutations etc. of a given finite set. To capture this intuition, we define a finite set A as *combinatorial* if its size is exponential with respect to another finite set B, i.e. $\log |A| = O(|B|)$. As an example, a combinatorial r.v. X^c can be defined over all the outcomes of *lotto 6 of 49*, in which 6 balls are selected from 49 enumerated balls to produce an outcome of the lottery. In this case, set B consists of 49 balls, while set A consists of $\binom{49}{6}$ possible choices of 6 balls from B. In a *fair* lottery, the distribution of X^c is uniform: each outcome is drawn with probability $1/\binom{49}{6}$. However, in an unfair lottery, some outcomes are more probable than others.

It is easy to come up with other examples of combinatorial r.v.'s: over all the possible translations of a sentence, over all the possible orderings in a ranked list of retrieved documents, etc. In Chapter 4 we consider combinatorial r.v.'s over all *partitionings* of a given set; in Chapter 6 we consider combinatorial r.v.'s over all *subsets* of a given set.

From the theoretical perspective, a combinatorial r.v. behaves exactly as an ordinary discrete random variable with a finite domain. However, from the practical point of view, a combinatorial r.v. is different: in most real-world cases, the event space of X^c is so large that the distribution $P(X^c)$ cannot be explicitly specified. Moreover, the Most Probable Explanation (MPE) task (see Chapter 2) for combinatorial r.v.'s can be computationally hard. Considering an unfair lottery example, in which the distribution of X^c is flat (close to uniform), say, the probability of value $\{7, 23, 29, 35, 48, 49\}$ is 0 and the probability of value $\{4, 18, 28, 37, 39, 43\}$ is $2/\binom{49}{6}$, while the rest of the values still have the probability $1/\binom{49}{6}$. An exponentially long sampling process is required to detect the most probable value.

3.2 Combinatorial Markov Random Fields

Definition 3.2.1 *A* combinatorial Markov random field (Comraf) *is a Markov Random Field, at least one node of which is a combinatorial random variable.*

In this thesis, we will consider only Comraf models, *every* node of which is a combinatorial r.v. As in any other MRF, random variables in Comraf models can be in either a *hidden* or *observed* state. A combinatorial r.v. is *hidden* if it can take any value from its event space. A combinatorial r.v. is *observed* if its value is preset and fixed. Chapter 4 presents Comraf models with only hidden variables. In Chapters 5 and 7 we introduce observed random variables to Comraf models.

An edge $e_{ij} = (X_i^c, X_j^c)$ in a Comraf graph corresponds to a statistical interaction between combinatorial r.v.'s X_i^c and X_j^c. A presence or an absence of edge e_{ij} articulates whether X_i^c and X_j^c stay in a tight statistical interaction or not. For example, consider three nodes in a Comraf graph for an email collection, one of which (M^c) corresponds to the modality of email messages, another (A^c) to the authors of the messages, and the third one (S^c) to the subject lines. Obviously, email messages stay in statistical interactions both with their authors and their subjects. However, it is

not straightforward whether the authors' modality interacts with the subject lines. Indeed, the subject line in the first message of an email thread was given by its sender, while all the other messages in this thread are often "forced" to use the same subject line, possibly given by another sender. Therefore, it might be natural to have edges (M^c, A^c) and (M^c, S^c) in the Comraf graph, and to drop the (A^c, S^c) edge.

One might argue that, while the (A^c, S^c) interaction is not clearly present, it might still exist in the data. As in any graphical model, there is a tradeoff between the Comraf's adequacy and the computational complexity of its inference procedure. The larger the Comraf model is, the more difficult the inference would be. Thus, it is the practitioner's responsibility to decide which edges will be present in the model and which will be absent. Let us emphasize this again: since Comraf models are usually compact, a model learning procedure can be used to automatically infer the optimal set of model's edges. Keeping in mind the model learning option, we leave it for our future research. Also note that, as in any other MRF, the lack of statistical interaction between variables X_i^c and X_j^c (and therefore the absence of edge e_{ij} in a Comraf graph) implies conditional independence of X_i^c and X_j^c given the rest of the model.

As discussed above in Section 3.1, even simple inference tasks (trivially performed on ordinary random variables) are computationally hard for combinatorial random variables. Since every combinatorial r.v. carry a large portion of a Comraf complexity, even small Comraf models (of just a few nodes) remain non-trivial. Inference in Comrafs is thus viewed from a different perspective than inference in other graphical models. Usually, an inference procedure is composed from traversing the graph G and performing computations at the graph's nodes. In most graphical models, where nodes are ordinary random variables, the computation step is simple, while traversing the (large) graph is a resource-demanding process. In these cases, it is very important to keep track of numerous intermediate computations. Simplicity and homogeneity

of such process play a crucial role in those models. For example, it is impractical to optimize different objective functions in different regions of the graph G. These considerations dramatically restrict practitioners in their choice of an objective function for their models. Most graphical models optimize the maximum likelihood objective (see Chapter 2). However, the situation is different for Comrafs. In Comrafs, computations performed in each node are the most intensive part of the inference process. Traversing the graph however is relatively inexpensive as the number of nodes is small in comparison to other models. Thus, practically unrestricted variety of objective functions can be considered, both probabilistic and non-probabilistic, homogenous and heterogenous in various regions of G.

Let us now show that optimizing an arbitrary objective function over G can be represented in terms of an MPE inference in a Comraf. As discussed in Chapter 2, the joint distribution of random variables in an MRF is factored over the graph G as:

$$P(\mathbf{x}) = \frac{1}{Z_{\mathbf{f}}} \exp \sum_i f_i(\mathbf{x}),$$

where log-potential functions f_i are arbitrary functions defined over cliques in G. If we fix the log-potentials f_i for each clique, the partition function $Z_{\mathbf{f}}$ becomes a constant. Thus, in the MPE inference, we directly optimize a non-normalized linear combination of the log-potential functions:

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} P(\mathbf{x}) = \arg\max_{\mathbf{x}} \exp \sum_i f_i(\mathbf{x}) = \arg\max_{\mathbf{x}} \sum_i f_i(\mathbf{x}), \tag{3.1}$$

which now solely depends on the choice of the log-potentials.

3.3 Algorithmic aspects of inference in Comrafs

As we have mentioned in Section 3.1, in most cases it is infeasible to explicitly specify the distribution $P(X^c)$, i.e. to represent it as a probability table in which

each value is assigned a certain probability mass. In such situations, estimating the joint distribution of all the Comraf nodes becomes even less possible. A somewhat traditional approach to dealing with the problem of combinatorial explosion is to transfer the probabilistic setup to the continuous space. However, it is well known (see, e.g., [86]) that such a transformation may potentially cause significant approximation errors. Another alternative is to apply a local search in the event space L of X^c. Yet another possibility is to apply more sophisticated combinatorial optimization methods, such as Branch and Bound [67].[1] In this thesis, we choose the local search approach. Let us proceed with definitions.

Definition 3.3.1 *A* transaction *is an elementary operation in traversing the event space L of a combinatorial r.v. X^c.*

In the other words, a transaction is a jump operation between neighboring points in the event space L (i.e., closest values of X^c). For each particular learning task, the event space of a combinatorial r.v. will be defined differently, and so will be a transaction. For now, let us assume that we know how to move from one value of X^c to another.

Definition 3.3.2 *A* path *in L is a sequence of transactions. A path is called* advantageous *if it leads to a more likely value of X^c, otherwise it is* disadvantageous.

In a Comraf model with more than one combinatorial r.v., the most straightforward version of an inference algorithm would be a variation of the *Iterative Conditional Modes (ICM)* method [20]. ICM optimizes the objective (3.1) for each node of an MRF iteratively (in a round-robin fashion), given its Markov blanket. A possible drawback of this approach can be evidenced when the linear combination (3.1) is

[1] Branch and Bound has been used for (uni-modal) clustering by Koontz et al. [62], however it is questionably applicable to multi-modal learning due to its high computational complexity.

Input:
G – Comraf graph of nodes $\{X_1^c, \ldots, X_m^c\}$ and edges $\mathbf{E}$
$P(X_i, \ldots, X_m)$ – joint probability distribution of data, factorized over G
l – number of optimization iterations
Output:
Most likely $x_{1,l}^c, \ldots, x_{m,l}^c$

Initialization:
For $i = 1, \ldots, m$ **do**
  Select a point in L_i to be an initial value $x_{i,0}^c$ of X_i^c
Compute the initial joint $P(x_{1,0}^c, \ldots, x_{m,0}^c)$, factorized over G
Main loop:
For $j = 1, \ldots, l$ **do**
  Select variable $X_{i'}^c$ for optimization
  Construct advantageous path $\left(x_{i',j-1}^c \rightarrow x_{i',j}^c\right)$ in $L_{i'}$
  For all $i \neq i'$ **do** $x_{i,j}^c = x_{i,j-1}^c$

Algorithm 1: A template of an MPE procedure in Comrafs.

taken over log-potential functions f_i, which are intrinsically different in their magnitude and/or semantics (such that explicitly taking their linear combination might not be beneficial). For these situations, we propose another version of an inference algorithm, called *clique-wise optimization (CWO)*, which is a variation of a local optimization method in an MRF. Similarly to ICM, the CWO algorithm iterates over nodes in the MRF. For each node, a clique that contains this node is chosen and the objective (3.1) is optimized with respect to the chosen clique *only*, i.e. independently of the rest of the model. Sutton and McCallum [102] apply a similar method (called *piecewise training*) in a supervised setting. Bouvrie [23] proposes to approach the multi-modal clustering problem by iteratively applying a bi-modal clustering algorithm. To some extend, Bouvrie's method can be considered as a special case of CWO.

A template pseudo-code for the MPE approximation in a Comraf is given in Algorithm 1. For each combinatorial r.v. X_i^c in the Comraf, we first select and fix its initial value as a point in the event space L_i. We then round-robin over each X_i^c, for which we search for an advantageous path in L_i. When this path is constructed, we fix its destination point to be a new value of X_i^c and move to another node. We

repeat this procedure l times. To transform this template into an actual algorithm, we need to make the following choices:

- How to select initial values for each combinatorial r.v. in the Comraf.
- How to determine an ordering for variables in the optimization procedure (and an ordering of cliques in CWO). One obvious approach is a plain or weighted round-robin, but more sophisticated choices can also be made.
- How to construct an advantageous path in L.

We will address these points in the following chapters of this thesis.

3.4 Summary

While technically being graphical models, Comrafs are very different from existing graphical models: all Comraf models we propose in this thesis are small models with 'rich' nodes, while existing graphical models are usually large models with 'simple' nodes. No existing inference techniques are applicable to Comrafs (as they cannot deal with nodes as complex as combinatorial r,v,'s), so we have developed a new inference framework for Comrafs.

The major advantage of Comrafs over existing graphical models is that Comrafs provide a more flexible modeling environment: existing models are able to model data only in terms of the graph G, while their objective function and their inference algorithm are generic rather than task-specific. Usually, this property is not considered to be a drawback of the graphical model framework: once the graph G is designed for a certain task, it is straightforward to apply an existing inference method to this graph. However, existing inference methods are approximations to the NP-hard inference problem, and thus make various assumptions that can potentially be inappropriate for the particular task being solved. The main disadvantage of generic

inference methods is that they make the *same* assumptions for *every* task. A practitioner can choose one of a handful of existing inference methods (such as mean-field, variational approximation, belief propagation, Gibbs sampling etc. [58]) for her task, some of which can work better for this task while some can work worse, but none is *specific* for the task.

Comrafs, in contrast, have three degrees of freedom: designing the graph G, the objective and the inference algorithm, all specific for the task in hand. And as we will show below, this flexibility leads to constructing models that demonstrate excellent performance on various unsupervised and semi-supervised learning tasks.

CHAPTER 4

COMRAFS FOR MULTI-MODAL CLUSTERING

Multi-modal (hard) clustering is a problem of simultaneously constructing m partitionings of m data modalities, e.g. of documents, their words, authors, titles etc. When clustering modalities simultaneously, one can overcome the statistical sparseness of the data representation, leading to a dense, smoothed joint distribution of the modalities that would result in (hypothetically) more accurate clusterings than the ones obtained when each modality is clustered separately. Based on our previous work [10, 12], we will empirically justify this hypothesis. In this chapter, we propose a Comraf model for multi-modal clustering (for motivation and discussion, see Chapter 1). Let us first introduce the notation.

Let $s_1, s_2, ..., s_N$ be a dataset of N i.i.d. samples drawn from some discrete distribution. Let $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ be the set of n unique values comprising the event space from which samples s_i are drawn. We now define a random variable X such that $P(X = x_i)$ is given by the empirical frequencies of samples with value x_i in the dataset (i.e., X has a multinomial distribution estimated using maximum likelihood).

Define a *hard clustering* x^c to be a partitioning of $\mathcal{X}$. Let $\mathcal{X}^c = \{x_1^c, x_2^c, ..., x_K^c\}$ be the combinatorial set of all K partitionings of $\mathcal{X}$, where K is exponential in the size of $\mathcal{X}$. We will refer to the subsets of the j-th partitioning x_j^c as $\{\tilde{x}_{j,1}, \tilde{x}_{j,2}, ..., \tilde{x}_{j,k_j}\}$. That is, the first subscript in $\tilde{x}_{j,i}$ is the index of a particular partitioning, and the second subscript is the index of a subset (*a cluster*) within that partitioning.

Define $\tilde{X}_j$ to be a random variable over the subsets (clusters) in a partitioning x_j^c, with the probability of selecting a cluster defined as the probability of selecting

any one of its members: $P(\tilde{X} = \tilde{x}_{j,i}) = \sum_{x \in \tilde{x}_{j,i}} P(x)$. Finally, define X^c to be a combinatorial r.v. with the event space $\mathcal{X}^c$. In this thesis, we shall use parallel notation for different modalities of data, replacing the "x's" in the above notation with variables appropriate for the data source. For example, w_i would represent a specific word in a dataset, $\tilde{w}$ would be a cluster, w^c would be a partitioning of words, and so on.

4.1 Choosing an objective function

As discussed in Section 3.2, interactions between combinatorial r.v.'s are represented by edges in a Comraf graph. To use the objective from Equation (3.1), we should choose relevant cliques in the Comraf graph and define log-potential functions over these cliques. To make the inference feasible, we consider only the smallest cliques, i.e. adjacent pairs. Since our inference objective allows us to use complicated log-potential functions (see, again, Section 3.2), we use the *mutual information (MI)* between r.v.'s defined over values of adjacent nodes. Let x_i^c and x_j^c be such values (particular partitionings of two modalities). A log-potential is then defined:

$$f(x_i^c, x_j^c) = I(\tilde{X}_i; \tilde{X}_j) = \sum_{i',j'} P(\tilde{x}_{i,i'}, \tilde{x}_{j,j'}) \log \frac{P(\tilde{x}_{i,i'}, \tilde{x}_{j,j'})}{P(\tilde{x}_{i,i'}) P(\tilde{x}_{j,j'})}. \tag{4.1}$$

Our motivation for choosing MI as a log-potential function is as follows: a linear combination of MI terms has traditionally been used as a clustering criterion, both in uni-modal clustering methods, such as Information Bottleneck (IB) [106], and in bi-modal methods [35]. Slonim et al. [97] generalize the IB clustering criterion to a multivariate case: in place of mutual information, they use Multi-Information[1]

$$\mathbf{I}(\tilde{X}_1; \ldots; \tilde{X}_m) = \sum_{i'_1, \ldots, i'_m} P(\tilde{x}_{i_1,i'_1}, \ldots, \tilde{x}_{i_m,i'_m}) \log \frac{P(\tilde{x}_{i_1,i'_1}, \ldots, \tilde{x}_{i_m,i'_m})}{P(\tilde{x}_{i_1,i'_1}) \ldots P(\tilde{x}_{i_m,i'_m})}, \tag{4.2}$$

[1]For alternative definitions and discussions on Multi-Information, see [114, 54].

which naturally factorizes over a directed graphical model. With little effort, we can show that Multi-Information also factorizes over a tree-structured undirected graphical model, reducing to a sum of pairwise MI terms defined over edges of the tree. However, in the case of an arbitrary Comraf graph, Multi-Information cannot be fully factorized. In general, objective functions based on high order statistics (including Multi-Information) are problematic for loopy Comraf graphs. From a statistical viewpoint, it is not clear whether we can extract reliable estimates for the full joint distribution $P(\tilde{X}_1, \ldots, \tilde{X}_m)$. Still, we can *approximate* Multi-Information by a sum of pairwise MI terms. Estimating the quality of such an approximation remains an open question.

Thus, substituting log-potentials (4.1) into the MPE inference model (3.1), our objective function for multi-modal clustering with Comrafs is then:

$$\mathbf{x}^{\mathbf{c}*} = \arg\max_{\mathbf{x}^{\mathbf{c}}} P(\mathbf{x}^{\mathbf{c}}) = \arg\max_{\mathbf{x}^{\mathbf{c}}} \sum_{(X_i^c, X_{i'}^c) \in \mathbf{E}} I(\tilde{X}_i; \tilde{X}_{i'}). \tag{4.3}$$

This maximization is performed subject to constraints on the cardinalities $k_i = |\tilde{X}_i|$, $i = 1, \ldots, m$ (i.e., the number of clusters is fixed). Without these constraints, the maximization would lead to a degenerative case of all singleton clusters. Note that these constraints do not necessarily imply the use of a *flat* clustering scheme (see Chapter 2). In a particular clustering algorithm, clusters can be split or merged, after which the number of clusters is fixed and the optimization of the objective function (4.3) is performed.

We apply the ICM scheme (see Section 3.3) to multi-modal clustering: we iterate over combinatorial r.v.'s in the Comraf graph, and at each iteration (over node X_i^c) we construct the most likely clustering x_i^{c*} by optimizing the objective function (4.3). It is important to note that in the general case the objective (4.3) has $O(|\mathbf{X}|^2)$ terms. However, at each ICM iteration only one node is optimized, therefore the objective

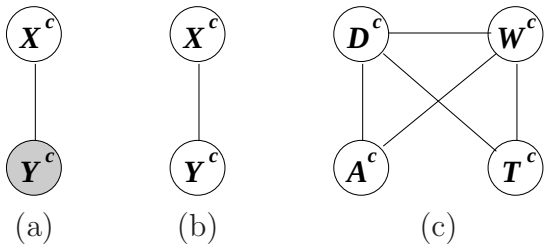


Figure 4.1. A Comraf graphs for: (a) hard version of Information Bottleneck; (b) information-theoretic co-clustering; (c) one of the possible 4-modal Comrafs.

is reduced to:

$$x_i^{c*} = \arg\max_{x_i^c} \sum_{i':\ (X_i^c, X_{i'}^c) \in \mathbf{E}} I(\tilde{X}_i; \tilde{X}_{i'}) \tag{4.4}$$

that sums over only $O(|\mathbf{X}|)$ neighbors of X_i^c (i.e. its Markov blanket).

The resulting model has two important special cases:

- **A hard version of Information Bottleneck** [106]. In Information Bottleneck, given two modalities $\mathcal{X}$ and $\mathcal{Y}$, a clustering x^{c*} is constructed that maximizes information about Y (and minimizes information about X):

$$x^{c*} = \arg\max_{x_j^c} \Big(I(\tilde{X}_j; Y) - \beta I(\tilde{X}_j; X) \Big), \tag{4.5}$$

  where β is a Lagrange multiplier. The *compression constraint* $I(\tilde{X}_j; X)$ can be omitted if the number of clusters is fixed: $|\tilde{X}_j| = k$. Consider graph G in Figure 4.1(a), where a shaded Y^c represents an *observed* variable.[2] Over the only clique in G, we define one log-potential which is the mutual information $I(\tilde{X}_j; Y)$. The MPE optimization objective for such Comraf is then:

$$x^{c*} = \arg\max_{x_j^c} P(x_j^c, y^c) = \arg\max_{x_j^c} I(\tilde{X}_j; Y),$$

[2]For discussion on observed variables see Chapters 5 and 7.

subject to $|\tilde{X}_j| = k$, which is clearly equivalent to the Information Bottleneck objective (4.5).

- **Information-theoretic co-clustering** [35] is a task of simultaneously clustering two modalities $\mathcal{X}$ and $\mathcal{Y}$, while minimizing the information loss $I(X;Y) - I(\tilde{X}_j, \tilde{Y}_j)$ under the constraint $|x_j^c| = k_1$ and $|y_j^c| = k_2$. Note that $I(X;Y)$ is a constant for a given dataset. This scheme is a special case of a Comraf as well: given graph G in Figure 4.1(b), in analogy to the Comraf model of Information Bottleneck, we define the only log-potential $I(\tilde{X}_j; \tilde{Y}_j)$. Then the information-theoretic co-clustering can be represented as an MPE inference in this Comraf:

$$(x^{c*}, y^{c*}) = \arg\max_{x_j^c, y_j^c} P(x_j^c, y_j^c) = \arg\max_{x_j^c, y_j^c} I(\tilde{X}_j; \tilde{Y}_j).$$

4.2 Clustering as combinatorial optimization

Given a variable X with n values clustered into k clusters, the combinatorial r.v. X^c has k^n values, all of which can be represented as points in an n-dimensional lattice L: a point $x^c = (i_1, i_2, \ldots, i_n)$ corresponds to the fact that value x_1 of X belongs to the i_1-th cluster, value x_2 belongs to the i_2-th cluster, ..., value x_n belongs to the i_n-th cluster.[3] In the lattice L there is a (possibly non-unique) point $x^{c*} = (i_1^*, i_2^*, \ldots, i_n^*)$ which is most likely. Since the lattice consists of an exponential number of points, the task of finding the most likely point can be computationally hard. In lattice L, a *transaction* (see Definition 3.3.1) is interpreted as an operation of transferring a value x_j from cluster $\tilde{x}_i$ to cluster $\tilde{x}_{i'}$, i.e. $(\ldots, i_j, \ldots) \to (\ldots, i'_j, \ldots)$, where $i_j \neq i'_j$.

[3]Recall that we consider only *hard* clustering: $P(\tilde{x}_{i_j}|x_j) = 1$, that is, a value x_j is assigned only to the i_j-th cluster.

Note that we can view both splits and mergers of clusters as transactions. A split of a cluster $i_{j'}$ is a transaction $(\ldots, i_{j'}, \ldots) \to (\ldots, i'_{j'}, \ldots)$, where $\exists j \neq j' : \ i_{j'} = i_j$ and $\forall j \neq j' : \ i'_{j'} \neq i_j$. That is, cluster $i_{j'}$ contained at least two elements (x_j and $x_{j'}$), one of which ($x_{j'}$) has been transferred into a newly created cluster $i'_{j'}$. A merger of clusters $i_{j'}$ and $i'_{j'}$ is a transaction $(\ldots, i_{j'}, \ldots) \to (\ldots, i'_{j'}, \ldots)$, where $\exists j \neq j' : \ i'_{j'} = i_j$ and $\forall j \neq j' : \ i_{j'} \neq i_j$, i.e. cluster $i_{j'}$ contained only one element that has been added to the existing cluster $i'_{j'}$ so that the cluster $i_{j'}$ does not exist anymore. These operations will help us to represent both agglomerative (bottom-up) and divisive (top-down) clustering schema as inference in Comrafs.

By applying splits, mergers and other transactions, we construct a path in the lattice L. Our goal is to make this path as advantageous as possible, such that a clustering at the end of this path will be the most probable clustering that could be found. Thus, we view the process of clustering a set $\mathcal{X}$ as an MPE approximation of a combinatorial r.v. X^c, where the MPE is approximated using a local search in the lattice L. To perform the local search, we apply the simplest, greedy combinatorial optimization method—*hill climbing*: at *each* ICM iteration, we attempt to construct the most advantageous path in L, given the available computational resources.

Let us discuss particular algorithms in more detail in the next section.

4.3 Multi-way Distributional Clustering (MDC)

In this section we describe our scheme for clustering m modalities that aims at maximizing our objective function (4.3). This scheme is called Multi-way Distributional Clustering (MDC) [10]. Let G be a Comraf graph over combinatorial random variables X_i^c, $i = 1, \ldots, m$. For each edge $e_{ii'}$ in graph G we are given a contingency table $T_{ii'}$ that provides the corresponding co-occurrence counts of the modalities X_i and $X_{i'}$. The input to the algorithm is the graph G, the tables $T_{ii'}$, as well as m desired cardinalities $k_1, \ldots, k_m$ of the final partitionings, and a *clustering schedule*

(the sequence of variables for optimization in the ICM loop, see below for details). The output of the algorithm is m partitionings $\tilde{X}_i$, $i = 1, \ldots, m$, each of cardinality $k_i = |\tilde{X}_i|$.

The desired cardinalities $k_1, \ldots, k_m$ are essential parameters of MDC, as our method cannot infer them. We believe that the problem of inducing the optimal number of clusters is generally ill-defined: imagine a dataset that is situated on a plane in a triangle, each corner of which consists of three triangles of data instances (27 instances overall). It is hard to decide what the best number of clusters would be in this case: three or nine. Admittedly, not all machine learning researchers would agree with this argument. Some existing clustering methods attempt to solve the problem of optimal number of clusters (such as, e.g., the Chinese Restaurant Process [30]). Still and Bialek [101] come up with the optimal number of clusters in an Information Bottleneck setting. While their method is well justified theoretically, it could not induce a meaningful number of clusters in our experiments.

To compute the objective function (4.3) we will need the following definitions and identities, where for the current discussion we re-notate $X = X_i$, $Y = X_j$ and $T = T_{ii'}$:

$$\begin{aligned} N_{XY} &= \sum_{x \in X;\ y \in Y} T(x, y); \\ p(\tilde{x}, \tilde{y}) &= \frac{1}{N_{XY}} \sum_{x \in \tilde{x};\ y \in \tilde{y}} T(x, y); \\ I(\tilde{X}; \tilde{Y}) &= \sum_{\tilde{x} \in \tilde{X}; \tilde{y} \in \tilde{Y}} p(\tilde{x}, \tilde{y}) \log \frac{p(\tilde{x}, \tilde{y})}{p(\tilde{x}) p(\tilde{y})}, \end{aligned} \tag{4.6}$$

where $p(\tilde{x}) = \sum_{\tilde{y} \in \tilde{Y}} p(\tilde{x}, \tilde{y})$, and $p(\tilde{y}) = \sum_{\tilde{x} \in \tilde{X}} p(\tilde{x}, \tilde{y})$.

Pseudo-code for the *multi-way distributional clustering (MDC)* algorithm is given in Algorithm 2. For simplicity, the pseudo-code abstracts away several details that

Input:
G – Comraf graph of nodes $\{X_1^c, \ldots, X_m^c\}$ and edges $\mathbf{E}$
$T_{ii'}$ – contingency tables for each $e_{ii'} \in \mathbf{E}$
S_{up}, S_{down} - bottom-up/top-down partition of $\{1, \ldots, m\}$
$S_l = i_1, i_2, \ldots, i_l$ – clustering schedule, where each $i_j \in \{1, \ldots, m\}$
Output:
Most likely clusterings $x_{1,l}^c, \ldots, x_{m,l}^c$
Initialization:
For each $i = 1, \ldots, m$ **do**
  If $i \in S_{down}$ **then**
    Place all values of X_i in one cluster
  Else If $i \in S_{up}$ **then**
    Place each value of X_i in a singleton cluster
Main loop:
For each i_j from S_l **do**
  Split/merge phase:
  If $i_j \in S_{down}$ **then**
    Split each cluster in $x_{i,j}^c$ uniformly at random to two halves
  Else If $i_j \in S_{up}$ **then**
    Merge each cluster in $x_{i,j}^c$ with its closest peer
  Optimization phase:
  For each values x of X_{i_j} **do**
    Pull x out of its current cluster
    Place x into a cluster, s.t. objective function (4.4) is maximized

Algorithm 2: Multi-Way Distributional Clustering (MDC).

are not essential for understanding the general idea but can be important for actual applications. We now discuss the algorithm and then provide the necessary details.[4]

The main loop of the algorithm is controlled by two parameters:

- Partition (S_{up}, S_{down}) of the set of variable indices. If $i \in S_{up}$, then the variable X_i is clustered using a bottom-up procedure. Otherwise (i.e. $i \in S_{down}$), X_i is clustered via the top-down procedure.

- *Clustering schedule* $S_l = i_1, \ldots, i_l$, which is a sequence of variable indices. The schedule S_l determines the order of processing the variables. While this mechanism allows for great flexibility, we always apply it in a straightforward manner where the sequence S_l specifies a (weighted) round-robin schedule. For example, in the case of bi-modal clustering (with two variables X_1 and X_2), we

[4]An efficient C++ implementation of MDC that was used in our experimental study can be downloaded from `http://sourceforge.net/projects/comraf`.

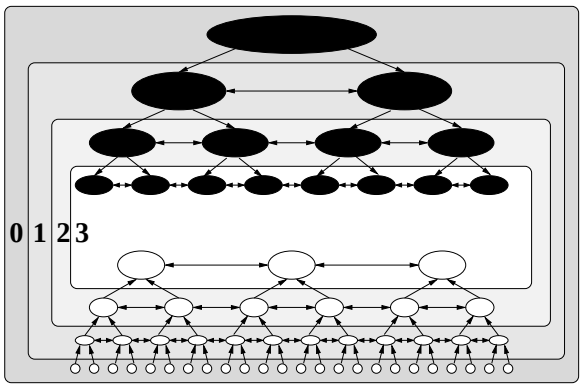


Figure 4.2. A schematic view of bi-modal MDC with a simple, non-weighted round-robin schedule. At each iteration black clusters are split and then white clusters are merged.

take (ignoring, for the moment, the desired cluster cardinalities) $S_{down} = \{1\}$, $S_{up} = \{2\}$ and $S_l = 1, 2, 1, 2, \ldots, 1, 2$. A schematic view of MDC (for this bi-modal instance) is given in Figure 4.2.

We propose two versions of the optimization phase of our algorithm: *sequential* and *shuffled*:

- In the sequential version, we iterate over all values x_i of X_i, in a random order (determined via a permutation selected uniformly at random). We assign x_i into its "best" cluster, i.e. such cluster that the objective from Equation (4.4) is maximized. Note that this optimization routine is similar to and inspired by the *sequential Information Bottleneck (sIB)* clustering algorithm [96]. We then iterate over all the values of X_i once again, in order to further optimize the objective, i.e. two optimization passes are performed overall.

- In the shuffled version, we repeat the following step a predefined number of times:[5] we uniformly at random select a data point x_i and a cluster $\tilde{x}_j$, and assign x_i into $\tilde{x}_j$ if this transaction increases the value of our objective. The

[5]We set it equal (for fair comparison) to the number of iterations in the sequential version.

shuffled approach opens the door to improving scalability of MDC (the number of iterations is constant and can be chosen arbitrarily small, at the cost of decreasing performance) and to parallelization.

Note that both sequential and shuffled procedures can never decrease the objective function. However, cluster mergers usually decrease it, so the optimization is non-convex in the general case.

The choice of index partition (S_{up}, S_{down}) is based on the following two crucial observations. First, for practical applications it is computationally infeasible to apply bottom-up procedures for all the variables. Second, applying only top-down procedures is likely to be useless, in terms of the clustering quality. This is easy to see when considering bi-modal applications, with respect to two variables X and Y. The objective function reduces to $I(\tilde{X};\tilde{Y})$ and we start with x^c and y^c each being a single cluster containing all points. Clearly, in this case $I(\tilde{X};\tilde{Y}) = 0$. We now split $\tilde{X}$ to get $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2\}$. For any $(\tilde{x}_1, \tilde{x}_2)$-partition we have $H(\tilde{Y}|\tilde{X}) = -\sum_i p(\tilde{x}_i, \tilde{Y}) \log p(\tilde{Y}|\tilde{x}_i) = 0$, since $p(\tilde{Y}|\tilde{x}_i) = 1$. Therefore, $I(\tilde{X};\tilde{Y}) = H(\tilde{Y}) - H(\tilde{Y}|\tilde{X}) = H(\tilde{Y}) = 0$, and the corrective step of the algorithm is useless here. The subsequent split of $\tilde{Y}$ strictly optimizes the objective function, but the resulting clustering is optimized to correlate with the initial random split of the X variable. This way, all the subsequent partitions are optimized with respect to a meaningless, random partition. A similar argument applies to the general MDC and implies that at least one of the clustering procedures must *not* be computed top-down. A natural choice for clustering this variable would be a bottom-up method because its initialization phase (singleton clusters) does not require any prior knowledge to be incorporated (for a discussion, see Chapter 2).

As mentioned above, in all our applications we construct (weighted) round robin schedules $S_l = i_1, \ldots, i_l$. In order to accommodate the required cardinalities $k_1, \ldots, k_m$ of clusterings $x_1^c, \ldots, x_m^c$, the MDC algorithm performs the following number of iter-

ations: $l_i = \lceil \log k_i \rceil$ for $i \in S_{down}$, and $l_i = \lceil \log(|X_i|/k_i) \rceil$ for $i \in S_{up}$. Thus, each index i appears l_i times in the sequence S_l, in a (weighted) round-robin fashion.

4.3.1 Computational complexity of MDC

We now analyze the time complexity of the sequential version of MDC[6] for a non-weighted round-robin schedule. The complexity issue should be taken into account when forming the partition (S_{up}, S_{down}), because the time complexity of the algorithm depends on $u = |S_{up}|$, i.e. on the number of modalities clustered bottom-up. Let $|X| = \max(|X_1|, \ldots, |X_m|)$, the size of the largest support of variables $X_1, \ldots, X_m$. At each iteration, (sequential) MDC performs three nested loops:

1. Pass over each value of X_i: $O(|X|)$ times;
2. For each value of X_i, pass over each cluster in $\tilde{X}_i$: $|\tilde{X}_i| = O(|X|)$ times;
3. For each cluster in $\tilde{X}_i$, pass over clusters in all the other clusterings (excluding $\tilde{X}_i$ itself): $O(m|X|) = O(|X|)$ times (the number of clustered variables, m, is a constant in our case).

Since the number of iterations is $n = O(\log |X|)$, in the worst case (when $u > 1$) the time complexity is $O(n|X|^3) = O(|X|^3 \log |X|)$. This complexity can be burdensome in some real-world applications. Note, however, that for each variable X_i, which is clustered top-down, at each iteration j the number of clusters is $|X_{i_j}| = O(k_i) = O(1)$. Thus, when $u = 1$, either loop 2 or loop 3 is performed $O(1)$ times, and the overall running time is $O(|X|^2 \log |X|)$, which is affordable for many applications.

In the bi-modal case, at each iteration the size of one clustering is doubled, and at the next iteration the size of the other clustering is halved. Therefore, at each

[6]The complexity of our implementation of the shuffled version is the same as the one of the sequential version, because we choose to fix the number of iterations in the shuffled version equal to the number of iterations in the sequential version.

iteration $|\tilde{X}_1| \cdot |\tilde{X}_2| \leq 2|X|$, i.e. the constant under the 'big-O' is only 2. The (non-hierarchical) co-clustering algorithm of [35] has the same complexity $O(|X|^2 \log |X|)$, but with a larger constant under the 'big-O'.

Based on this analysis, in all our experiments we fix $u = 1$, i.e., only one variable is clustered bottom-up. Finally, note that if variable X_i has a small support, $|X_i| \ll |X|$, then the decision whether $i \in S_{up}$ or $i \in S_{down}$ can be made independently of time complexity considerations.

4.4 Clique-wise MDC

As we discussed in Section 3.3, global optimization of the objective function (3.1) is not always beneficial. As an alternative, we proposed a clique-wise optimization (CWO) procedure. In this section, we propose a clique-wise version of the MDC algorithm, which is inspired by Bouvrie's algorithm [23]. Its pseudocode is given in Algorithm 3. To keep the procedure as simple as possible, we consider only the smallest cliques, i.e. edges in the Comraf graph G. In contrast to the original MDC that iterates over nodes in G, the CWO version iterates over *edges*, in a round-robin fashion. For every edge $e_{ii'}$, the algorithm performs the MPE optimization of a portion of G that consists of only one edge $e_{ii'}$ and its vertices X_i^c and $X_{i'}^c$. This optimization is performed independently of the rest of the model. The best values of X_i^c and $X_{i'}^c$ found during this optimization step are then used as initial values for the next optimization steps.

In this setup, an application of hierarchical clustering appears unnatural: after the j-th optimization iteration over one clique, the constructed clusterings $x_{i,j}^c$ and $x_{i',j}^c$ are supposed to have the desired number of clusters (k_i and $k_{i'}$ respectively). Using these clusterings as initial values of further optimization steps leaves no room for exploring the clustering hierarchy. For this reason, and also for simplicity, at each optimization step we apply a *flat* clustering method, similar to the sequential Infor-

Input:
G – Comraf graph of nodes $\{X_1^c, \ldots, X_m^c\}$ and edges $\mathbf{E}$
$T_{ii'}$ – contingency tables for each $e_{ii'} \in \mathbf{E}$
$k_1, \ldots, k_m$ – the desired number of clusters for each node
$S'_l = (i_1 i'_1), \ldots, (i_l i'_l)$ – clustering schedule, where each pair ii' corresponds to edge $e_{ii'}$
Output:
Most likely clusterings $x^c_{1,l}, \ldots, x^c_{m,l}$
Initialization:
For each $i = 1, \ldots, m$ **do**
  Distribute all values of X_i uniformly at random over k_i clusters
Main loop:
For each $(i_j i'_j)$ from S'_l **do**
  For each value x of X_{i_j} **do**
    Pull x out of its current cluster
    Place x into a cluster, s.t. $I(\tilde{X}_{i_j}; \tilde{X}_{i'_j})$ is maximized
  For each value x of $X_{i'_j}$ **do**
    Pull x out of its current cluster
    Place x into a cluster, s.t. $I(\tilde{X}_{i_j}; \tilde{X}_{i'_j})$ is maximized

Algorithm 3: Clique-wise MDC.

mation Bottleneck [96]. Quite surprisingly, the results of this flat clustering procedure are comparable to the ones of the original (more complex) MDC (see Section 4.6.5).

The computational complexity of each sequential optimization step is $O(k^2|X|)$, where $|X|$ is the size of the largest support among the variables in $\mathbf{X}$, and k is the largest final number of clusters. The number of iterations is $O(m^2)$, as the number of edges in graph G is in the worst case quadratic in the number of combinatorial random variables. The resulting complexity is then $O(m^2k^2|X|)$, which is asymptotically linear in the size of the data. However, the constants can be very large. Still, in practical cases, Comraf models are very compact such that the m^2 constant is not restrictive, and the clique-wise MDC is substantially faster than its original ICM-based version.

4.5 Related work

The study of distributional clustering based on co-occurrence data using information theoretic objective functions was initiated by Pereira et al. [88]. Much of the subsequent related work is inspired by that paper and the Information Bottleneck

(IB) ideas of Tishby et al. [106]. In this context, the first work considering two-way clustering of both words and documents is by Slonim and Tishby [99], which is subsequently improved by El-Yaniv and Souroujon [39], and then more thoroughly studied by Dhillon et al. [35].

The more general Multivariate Information Bottleneck (mIB) framework [97] also considers simultaneous clusterings based on interaction between variables, as we propose here. For two variables (bi-modal clustering) the algorithm proposed here can be viewed as a particular implementation of the "hard case" of mIB. However, for more than two variables, the framework we propose here is not a special case of the mIB framework since the interactions between variables in mIB are described via a directed Bayesian network, in which cycles cannot be factorized to pairwise dependencies (see Section 4.1). Our scheme employs undirected graphs that represent pairwise interactions, and therefore do not preclude loops. It is important to note that our clustering algorithm (MDC) is inspired by the sequential IB method [96]. Finally, we note that the idea of multi-modal clustering also appears in Bouvrie [23], where multiple clusterings are constructed by an iterative application of a bi-modal clustering algorithm, and the resulting system is applied to computer vision tasks.

4.6 Experimentation: email clustering

In this section, we present our experimental results on the document clustering task. Two particular tasks we consider are similar to each other: (a) automatic categorization of email into folders; (b) automatic routing of newsgroup messages into appropriate newsgroups.

Email foldering is a rich and multi-faceted problem, with many difficulties that make it different from traditional topic-based categorization. Email users create new folders, and let other folders fall out of use. Email folders do not necessarily correspond to simple semantic topics—sometimes they correspond to unfinished todo tasks,

project groups, certain recipients, or loose agglomerations of topics. It is important to note that email content and foldering habits differ drastically from one email user to another—so while automated methods may perform well for one user, they may fail horribly for another. In this thesis, however, we test the Comraf's performance on email clustering under a simplifying assumption that folders roughly correspond to semantic topics. In our future work, we will adapt our clustering system to specific needs of particular users.

Despite the fact that clustering is rarely used as a stand-alone application—it is usually a part of another, more global task—we choose to focus on evaluating the quality of the clustering results *per se*, i.e. not with respect to the global task. This way, our evaluation is not skewed by various aspects of a particular real-world problem.

4.6.1 Evaluation measure

Following [96, 35] and many other works, we use *micro-averaged accuracy* for evaluation of our clustering methods. Let x^c be a clustering of the data $\mathcal{X}$. Let $\mathcal{T}$ be the set of ground truth categories. We fix the number of clusters to match the number of categories $|x^c| = |\mathcal{T}| = k$. For each cluster $\tilde{x}_j$, let $\gamma_{\mathcal{T}}(\tilde{x}_j)$ be the maximal number of $\tilde{x}_j$'s elements that belong to one category. Then, accuracy $Acc(\tilde{x}_j, \mathcal{T})$ of a cluster $\tilde{x}_j$ with respect to $\mathcal{C}$ is defined as $Acc(\tilde{x}_j, \mathcal{T}) = \gamma_{\mathcal{T}}(\tilde{x}_j)/|\tilde{x}_j|$. The micro-averaged accuracy of the clustering x^c is:

$$Acc_m(x^c, \mathcal{T}) = \frac{\sum_{j=1}^{k} \gamma_{\mathcal{T}}(\tilde{x}_j)}{\sum_{j=1}^{k} |\tilde{x}_j|} = \frac{\sum_{j=1}^{k} \gamma_{\mathcal{T}}(\tilde{x}_j)}{|\mathcal{X}|}. \tag{4.7}$$

In Section 4.6.5 also present *macro-averaged accuracy* results, where the macro-averaged accuracy is defined as:

$$Acc_M(x^c, \mathcal{T}) = \frac{\sum_{j=1}^{k} Acc(\tilde{x}_j, \mathcal{T})}{k}. \tag{4.8}$$

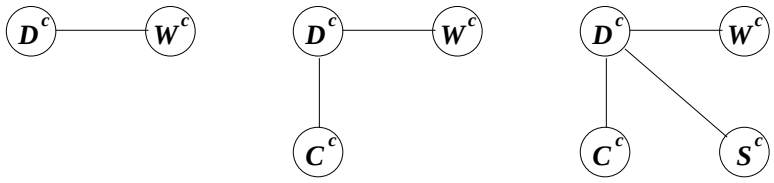


Figure 4.3. Comraf graphs for 2-modal, 3-modal and 4-modal Comrafs used in our experiments. We consider interactions between combinatorial random variables that correspond to documents D^c, words W^c, email correspondents C^c and email *Subject* lines S^c. Note that we use only tree-structured models, as they are simpler than loopy models and on the email foldering task they show comparable results to those obtained with loopy models (see Section 4.6.6 for a discussion). In Section 4.8 we present a result when a loopy model is significantly superior to a tree-structured one.

4.6.2 Datasets

We evaluate the Comraf models on six text datasets. In addition to the standard benchmark 20 Newsgroups dataset (20NG) we use five real-world email directories. Three of them belong to participants in the CALO project[7] and the other two belong to former Enron employees.[8]

On the 20NG dataset we apply a bi-modal Comraf where the modalities are messages (documents) and words. CALO and Enron datasets are particularly useful for evaluating 3-modal and 4-modal Comrafs. Here we take as variables (1) messages; (2) words; (3) people names associated with messages—we consider the entire list of correspondents (both senders and recipients); and (4) email *Subject* lines, represented by their bags of words. Comraf graphs for the three setups are given in Figure 4.3.

Table 4.1 provides basic statistics of the six datasets. For details on collecting the CALO and Enron data, see [14]. Below we briefly describe the data and preprocessing steps undertaken.

---

[7] `http://www.ai.sri.com/project/CALO`

[8] The Web page of the original Enron Email Dataset is `http://www.cs.cmu.edu/~enron`. Our preprocessed Enron email directories can be obtained from `http://www.cs.umass.edu/~ronb/enron_dataset.html`.

Dataset	Size	Min/max class size	Number of distinct words	Number of correspondents	Number of classes
CALO:ACHEYER	664	3/72	2863	67	38
CALO:MGERVASIO	777	6/116	3207	61	15
CALO:MGONDEK	297	3/94	1287	50	14
ENRON:KITCHEN-L	4015	5/715	15579	2278	47
ENRON:SANDERS-R	1188	4/420	5966	933	30
20NG	19997	997/1000	39764	N/A	20

Table 4.1. Statistics of email datasets. Number of distinct words and number of correspondents are after preprocessing.

4.6.2.1 20 Newsgroups

The 20 Newsgroups (20NG) corpus contains 19997 messages taken from the Usenet newsgroups collection.[9] Each message is assigned into one or more semantic categories and the total number of categories is 20, all of which are of about the same size. Most of the documents have only one semantic label, however it turns out that about 4.5% of documents have two or more labels. Those documents are simply duplicated in the dataset (one copy per category). In this thesis, for easier replicability of our results, we decided to refrain from taking steps of any kind to resolve the duplication issue.

We preprocess the 20NG dataset as described in [11]. First, we remove message headers and markup (such that only the subject lines and actual text remained). Next, we filter out lines that seem to be part of binary files sent as attachments or pseudo-graphical text delimiters. A line is considered to be a "binary" (or a delimiter) if it is longer than 50 symbols and contains no white spaces. Overall, we remove 23057 such lines (most of them appeared in a handful of articles). Finally, we represent documents as their Bags-Of-Words, lower the case of letters and remove stopwords as well as low-frequency words.

[9] `http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html`

4.6.2.2 Enron Email Dataset

The archived email from many of the senior management of Enron Corporation was subpoenaed, and is now in the public record. The data consists of over 500,000 email messages from the email accounts of 150 people. The dataset is provided by SRI after major clean-up and removal of attachments. The dataset version we use was released on February 3, 2004.

Although the size of the dataset is large, many users' folders are sparsely populated. We use the email directories of two former Enron employees: KITCHEN-L and SANDERS-R. Those directories are among the largest ones in the dataset.

We remove standard non-topical folders *"all_documents", "calendar", "contacts", "deleted_ items", "discussion_threads", "inbox", "notes_inbox", "sent", "sent_items"* and *"_sent _mail"*. We then flatten all the folder hierarchies and remove all the folders that contain fewer than three messages. We also remove the *X-folder* field in the message headers that actually contains the class label. As for 20NG, we finally represent documents as their Bags-Of-Words, lower the case of letters and remove stopwords and low-frequency words.

4.6.2.3 CALO Email Dataset

A smaller but also significant corpus of real-world, foldered email has been created as part of the CALO DARPA/SRI research project. This corpus consists of snapshots of the email folders of 196 users, containing approximately 22,000 messages. From the February 2, 2004 snapshot of CALO directories, we select three users with large number of messages: ACHEYER, MGERVASIO, and MGONDEK. As in the preprocessing step of the Enron datasets, we first remove standard non-topical folders (*"Inbox", "Drafts", "Sent"* and *"Trash"*). Then the folder hierarchy is flattened, and folders that contain fewer than three messages are removed. Finally, as for all the other

datasets, we represent documents as BOW, lowercase the text and filter out stopwords and low-frequency words.

4.6.3 Baseline algorithms

We compare the performance of Comraf clustering algorithms with the following five well known benchmark clustering algorithms:

1. **K-means.** We use the `SimpleKMeans` implementation of WEKA [112];

2. **Agglomerative Information Bottleneck (aIB).** A simple, deterministic *uni-modal* Information Bottleneck clustering algorithm [98];

3. **Sequential Information Bottleneck (sIB).** A randomized *uni-modal* Information Bottleneck clustering algorithm [96], which exhibited striking performance in the text domain;

4. **Information-theoretic co-clustering (ITCC).** A *bi-modal* clustering algorithm [35] (see Section 4.1);

5. **Latent Dirichlet Allocation (LDA).** A popular generative model for representing document collections, proposed by Blei et al. [22]. Each document is represented as a distribution of topics, and parameters of those distributions are learned from the data. Documents are then clustered based on their posterior distributions (given the topics). We use Xuerui Wang's LDA implementation [78] that applies Gibbs sampling with 10000 sampling iterations.[10]

Note that the latter three algorithms are widely considered to be state-of-the-art in unsupervised text categorization.

[10]We also tried David Blei's LDA-C [22] that implements variational approximation and obtained significantly inferior accuracy.

To gain some perspective on the performance of the unsupervised methods we tested, we also report on the results of a trivial "random clustering", which simply places each document in a random cluster. At the other extreme, we report on the categorization accuracy of a *supervised* application of a support vector machine (SVM), applied with linear kernel and with cross-validated parameter tuning (using the same setup as described in Bekkerman et al. [11]). We stress that the supervised categorization accuracy cannot be directly compared with the clustering accuracy, however, it provides some perspective on on datasets' "complexities".

4.6.4 Implementation details

The following technical details are important for replicating our experimental results:

1. Unless stated otherwise, we use the bottom-up scheme for documents and the top-down scheme for all the other clusterings.

2. As discussed in Section 4.3, we merge each document cluster with its closest peer. Following Slonim & Tishby [98], we choose the *Jensen-Shannon divergence* between clusters as the underlying "metric".

3. At the MDC's last iteration (at which the required number of document clusters is obtained), we apply the optimization routine after merging *each* pair of clusters.

4. We perform 10 random restarts at each iteration of MDC. For a fair comparison, we perform the same number of random restarts in our implementations of both sIB and ITCC algorithms.

5. We use the same clustering schedule S_l for every dataset. The schedule starts with splits of top-down clusters—as discussed in Section 4.3, it cannot start with a merger of document clusters otherwise the objective function (4.3) would be

0. Also, we notice that it is not beneficial to start merging document clusters before a significant number of word clusters is obtained, otherwise the objective function is still too close to 0. Thus, before doing the first iteration over documents, we perform four iterations over words, and continue with a plain (non-weighted) round-robin then.

4.6.5 Comparative results

Micro-averaged accuracy (averaged over ten independent runs,[11] whenever applicable) for the six datasets is reported in Table 4.2. It is evident that the results of our bi-modal Comraf clustering (with the underlying MDC algorithm) are significantly superior to those obtained by other methods. The only statistically insignificant improvement is recorded for MDC over sequential IB on the CALO:ACHEYER dataset; all the other gaps are statistically significant. Of particular importance is the striking 69.5% micro-averaged accuracy achieved by the bi-modal MDC on 20NG.[12] This impressive result is 12% higher than the best previously reported result on this dataset. Specifically, a micro-averaged accuracy of 57.5% on 20NG is reported for sIB in [96]. This result is obtained with only 2,000 "most discriminating" words. Also, in that work, duplicated and small documents are removed, leaving only 17,446 documents. In our implementation of sIB, our use of almost 40,000 words leads to 61% accuracy on the entire dataset of 19,997 documents. More than 5% absolute improvement is also obtained on ENRON:KITCHEN-L and CALO:MGONDEK datasets.

---

[11] Randomized algorithms, such as MDC, may obtain different results each time they are applied to the same dataset. We perform ten independent runs of each randomized algorithm on the same data, and compute the mean of the obtained results, as well as the standard error of the mean.

[12] In [10] we reported on a slightly better result of MDC on 20NG. This better performance was obtained using a cluster balancing heuristic that reduced the probability of small clusters to be further split and of large clusters to be further merged. Later we discovered that this heuristic is not uniformly effective across datasets and we therefore abandoned it.

Method	*CALO: acheyer*	*CALO: mgervasio*	*CALO: mgondek*	*Enron: kitchen-l*	*Enron: sanders-r*	*20NG*
Random	17.8 ± 0.5	18.3 ± 0.3	32.4 ± 0.1	17.9 ± 0.1	35.4 ± 0.1	6.3 ± 0.1
K-means	24.7	24.1	37.0	29.6	45.5	OOM
Agglom. IB	36.4	30.9	43.3	31.0	48.8	26.5
Sequent. IB	47.0 ± 0.5	35.1 ± 0.6	68.2 ± 1.2	34.6 ± 0.5	63.1 ± 0.6	61.0 ± 0.7
ITCC	46.1 ± 0.3	34.2 ± 0.5	63.4 ± 1.1	31.8 ± 0.2	60.2 ± 0.4	57.7 ± 0.2
LDA	44.3 ± 0.4	38.5 ± 0.4	68.0 ± 0.8	36.7 ± 0.3	63.8 ± 0.4	56.7 ± 0.6
2-modal Comraf (sequential)	$\mathbf{47.8 \pm 0.4}$	42.4 ± 0.4	$\mathbf{75.9 \pm 0.6}$	$\mathbf{42.4 \pm 0.6}$	67.4 ± 0.3	$\mathbf{69.5 \pm 0.7}$
2-modal Comraf (shuffled)	47.1 ± 0.4	$\mathbf{44.0 \pm 1.0}$	75.5 ± 0.5	41.6 ± 0.8	$\mathbf{67.6 \pm 0.3}$	67.2 ± 0.8
SVM (supervised)	65.8 ± 2.9	77.6 ± 1.0	92.6 ± 0.8	73.1 ± 1.2	87.6 ± 1.0	91.3 ± 0.3

Table 4.2. Micro-averaged accuracy ($\pm$ standard error of the mean, when applicable) on the six datasets. The SVM *supervised* classification accuracies are obtained with 4-fold cross validation. "OOM" means "out of memory": WEKA was unable to cluster 20NG, on a 4GB RAM machine. Bold numbers are the best results over all.

Surprisingly, on CALO and Enron datasets, the sequential version of MDC and its shuffled version obtain almost identical results (the difference is statistically insignificant). Note that in both versions we perform the same number of optimization steps. However, on 20NG, sequential MDC is significantly superior. This can be explained by the fact that sequential MDC is guaranteed to iterate over all the data instances, while shuffled MDC is not. On smaller datasets (CALO and Enron), the number of optimization steps is large enough to make the shuffled version iterate over (almost) every data instance. On a larger dataset (20NG), however, shuffled MDC is less likely to iterate over every data instance, and therefore is sub-optimal.

Table 4.3 shows macro-averaged accuracy results on CALO and Enron datasets. Compared with micro-averaged accuracy, macro-averaged accuracy favors smaller clusters over larger clusters. We can see in the table that Comraf's sequential MDC method is still significantly better than the baselines (here we show the results of only three most prominent baselines: sIB, ITCC and LDA). The only exception is an *insignificant* improvement MDC achieves over sIB on the ENRON:KITCHEN-L

Method	*CALO: acheyer*	*CALO: mgervasio*	*CALO: mgondek*	*Enron: kitchen-l*	*Enron: sanders-r*
Sequent. IB	57.4 ± 0.7	53.1 ± 0.7	65.9 ± 0.6	46.7 ± 0.4	69.2 ± 0.7
ITCC	57.3 ± 0.4	50.0 ± 1.2	67.0 ± 0.8	43.3 ± 0.4	65.6 ± 0.4
LDA	53.0 ± 0.6	52.2 ± 0.8	63.6 ± 0.7	39.1 ± 0.2	66.7 ± 0.2
2-modal Comraf (sequential)	$\mathbf{59.9 \pm 0.5}$	$\mathbf{58.5 \pm 0.7}$	$\mathbf{76.9 \pm 0.8}$	$\mathbf{47.0 \pm 0.7}$	$\mathbf{74.6 \pm 1.1}$

Table 4.3. Macro-averaged accuracy ($\pm$ standard error of the mean) on CALO and Enron datasets. Each number is an average over ten independent runs. Bold numbers are the best results over all.

dataset. Note that the macro-averaged accuracies shown in Table 4.3 are in most cases higher than micro-averaged accuracies (Table 4.2). This implies that small clusters constructed by the discussed clustering methods are generally cleaner than large clusters.

As shown in Table 4.4, our tri-modal Comraf (documents/words/correspondents) consistently improves the bi-modal Comraf performance on the CALO email datasets. On MGERVASIO, the addition of correspondents' modality leads to an impressive absolute improvement of 10%. On Enron email, however, tri-modal Comraf shows mixed results: a significant improvement on SANDERS-R and a drop on KITCHEN-L. A closer inspection reveals that the email correspondent input stream in Enron datasets is extremely noisy. That is, the information on the same person can be represented in dozens of different formats, delimiters between separate records are sometimes non-existent, and many email messages have very long lists of recipients (which would probably imply that email data not always strongly correlate with the recipient data).

When comparing the ICM and CWO optimization methods for Comrafs (see Section 3.3), we can see that ICM usually outperforms CWO. However, CWO is a somewhat simpler and significantly faster method (for a discussion, see Section 4.4).

Our experimentation with 4-modal Comraf (documents/words/correspondents/subject lines) on the CALO datasets shows further (insignificant) improvement over

Method	*CALO: acheyer*	*CALO: mgervasio*	*CALO: mgondek*	*Enron: kitchen-l*	*Enron: sanders-r*
2-modal Comraf—ICM	47.8 ± 0.4	42.4 ± 0.4	75.9 ± 0.6	$\mathbf{42.4 \pm 0.6}$	67.4 ± 0.3
3-modal Comraf—ICM	49.1 ± 0.4	52.4 ± 0.7	80.1 ± 0.7	40.2 ± 0.3	$\mathbf{69.0 \pm 0.4}$
3-modal Comraf—CWO	47.2 ± 0.3	48.4 ± 0.5	76.1 ± 1.2	39.5 ± 0.5	63.9 ± 0.2
4-modal Comraf—ICM	$\mathbf{50.2 \pm 0.6}$	$\mathbf{54.1 \pm 0.5}$	$\mathbf{80.9 \pm 0.5}$	34.2 ± 0.2	63.1 ± 0.4
4-modal Comraf—CWO	47.6 ± 0.2	48.6 ± 0.6	78.7 ± 1.1	38.7 ± 0.4	63.4 ± 0.4

Table 4.4. Micro-averaged accuracy ($\pm$ standard error of the mean) on CALO and Enron datasets. Each number is an average over ten independent runs. Comrafs models are 2-modal, 3-modal and 4-modal, with the *sequential* optimization applied at each node. Bold numbers are the best results over all.

the tri-modal Comraf performance. On Enron, in contrast, a significant drop can be observed. An important observation is that the subject line modality is substantially sparser than other modalities in the Enron datasets. It is evident that the addition of a sparse modality appears to be non-beneficial for multi-modal clustering. A formal method for learning a Comraf model structure is emerging, which we leave for our future work (for a discussion, see Section 4.6.6 below).

4.6.5.1 Experimentation with clustering schedule

On CALO data, we test another algorithmic setup of the bi-modal MDC, in which both words and documents are clustered *bottom-up*. The results are very similar to our original bi-modal MDC accuracies. However, this setting is not applicable to larger datasets: taking constants into account, on the 20NG dataset the bottom-up version of MDC would be 300 times slower than the original (top-down / bottom-up) MDC.

In addition, we test a *reverse* clustering schedule, where we apply bottom-up clustering to *words* and top-down clustering to *documents*. On the 20NG dataset, we perform five splitting iterations over documents (obtaining 32 clusters) and then apply the last *exhaustive* clustering iteration as explained in Section 4.6.4, reducing the number of clusters to 20. The micro-averaged clustering accuracy obtained by the reverse schedule is $69.3 \pm 0.4\%$, which is statistically indistinguishable from

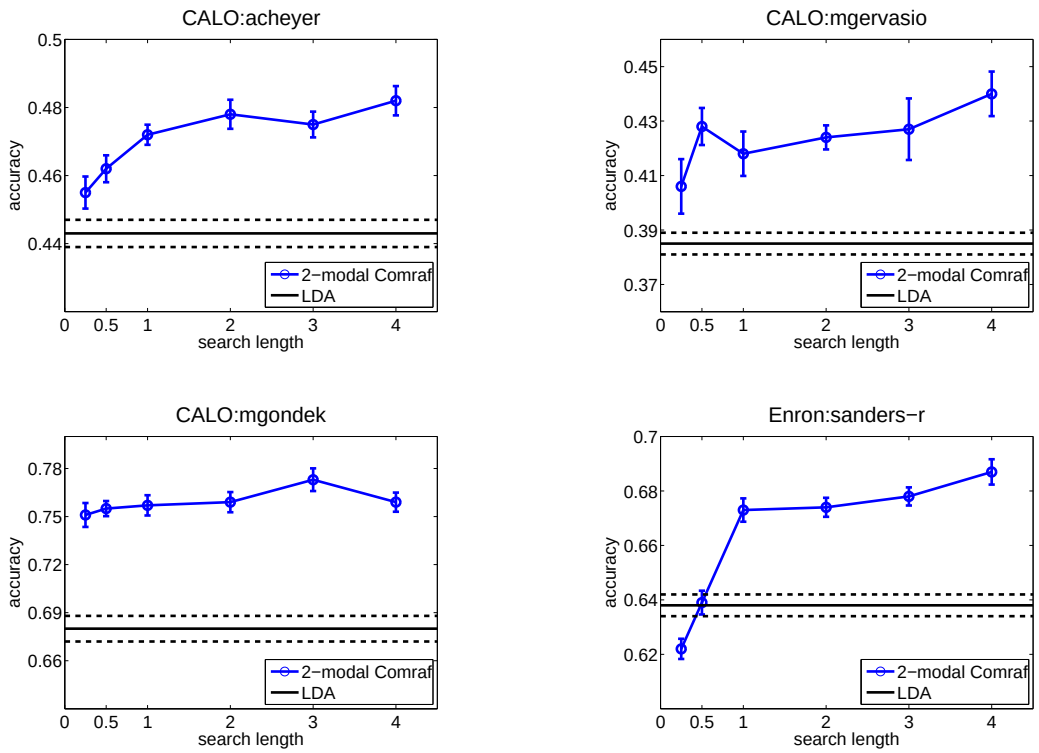


Figure 4.4. Clustering accuracies as a function of the length of local search in sequential MDC: '0.5' on the x-axis means that the MDC's optimization routine was executed over one half of the data points (chosen uniformly at random), while '3' means that the optimization routine was executed over every data point 3 times. All our results are averaged over 10 independent runs.

the the original MDC's performance. Note that the reverse scheme is significantly faster than the original MDC (8 clustering iterations vs. 21 iterations on 20NG). On email datasets, similar results are obtained in three of the five cases, whereas in two others the reverse schedule shows significantly poorer performance (3% decrease on CALO:MGERVASIO and 7% decrease on ENRON:KITCHEN-L).

4.6.5.2 Experimentation with the length of local search

Figure 4.4 presents the micro-averaged clustering accuracy of sequential MDC (in a bi-modal Comraf) as a function of the length of local search performed in the lattices of all possible word and document clusterings. Recall that in Algorithm 2

we perform a local search (i.e. an *optimization phase*), in which every data point is sequentially pulled out of its cluster and assigned into a cluster such that the objective function is maximized. In their sequential IB algorithm, Slonim et al. [96] propose to execute such an optimization routine a number of times, up to the convergence of the objective function to its local maximum. Their approach has a drawback of a potentially unlimited execution time: while it is guaranteed that the objective function occasionally converges, it is uncertain how long this can take.

In our MDC's implementation, we perform the optimization routine twice (see Section 4.3), in order to approach the local maximum, while not setting our stopping criterion at achieving the full convergence. In this section, we ask the question whether or not the length of the local search is a crucial parameter of our system. Our experiment is conducted as follows: in a bi-modal Comraf, we set the length of the optimization routine to be a function of the number of data points (words or documents). We start with the case where we explore only one quarter of the data (chosen uniformly at random), then we try one half, and then we perform from 1 to 4 full passes (over all the data points). We perform this experiment on 4 email datasets (excluding the large KITCHEN-L and 20NG collections).

As can be seen on Figure 4.4, the correlation of local search length and the clustering accuracy is quite weak, as soon as at least one pass over all the data is performed. In some cases, shorter searches are quite effective (such that the one on MGONDEK), while in the others (SANDERS-R) a significant drop is recorded. Searches longer than two data sizes are generally not beneficial: while a (rather insignificant) improvement can usually be seen, the run time increase trades off against this improvement.

Finally, let us emphasize that we approximate a *local* maximum of our objective function. Following Slonim [95], we note that obtaining a *global* maximum is very unlikely in our non-convex combinatorial optimization environment, where (in the worst case) all possible configurations should be tested in order to achieve a global

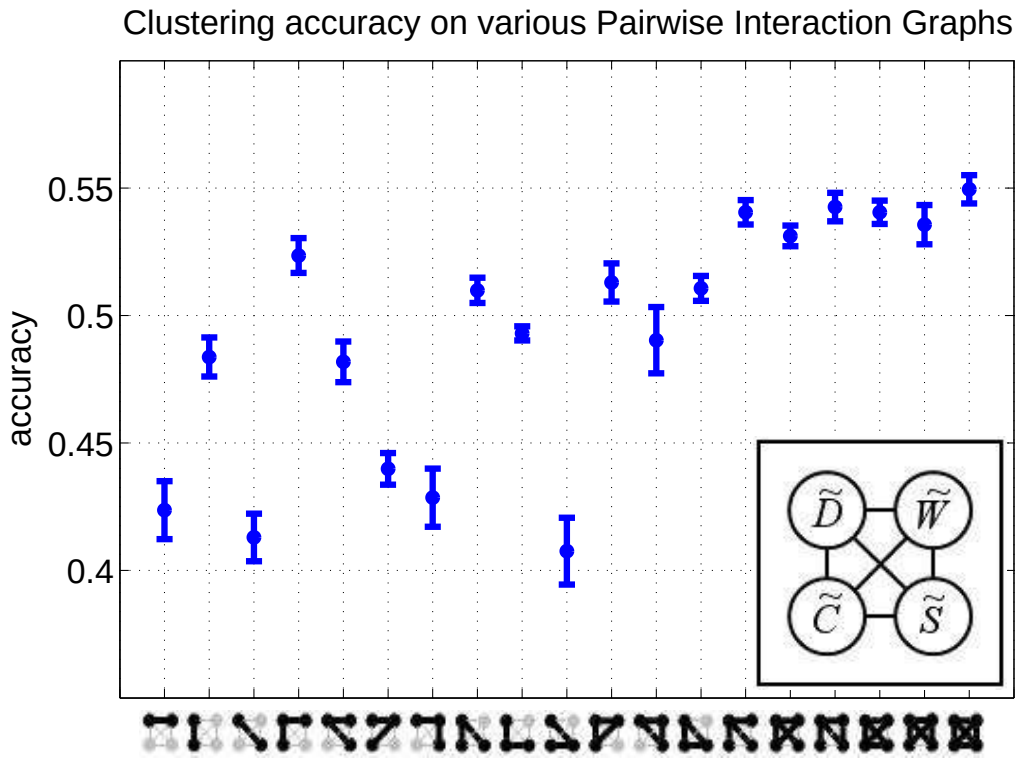


Figure 4.5. Experimenting with various Comraf graphs on MGERVASIO.

optimum. Since the number of possible configurations is astronomical even in the smallest real-world setups, approximating a global maximum is practically impossible.

4.6.6 Model analysis

As shown in Section 4.6, multi-modal clustering based on more than two entities may or may not improve performance relative to the bi-modal clustering. Given a Comraf graph (and the corresponding pairwise data), an interesting question is which of the pairwise interactions can contribute useful information to clustering the target variable.

We investigate this problem with respect to the MGERVASIO dataset. Specifically, we test all possible Comraf graphs and measure their effectiveness in accurately clustering the target variable. Figure 4.5 summarizes our findings (for better visibility, we present only the most interesting cases). As can be seen at the figure, the choice

of a Comraf graph can dramatically affect the clustering performance (within a 15% accuracy range). Also, this experiment illustrates the fact that *model learning* is feasible in Comrafs (which usually contain a small number of nodes).

Some variables can be crucial for obtaining good clustering results, while some others can be unnecessary or even harmful. For example, when substituting the *words* variable with email *subjects*, a decrease in the results can always be seen (naturally, email bodies provide more information than email subjects). In contrast, the *correspondents* variable plays a positive role in foldering email of MGERVASIO. Somewhat surprisingly, the bi-modal documents/correspondents clustering setup leads to a 6% absolute improvement over the ordinary documents/words setup. A possible explanation is that most of the folders in this dataset are created according to people groups in the email owner's social network.

Some interactions are more important than others. For example, in the documents/correspondents/titles triangle, a missing documents/correspondents interaction can cause a 10% drop in the accuracy. However, when crucial interactions are selected, adding other interactions would not significantly affect the performance, but rather will add a certain computational burden. Therefore, a desirable goal would be to select only crucial interactions, which are the ones presented in Figure 4.3. When using the CWO inference method, however, constructing the full Comraf graph is sometimes beneficial. For example, in a tri-modal setting, the addition of the correspondents/words interaction leads to a significant improvement in the (micro-averaged) document clustering accuracy on three of the five datasets: $51.1 \pm 0.4\%$ vs. $48.4 \pm 0.5\%$ on MGERVASIO, $42.2 \pm 0.4\%$ vs. $39.5 \pm 0.5\%$ on KITCHEN-L, and $68.8 \pm 0.2\%$ vs. $63.9 \pm 0.2\%$ on SANDERS-R.

4.6.7 Multi-modal clustering for social network analysis

The goal of multi-modal clustering presented in this chapter can be not only to document clustering, but also word clustering or clustering of people for the purposes of social network analysis. We apply our tri-modal Comraf to simultaneously cluster email messages, their words and correspondents, and evaluate the quality of the constructed clusters of email correspondents. To obtain the ground truth data, we asked Dr. Melinda Gervasio, the creator of the CALO:MGERVASIO email directory, to classify her 61 correspondents to semantic groups. She created four categories: SRI management, SRI CALO collaborators, non-SRI CALO participants and other SRI people not involved in the CALO project.

We evaluate two clusterings—one constrained to produce four clusters, the other to produce eight. Both produced results are highly correlated with Melinda Gervasio's labelings. In our four-cluster setup, the category of SRI management is united with the category of non-SRI people, while the category of SRI CALO collaborators (the largest one) is split to two clusters. The forth category (other SRI people) forms a single clean cluster, and the borders between the categories are successfully identified, leading to $62.3 \pm 1.4\%$ accuracy averaged over four independent runs.

In the eight-cluster result, categories of SRI management and non-SRI people are almost perfectly split to two different clusters, while other SRI employees still form one cluster, and the category of SRI CALO participants is now distributed over five clusters, one of which contains only one person who is Melinda Gervasio herself. The overall precision of the eight-cluster system is as high as $76.6 \pm 2.8\%$.

4.7 Experimentation: Web appearance disambiguation

In this section, we illustrate the application of Comraf clustering to a real-world task. In [13] we introduced *Web appearance disambiguation (WAD)* as the problem of inferring a model $\mathcal{M}$ that provides a binary function $f(d, h, \mathcal{K})$ answering whether

or not a Web page d refers to a particular person h, given the background knowledge $\mathcal{K}$. For simplicity, we consider only the case when h's name is explicitly mentioned in the page d. The problem might be easy when h's name is unique, but becomes difficult when h has a common name, such as "Tom Mitchell". Moreover, we do not know a priori whether a given person h has a unique name or not.

Note that the WAD problem is similar to, but not a special case of the problem of *person name disambiguation*. In person name disambiguation, given a collection of documents all of which mention a person name, the goal is to distinguish between documents that mention different people who have this name. In WAD, in contrast, the goal is to find a subset of the document collection in which the person of interest is mentioned, while filtering out documents that mention unrelated namesakes. To our opinion, the WAD setup is more realistic than person name disambiguation in the context of Web search, where one is usually interested in finding information about a particular person, rather than about *all* people with the same name.

As perfect background knowledge $\mathcal{K}$ is in most cases unavailable, the disambiguation decision must be made using some limited available information. Note that given no background knowledge at all, the WAD problem becomes ill-defined: in order to automatically perform the task, the person h must have an electronic representation, which cannot be constructed without any prior knowledge about the person. If $\mathcal{K}$ includes training data—pages that are related or unrelated to the person—the WAD problem is reduced to a binary classification task. In this thesis, however, we consider an unsupervised scenario.

We notice that as soon as we are given not just one, but at least two names of people who are known to belong to one social network, the WAD problem becomes well-defined and solvable. An example can be "Tom Mitchell" and "William Cohen". Since William Cohen's name appear in conjunction with Tom Mitchell's, it is apparent that we refer to William Cohen the CMU Professor, and not to the former US

Secretary of Defense. It is a rare case that two people in one social network have two namesakes in another. However, the probability of having a collision like that is not zero. We can minimize this probability by considering $N > 2$ names. To summarize, our background knowledge $\mathcal{K}$ is a list of names of people who are believed to belong to h's social network.

In a recent followup paper [113], Yang et al. claim that obtaining a few names of people who belong to the same social network is very hard. However, it is usually not the case. In many real-world cases a person name appears in a context of other people's names. These can be co-authors of a scientific paper, recipients of the same email message, attendants of a meeting or a conference etc. It is important to note that two people can belong to the same social network without even knowing each other. For instance, given two *randomly* chosen names of machine learning researchers h_1 and h_2, who may or may not be acquaintances, the disambiguation task is nevertheless likely to be solved, as Web pages referring to h_1 and h_2 are likely to be close in content, *or* close in the Web graph (the graph of hyperlinks).

In this section, we address the WAD problem as a clustering task in a Comraf. For each person h (out of a list of N people from one social network), we retrieve n_h documents that mention h's name. The resulting collection of $N \cdot n_h$ documents is clustered using the MDC method in a Comraf. For this task, our Comraf model is very simple (see Figure 4.3 left): we simultaneously cluster documents are their words.

Out of the k document clusters constructed, we choose one cluster to be the subset of documents that mention people of interest, and we delete all the other clusters that potentially mention unrelated namesakes. Our criterion for choosing the "relevant" cluster is the level of interconnectedness of documents in the cluster: for each document d_i we construct a set $\mathcal{L}_i$ of its hyperlinks (see Section 4.7.4 for the precise definition of $\mathcal{L}_i$); for each document cluster c_j we construct a set $\mathcal{CL}_j =$

$\bigcup_{(d_i,d_{i'})\in c_j}(\mathcal{L}_i \cap \mathcal{L}_{i'})$, i.e. the union of pairwise intersections of hyperlink sets; finally we cluster c with the largest set $\mathcal{CL}$. In Section 6.6.1 we propose another, possibly more adequate solution to the WAD problem.

4.7.1 Related work

Prior to our paper [13] where the WAD framework was introduced, only a handful of papers addressed the problem of person name disambiguation. Some work was done on person name disambiguation in a collection of scientific papers [51]. In the Web domain, we are aware of three related works [4, 74, 43], within the general framework of entity coreference (see, e.g. [83, 49]). Agglomerative clustering is applied in all three. Bagga and Baldwin [4] use agglomerative clustering over traditional vector space models of text windows around a personal name mention. Mann and Yarowsky [74] propose a richer document representation involving automatically extracted features. Their clustering technique however can be basically used only for separating two people with the same name. Fleischman and Hovy [43] construct a MaxEnt classifier to learn distances between documents that are then clustered. This method needs to be provided with a large training set. Since 2005, many followup papers have been published, see [76, 111, 113] and about 30 others.

4.7.2 Evaluation criterion

To define our evaluation criterion, let c be the constructed cluster of documents that we believe refer to people of our interest, and let c_r be its portion consisting of documents that actually refer to people of our interest. Let $\mathcal{D}_r$ be a portion of the dataset $\mathcal{D}$, that consists of documents referring to people of our interest. Precision of the cluster c is then defined as Prec $= |c_r|/|c|$, recall as Rec $= |c_r|/|\mathcal{D}_r|$, and F-measure, standardly, as (2 Prec Rec)/(Prec+Rec).

4.7.3 WAD dataset

For evaluation of our methods, we gathered and labeled a dataset of 1085 Web pages. In this section we describe the dataset and provide some interesting insights into its structure.

From the Feb 2, 2004 snapshot of the CALO email data (see Section 4.6.2), we selected one folder from Dr. Melinda Gervasio's email directory and extracted 12 person names that appeared in headers of messages found in this folder. The names are primarily of SRI employees and CS professors from various universities. All of the individuals are likely to be present on the Web.

In May 2004, these 12 names (in quotation marks, i.e. treated as phrases) were issued as queries to Google and for each query the first 100 pages were retrieved. We manually filtered the pages, removing pages in non-textual formats, HTTPD error pages and empty pages. We labeled the remaining pages by the occupation of the individuals whose name appeared in the query. In 10 out of 12 cases, the names were heavily ambiguous, thus pages representing 187 different people were retrieved given the 12 names of people in Melinda Gervasio's social network. In some cases, it was difficult to decide to which of the namesakes the page referred. To determine this, we often performed manual Web investigations. Table 4.5 shows some statistics of the dataset.

Finally, all the pages were cleaned of their HTML markup and scripts. All the URLs mentioned in the pages were extracted and placed at the end of each page, together with the URL of the page itself. The dataset is publicly available at `http://www.cs.umass.edu/~ronb/name_disambiguation.html`.

The most ambiguous personal name among the twelve is Tom Mitchell. Although the CMU Professor's pages are prevalent over all the others, 37 different Tom Mitchells can be distinguished in the 100 first Google hits, including professors in different fields, musicians, executive managers, an astrologist, a hacker and a rabbi. Two personal

Person name	*Position*	*Number of pages*	*Number of categories*	*Number of relevant pages*
Adam Cheyer	SRI Manager	97	2	96
William Cohen	CMU Professor	88	10	6
Steve Hardt	SRI Engineer	81	6	64
David Israel	SRI Manager	92	19	20
Leslie Pack Kaelbling	MIT Professor	89	2	88
Bill Mark	SRI Manager	94	8	11
Andrew McCallum	UMass Professor	94	16	54
Tom Mitchell	CMU Professor	92	37	15
David Mulford	Stanford Undergrad	94	13	1
Andrew Ng	Stanford Professor	87	29	32
Fernando Pereira	UPenn Professor	88	19	32
Lynn Voss	SRI Engineer	89	26	1
OVERALL:		1085	187	420

Table 4.5. Statistics of the WAD dataset. Categories are different namesakes or *other* in case if the page does not refer to any of the namesakes. The last column shows the number of pages that actually mention the person of our interest.

names out of the 12, Adam Cheyer and Leslie Pack Kaelbling, seem to be unique in the Internet. However, for either of them, one page was retrieved that did not contain any part of their names. These two pages were put into respective categories *other*. Two other people, David Mulford and Lynn Voss, seem to have very little Web presence. Only one page out of the 100 was related to any of the two. William Cohen's and David Mulford's namesakes are well known politicians: the former US Secretary of Defense William S. Cohen and the current US Ambassador to India David C. Mulford. Naturally, the distributions of Cohen's and Mulford's pages are heavily biased toward the politicians who are well represented on the Web.

An interesting phenomenon is observed for the names David Israel and Bill Mark. Many of pages that responded to these queries only accidently contain the two words adjacent to each other: Bill Mark's pages often refer to mark-ups of certain bills, or just list people's first names (e.g. "Thanks Bill, Mark!"), while some of David Israel's pages discuss Israeli history and King David. None of these pages were removed from the dataset, despite the fact that they are clearly unrelated to a particular living person.

A major challenge for the WAD system is the pages of Bill Mark and Fernando Pereira. Both researchers have namesakes who are also researchers in Computer Science: another Bill Mark is a UTexas Professor, while another Fernando Pereira is a Professor at Instituto Superior Técnico in Portugal. We term these pairs "doubles". To separate them is an especially difficult task. The opposite problem occurs with Steve Hardt: he appears on the Web not only as an SRI engineer, but also as a creator of an online game. We ourselves are actually unsure whether this is one person or two different people, but most likely this is one person.

4.7.4 Baseline: link structure model

As our baseline, we propose a one-class clustering method based on link structure analysis of Web pages (see [13] for some additional details).[13] Let graph $G_{LS} = (\mathcal{D}, \mathcal{HL})$ be the *Link Structure Graph* over a set of Web pages $\mathcal{D}$, where $\mathcal{HL}$ is a set of hyperlink connections between Web pages in $\mathcal{D}$. We say that two Web pages d_i and $d_{i'}$ have a hyperlink connection, if the sets of their hyperlinks, $\mathcal{L}_i$ and $\mathcal{L}_{i'}$, have a non-empty intersection: $\mathcal{L}_i \cap \mathcal{L}_{i'} \neq \emptyset$. Let us now define the set of hyperlinks.

For a Web page d, we define a function $URL(d)$ to be the domain of d's URL with its first directory in case if this directory exists. For example. given page d_1 with URL `http://www.cs.umass.edu/~ronb/timeline.html` the function $URL(d_1)$ will return `www.cs.umass.edu/~ronb`. Given page d_2 with URL `http://www.cs.umass.edu/` the function $URL(d_2)$ will return `www.cs.umass.edu`. By this, we capture the intuition that full URLs can be too specific, while URLs' domains can be too general.

Define a set POP to be a set of URLs with extremely popular domains, such as `www.amazon.com`. The popularity of a domain is determined using operator `:link` of Google's command line. For a Web page d, define a set $HOP(d)$ as a set of Web pages that can be reached from d while following d's hyperlinks.

---

[13] In [18] we proposed another link analysis method, based on a heuristic search in the Web graph.

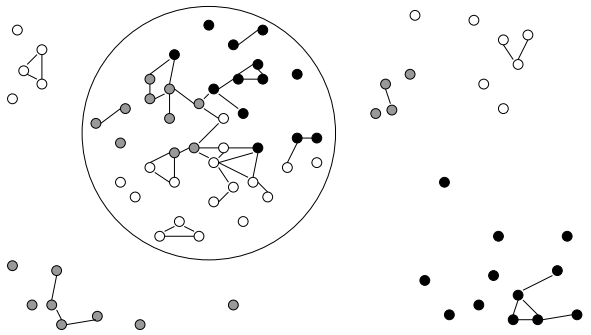


Figure 4.6. Relevant and irrelevant Web pages according to the Link Structure model. Relevant pages are within the δ-radius from the *Core Connected Component*. White, gray and black colors indicate that the pages are retrieved by three different queries.

Definition 4.7.1 *A set of hyperlinks $\mathcal{L}_i$ of a Web page d_i is defined as*

$$\mathcal{L}_i = (URL(d_i) \cup URL(HOP(d_i))) \setminus POP.$$

That is, $\mathcal{L}_i$ is d_i's URL and URLs that appear in d_i, after a generalization (using the function URL) and removal of URLs with too popular domains.

The graph G_{LS} consists of a number of connected components. Our task is to find a *Core Connected Component (CCC)* of Web pages that mention people of our interest. We naturally expect Web pages from CCC to interconnect much more than non-CCC Web pages would interconnect. Of special importance is that CCC pages referring to *different* people are likely to interconnect, while non-CCC pages referring to different people would probably not connect to each other. We could have decided that the *Maximal Connected Component (MCC)* of graph G_{LS} would be the core connected component. However, there can be a case where the MCC consists only of Web pages retrieved in response to a *single* query—this can happen when pages of one person h are heavily interconnected. If this person h appears to be an irrelevant namesake, such MCC will be totally irrelevant. Therefore, we come up with the following definition:

Definition 4.7.2 *Denote* Core Connected Component (CCC) c_0 *as the largest connected component in* G_{LS} *that consists of pages retrieved by more than one query.*

Definition 4.7.3 *The* Link Structure Model $\mathcal{M}_{LS}$ *is a pair* $(\mathcal{CC}, \delta)$*, where* $\mathcal{CC}$ *is the set of all connected components of the graph* G_{LS} *(note that* $c_0 \in \mathcal{CC}$*), and* δ *is a distance threshold.*

Our intuition is that the pages of the CCC and of a few connected components that are *close* to the CCC refer to people of our interest, while the others do not. Figure 4.6 illustrates this intuition. To find the connected components that are close to CCC, we apply the popular *cosine similarity* measure, while introducing a novel variation of the *tfidf* term weighting function, that we call *Google tfidf*:

$$Google\_tfidf(w) = \frac{tf(w)}{\log Google\_df(w)}, \tag{4.9}$$

where $Google\_df(w)$ is the *estimated total results count* of the term w if provided as a query to Google. This document frequency count appears to be the most adequate measurement of the commonness of the term w. The estimated total results counts of words in our dataset were obtained using Google API.[14]

We do not explicitly set the distance threshold δ. Instead, given that in our dataset (see Section 4.7.3) roughly one third of all Web pages refer to people of our interest, we set δ such that one third of the pages in the dataset are within the threshold.[15]

4.7.5 Comparative results

Along with our baseline method from Section 4.7.4, we implemented greedy agglomerative clustering (as applied in the related work [4, 74, 43]), based on the cosine

[14] http://www.google.com/apis/

[15] As in any unsupervised learning problem, the choice of the desired number of clusters or, dually, of the cluster sizes, is a problematic issue. We do not attempt to address this issue here; instead, we fix the size of the desired cluster based on our domain knowledge.

Method	*Precision*	*Recall*	*F-measure*
Agglomerative	61.7	53.3	57.2
Link Structure	84.2	71.8	77.5
2-modal Comraf	87.3 ± 1.7	71.3 ± 2.5	78.4 ± 0.9

Table 4.6. Web appearance disambiguation results. Bi-modal Comraf results are averaged over 4 independent runs, with the standard error of the mean reported after the $\pm$ sign.

similarity measure between clusters and the augmented *tfidf* weighting function from Equation (4.9). We did not measure interconnectedness of the clusters, we simply chose the cluster whose F-measure was the highest among all the clusters. The motivation for this choice was that we would like to show that our methods overcome the best possible results of agglomerative clustering.

The summary of the results is in Table 4.2. As it can be seen from the table, both link structure and MDC methods significantly outperform agglomerative clustering, while MDC shows slightly better performance than the link structure method. A relatively high deviation in precision and recall of the MDC algorithm is caused by the fact that it never ends up with clusters of exactly the same size. Interestingly, this deviation almost does not affect the F-measure: the precision trades off quite well against the recall.

Analyzing the results by person, we can see that for quite a few people both precision and recall are amazingly high, e.g. for David Israel, Leslie Pack Kaelbling, Andrew McCallum, and Andrew Ng. It is also notable that the only relevant page of David Mulford (the Stanford student) is found. As could be anticipated, the worst precision is for Bill Mark and and Fernando Pereira, because both of them have "doubles". However, only 9 of 23 pages that refer to Bill Mark the UTexas Professor appear in the category of relevant pages. The worst recall is for Steve Hardt and Adam Cheyer. This can be easily explained for Steve: most of his pages refer to an online game he created—relevance of these pages would be too difficult to determine.

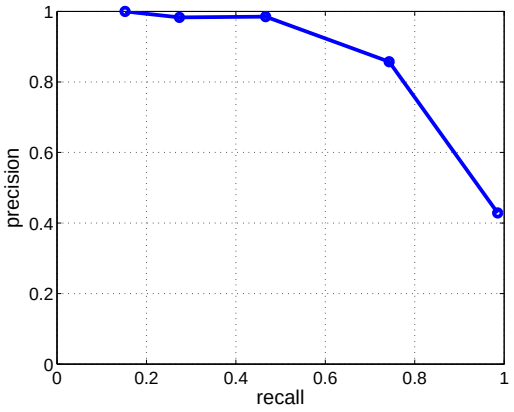


Figure 4.7. Precision/recall curve of the MDC algorithm. Points correspond to consequent iterations of the algorithm (merges of Web page clusters).

As for Adam, the low result is a bit surprising, but it still makes sense: Adam's name often appears in an industrial context, while the language of most correctly-found pages is purely academic—many of Adam's pages fall too far from the central cluster. Unfortunately, the single relevant page about Lynn Voss was not found, probably for the same reason: it uses an industrial vocabulary.

The problem of disambiguating the "doubles"—the two Bill Marks and two Fernando Pereiras who all work in Computer Science—can in fact be handled within the Comraf framework. At some intermediate stages during the course of the MDC algorithm the most interconnected cluster is relatively small but extremely clean. Figure 4.7 shows the precision/recall curve for one run of the MDC algorithm. It can be seen in the graph that when the recall of the relevant cluster is around 45% (there are five clusters overall), the precision is very high (above 98%).[16] This cluster contains two pages of Bill Mark the SRI Manager and none of the pages of Bill Mark the UTexas Professor; it also contains 15 pages of Fernando Pereira the UPenn Professor and only one page of Fernando Pereira the Professor of Instituto Superior Técnico.

[16]Notably, when the recall is around 15% (17 clusters overall), we obtain 100% precision.

4.8 Experimentation: clustering scientific papers

In this section, we test the Comraf model on another type of data: a collection of scientific papers. The goal of this experiment is as follows. From the model analysis in Section 4.6.6 we can infer that in many cases tree-structured models perform comparably to loopy models. The question that we ask in this section is whether there exists a case where a loopy model performs significantly better than a corresponding tree-structured one. In Section 4.6.6 we provided an evidence for the advantage of loopy models, where the underlying inference method is CWO. In this section, we show the advantage of loopy models when the underlying algorithm is MDC.

The evidence given in this section has an important implication: as discussed in Section 4.1, if a Comraf graph is tree-structured, then our objective function (4.3) is a factorized version of Multi-Information (4.2). That is, Comraf models of a tree structure are equivalent in their modeling power to the hard version of multivariate Information Bottleneck (mIB) [97] where the Multi-Information is used. Loopy Comraf models, however, are not equivalent to mIB. As we show below, in some cases loopy Comraf models obtain higher results than corresponding tree-structure ones, which means that in those cases the Comraf framework is preferable over mIB.

Our dataset was created by David Mimno from a repository of scientific papers collected for the REXA project.[17] The dataset consists of 4887 conference papers, published at ten venues: ACL, ICCV, ICRA, IJCAI, KDD, NIPS, SIGIR, SIGMOD, STOC, and WWW. In our data, a significant number of papers belong to each of the ten venues: between 224 and 933 papers. From the paper titles, we extracted 1436 words, each of which appeared in at least 2 titles. We also extracted 9841 words from paper *abstracts*, each of which appeared in at least 2 abstracts. Citations in the papers were automatically co-referenced using the REXA software system. Again, as

[17] `http://rexa.info/`

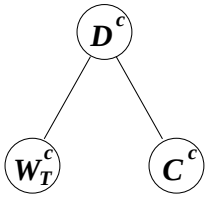
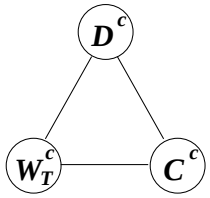
(a)	(b)	(c)	(d)	(e)
				
$38.8 \pm 0.5\%$	$40.7 \pm 0.7\%$	$55.0 \pm 0.7\%$	$61.4 \pm 0.6\%$	$63.9 \pm 0.7\%$

Table 4.7. Clustering scientific papers. Comraf models for clustering: (a) documents and title words; (b) documents and citations; (c) documents, title words and citations in a tree-structured model; (d) documents, title words and citations in a loopy model; (e) documents and abstract words. The bottom line is the micro-averaged clustering accuracy obtained by those models.

in the case of words, we removed citations that appeared in only one paper, resulting in 11,143 distinct citations.

Our goal is to cluster documents by their venues. We consider five Comraf models presented in Table 4.7. First, we test two bi-modal Comrafs, where documents D are clustered with their title words W_T and with their citations C. Second, we experiment with two tri-modal Comrafs (tree-structured and loopy), where D, W_T and C are clustered simultaneously. Finally, we present a bi-modal Comraf for clustering documents D and *abstract* words W_A.

Our underlying clustering method is a sequential MDC (see Section 4.3). We cluster words and citations top-down, while clustering documents bottom-up. Our clustering schedule is a plain round-robin. The algorithm stops when the desired number of document clusters (i.e. 10, which equals the number of venues) is reached.

The micro-averaged clustering accuracy results are presented in the bottom line of Table 4.7. As can be seen, neither title words nor citations are good document representations. Only about 40% accuracy is obtained in a bi-modal Comraf using either title words or citations. However, the result of a tree-structured tri-modal Comraf (where documents are clustered simultaneously with title words *and* citations) is no-

tably 15% higher. Of particular importance is that adding the title words/citations interaction improves this result by another 6% accuracy (on the absolute scale). Since a loopy Comraf model like this is not equivalent to any model in the multivariate IB framework, this result demonstrates the superiority of the Comraf modeling framework over multivariate IB.[18]

The Comraf model that achieves the best performance on the scientific paper clustering task is the one where papers are represented over words in their *abstracts* (see the last column in Table 4.7). Adding another modality to this setup (such as citations) causes a significant drop in the clustering accuracy. This result implies that the abstract words' modality is dense enough and much less noisy than title words or citations. Whenever the abstracts' data is available, using it would be preferable over using the other modalities. However, if the abstracts' data is unavailable, we show that using a combination of two noisy modalities such as title words and citations leads to almost the same result.

4.9 Experimentation: clustering documents by genre

So far, we have considered clustering documents *by their topic*. Topics, however, are not the only way in which someone might want to select groups of documents. Aspects such as genre, opinion, authorship, style, author's mood, and so on are interesting dimensions along which clustering results might break. In this section, we focus on techniques appropriate for such non-topical clustering, with a particular emphasis on genre. Although the field of non-topical (supervised) classification is well explored in the literature (a lot of work was done on classification by genre

[18]Note that this is not the only advantage of Comrafs over multivariate IB models. The Comraf framework is substantially simpler and more intuitive (e.g. the multivariate IB introduces in-space and out-space concepts which are unnecessary in Comrafs). In contrast, Comraf inference algorithms are more complex and effective than those proposed for the multivariate IB (see Section 4.6.5 for a discussion).

[59, 60, 42, 71, 92], by text authorship [77, 3], by writer's gender [63], tone [107, 85] and mood [82], as well as by familiarity with the topic of the discussion [64]), we believe that the problem of genre *clustering* had not been comprehensively studied before we approached it in [9, 15].

To apply the Comraf framework to the task of clustering by genre, we first have to decide about modalities that would best match the task. Documents are labeled with genres on the basis of external criteria such as intended audience, purpose and activity type [70]. The notion of genre can be described in terms of the syntax/semantics duality of text: documents of different genres use different syntactic constructions and/or different vocabulary. It is not obvious whether syntactic or semantic features play a major role in clustering documents by genre. We propose to take advantage of both. Besides the document modality, we consider two other modalities: words (that correspond to documents' vocabularies) and Part-Of-Speech (POS) n-grams (that correspond to the syntactic structure of text). POS n-grams are extracted from sentences in an incremental manner: the first n-gram starts with the POS tag of the first word in the sentence, the second one starts with the tag of the second word etc. For example, out of the sentence

```
<PNP>It <VBZ>'s <AT0>a <AJ0>real <NN1>holiday <PUN>.
```

we extract four trigrams:

```
PNP_VBZ_AT0, VBZ_AT0_AJ0, AT0_AJ0_NN1, AJ0_NN1_PUN.
```

Given a document collection, let D be a random variable over its documents, W be a random variable over its words, and S be a random variable over the POS n-grams of its words. We apply a multi-modal Comraf model (Section 3) for constructing a clustering d^{c*} of documents, a clustering w^{c*} of words and/or a clustering s^{c*} of POS n-grams, by maximizing the objective derived from Equation (4.3). In this section, we consider four Comraf models for clustering by genre:

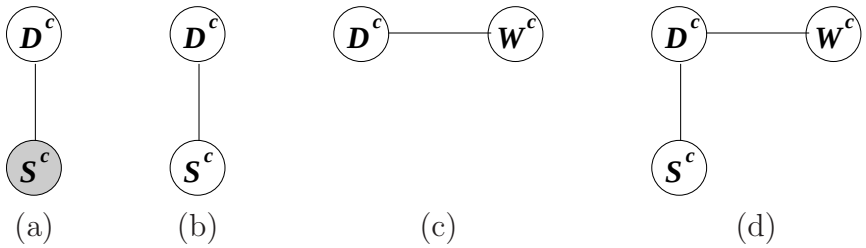


Figure 4.8. Comraf graphs for: (a) 1-way document clustering with POS unigrams as an *observed* r.v. (shaded node); (b) 2-way clustering of documents and POS bigrams (same as for POS 3-grams or 4-grams); (c) 2-way clustering with BOW; (d) 3-way clustering with POS bigrams and BOW.

1. **POS unigrams:** Since the number of POS tags in any tagging system is relatively small, it makes no sense to cluster POS unigrams. Therefore, we apply a 1-way model for clustering documents using the Comraf graph shown in Figure 4.8(a). The objective function from Equation (4.3) in this simple case has the form of $I(\tilde{D}; S)$.

2. **POS n-grams, where $\mathbf{n > 1}$**. The number of unique POS n-grams of order higher than 1 is exponential in n, so clustering them would be necessary. We perform a 2-way clustering with the Comraf graph from Figure 4.8(b) and the objective $I(\tilde{D}; \tilde{S})$.

3. **Bag-Of-Words:** The number of unique words in our dataset is comparable with the number of POS trigrams, so in analogy to the previous model, we perform a 2-way clustering with the Comraf graph of Figure 4.8(c) and the objective $I(\tilde{D}; \tilde{W})$.

4. **BOW+POS hybrid:** We combine contextual information of BOW and stylistic information of POS n-grams into a 3-way clustering model, where we simultaneously cluster documents, words and bigrams of POS tags. Over the Comraf graph of Figure 4.8(d), we maximize the sum $I(\tilde{D}; \tilde{S}) + I(\tilde{D}; \tilde{W})$.

Doc representation	k-means	LDA	Comraf
Bag-Of-Words	9.1%	$55.4 \pm 0.1\%$	$55.7 \pm 0.2\%$
POS bigrams	23.2%	$44.7 \pm 0.2\%$	$51.0 \pm 0.2\%$
BOW + POS bigr	n/a	n/a	$\mathbf{58.5 \pm 0.6}\%$

Table 4.8. Clustering by genre. Micro-averaged clustering accuracy on the BNC corpus, averaged over four independent runs. Standard error of the mean is shown after the ± sign. Comraf results with other POS tuples, besides bigrams, are in Figure 4.9(left). The BOW+POS hybrid setup is only applicable in Comrafs.

4.9.1 Dataset

We evaluate our models on the British National Corpus (BNC) [24]. We employ David Lee's ontology of BNC genres [70] with 46 genres covering most aspects of modern literature such as *fiction prose, biography, technical report, news script* and others. To perform fair evaluation using micro-averaged clustering accuracy (Section 4.6.1), we choose 21 largest categories, for each of which we uniformly at random choose 32 documents, so our resulting dataset consists of 672 documents. The BNC texts are formatted using the SGML markup language. We remove all markup, lowercase the text, and delete stopwords and low-frequency words. All words in the BNC corpus are semi-manually tagged using 91 POS tags, four of which refer to punctuation. The resulting dataset has 63,634 unique words; and 5864 POS bigrams. Since the overall number of unique POS trigrams and fourgrams is prohibitively large, we apply more aggressive term filtering: we consider trigrams that appear in at least 10 documents (44,499 trigrams overall) and fourgrams that appear in between 10 and 99 documents (114,476 fourgrams).

4.9.2 Comparative results

We compare the results of Comraf models (with the MDC optimization algorithm) with the results of k-means (Weka implementation), as well as of Latent Dirichlet Allocation (LDA). As in Section 4.6, we use Xuerui Wang's LDA implementation [78]

that performs Gibbs sampling with 10000 sampling iterations. Table 4.8 summarizes our results.

As we can see from Table 4.8, MDC achieves more than 50% accuracy with both BOW and POS bigram document representations. Note that a random assignment of documents into clusters would lead to about 5% accuracy on our dataset, so above 50% accuracy is an impressive result for a purely unsupervised method on a large, well-balanced dataset. The LDA+BOW system obtains exactly the same accuracy as MDC+BOW does. However, LDA demonstrates strictly inferior performance (lower than MDC by 6% absolute) on the POS bigram representation. We can also see that MDC+BOW significantly outperforms MDC+POS (by more than 4% absolute). This observation may imply that contextual features (such as words) play a more important role for genre classification than stylistic features (such as POS n-grams).

To give some insight on the differences in MDC performance on BOW and POS bigrams, we present Table 4.9 that shows the distribution of documents of each genre over the generated clusters. For each genre we show a list of sizes (in number of documents) of this genre's representation in various clusters. We sort this list by the size of the representation from the largest to the smallest. An asterisk after the number of documents means that this genre is dominant in the corresponding cluster. A heavy tailed distribution (such as the one of `W_non_ac_soc_science`) implies that the genre is spread over many clusters which is clearly a failure. In contrast, a peaked distribution (e.g., of `W_non_ac_tech_engin`) with an asterisk on its largest component means that the genre was successfully identified.

As we can see from the table, MDC performs similarly on BOW and POS bigrams. However, some significant differences can be found. For example, genres `W_biography`, `W_commerce` and `W_institut_doc` are successfully identified by MDC+BOW but not by MDC+POS, while MDC+POS better recognizes `W_newsp_brdsht_nat_social` and

Genre	MDC with POS bigrams	MDC with BOW	LDA with BOW	MDC with BOW and POS bigrams
W_ac_humanities_arts	9* 6* 6 4 2 2 1 1 1	9* 6* 5 5 3 2 1 1	7 6 5 5 4 4 1	9 6* 5 5 4 1 1 1
W_ac_nat_science	23* 4 2 2 1	24* 6 1 1	12* 11* 9	27* 4 1
W_ac_polit_law_edu	14* 8 5 2 1 1 1	20* 5 2 2 1 1 1	19* 7 4 2	17 6 4 2 1 1 1
W_ac_soc_science	11* 9* 6 5 1	12* 10* 7 1 1 1	12* 9* 8* 1 1 1	16* 7 6 3
W_advert	14* 11 3 2 2	18* 3 3 2 2 1 1 1 1	22* 2 2 2 1 1 1 1	23* 2 1 1 1 1 1 1 1
W_biography	15* 8 6 1 1 1	12 7 6 3 2 1 1	16* 6 4 2 2 1 1	16* 6 6* 2 1 1
W_commerce	10* 5 5 4 2 2 1 1 1 1	13 10 6 1 1 1	16 5 4 2 2 1 1 1	9* 9 4 3 3 2 1 1
W_fict_prose	22* 7 3	25* 6 1	30* 2	24* 6 2
W_institut_doc	15* 6 5 5 1	18 6 4 1 1 1 1	17* 7 4 2 2	14 11* 3 1 1 1 1
W_newsp_brdsht_nat_arts	25* 5 1 1	28* 1 1 1 1	30* 2	27* 2 2 1
W_newsp_brdsht_nat_commerce	26* 2 1 1 1 1	32*	28* 2 1 1	31* 1
W_newsp_brdsht_nat_report	32*	32*	30* 2	32*
W_newsp_brdsht_nat_social	9 7 4 4 2 2 1 1 1 1	11* 6 4 3 2 2 1 1 1 1	10 7 6 3 2 1 1 1 1	14 6 3 2 2 2 1 1 1
W_news_script	32*	32*	31* 1	32*
W_non_ac_humanities_arts	11* 8 3 2 2 2 1 1 1 1	9* 6 5 3 3 2 2 2	10* 7 5 3 2 2 2 1	14* 5 3 3 3 2 1 1 1 1 1
W_non_ac_nat_science	14* 5* 3 2 2 2 1 1 1 1	18* 11 2 1	11* 9 7 2 2 1	29* 1 1 1
W_non_ac_polit_law_edu	11* 4 4 3 3 2 2 1 1 1	11 10* 5 3 2 1	10* 10* 3 3 2 2 1 1	10* 6 5 5 2 2 1 1
W_non_ac_soc_science	5 5 4 3 3 2 2 2 2 1 1 1 1	7 5 4 4 3 2 2 2 2 1	7 6 5 5 3 2 1 1 1 1	5 5 4 3 3 3 2 2 2 1 1 1
W_non_ac_tech_engin	32*	32*	32*	32*
W_pop_lore	11 6 6 5 4	10* 9* 4 4 2 2 1	12 8 6 3 2 1	16* 8 3 2 2 1
W_religion	11* 5 4 4 2 2 1 1 1 1	18* 6 2 1 1 1 1 1 1	20* 6* 2 1 1 1 1	18* 6* 3 1 1 1 1 1

Table 4.9. Performance of various methods per genre. For each genre we show a list of sizes (in number of documents) of this genre's representation in various clusters. We sort this list by the size of the representation from the largest to the smallest. An asterisk after the number of documents means that this genre is dominant in the corresponding cluster.

`W_pop_lore`. A 3-way MDC with both BOW and POS that would take advantage of the both approaches may have a good chance to show even better results.

Indeed, we obtain a strong result with the 3-way MDC: 58.5% accuracy. The last column of Table 4.9 presents the analysis of this result by genre. For many genres (such as `W_non_ac_nat_science`) we enlarge their dominant representations. We also manage to identify four of the five genres that were in disagreement between BOW and POS models (as discussed above). However, we no longer recognize `W_ac_polit_law_edu`, which indicates that the results might potentially be improved even more.

One could argue that the direct comparison of results obtained by the BOW and POS bigram models is actually unfair because the number of BOW features is one

order greater than the number of POS bigrams, so that the BOW model naturally outperforms the POS bigram model because it just contains more information. However, this argument cannot be empirically proved. We test MDC with POS trigrams and fourgrams, as well as with POS unigrams, and show that while the MDC performance with unigrams is significantly lower than with bigrams, trigrams and fourgrams do not significantly improve the results of bigrams. In Figure 4.9(a) we can see that when moving from bigrams to trigrams and fourgrams, the graph has a slightly positive slope, however the results become noisier (the standard error becomes higher) which diminishes statistical significance of the improvement. A conclusion that can be made from this experiment is that the Bag-Of-POS-bigrams model appears to be rich enough to capture genres of documents.

A common belief is that stopwords and other high frequency words can be good features for discrimination of documents by genre (see, e.g. [100]). It is interesting to see whether we can support this hypothesis with empirical evidence. To show this, we conduct the following experiment. We put various thresholds on the low frequency words in the BOW representation of the documents. We consider four such thresholds: our initial setup, when we filter out words that appear in less than 3 documents, as well as three new ones: 10, 20 and 50 documents. Note that the new thresholds and especially the most restrictive one (50) leave us with highly frequent words only: since our dataset consists of 672 documents, filtering out words that appear in less than 50 documents causes removal of over 93% of unique words from the dataset. We run MDC on the four representations. Figure 4.9(b) shows results of this experiment. We can see that although the graph has a negative slope, the decrease in the results is insignificant. With 7% of words from the original dataset the MDC system obtains only 2.5% lower accuracy than with 38% of words (where the rest appear in only one or two documents and can be removed with high confidence). This result confirms that high frequency words are important for genre classification.

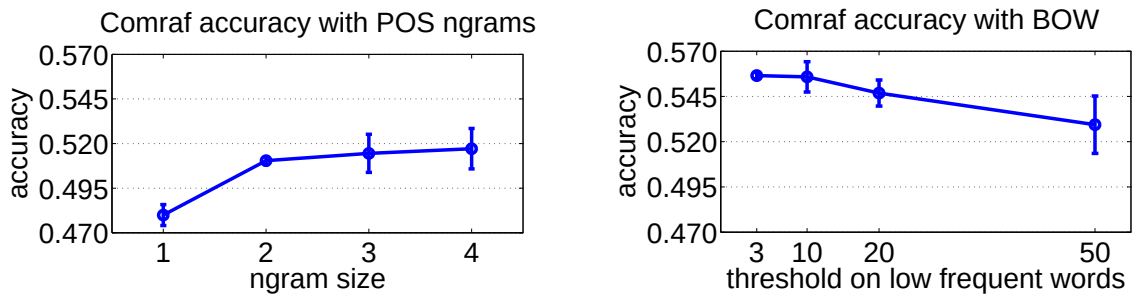


Figure 4.9. Clustering by genre. Micro-averaged clustering accuracy of Comraf models as a function of: (left) size of POS n-gram (1-grams, 2-grams, 3-grams and 4-grams); (right) threshold on low frequency words—a point i on the X axis means that in this experiment words that appear in less than i documents are removed.

4.10 Summary

In this chapter, we have proposed the objective function for Comraf clustering and presented two inference methods in Comrafs: a global optimization method (MDC) and a local optimization method (CWO). Comraf models have been successfully applied to document clustering. We have tested Comrafs on a variety of clustering tasks:

- On email clustering (see Section 4.6), a bi-modal Comraf is compared with three state-of-the-art clustering methods. It outperforms a (uni-modal) sequential IB method because it benefits from the multi-modal nature of the data. The advantage of the bi-modal Comraf over the bi-modal (flat) ITCC method suggests that the power of our inference algorithm stems from a better exploitation of the clustering hierarchy. The Comraf model demonstrates superior performance in comparison to LDA—a generative graphical model—because Comrafs provide a more flexible modeling environment (see Section 3.4). Also, we provide evidence that extending a bi-modal Comraf to 3-modal and 4-modal setups can further improve document clustering results.

- In Section 4.7 we apply a Comraf model to the real-world task of Web appearance disambiguation (WAD) of people names. We show that it slightly outperforms a strong baseline method that employs link structure analysis of Web pages. In [13] we show that the best results are achieved when using a hybrid of the Comraf clustering and link structure analysis. In Section 6.6.1 we will show a better method for WAD that is based on one-class clustering of documents.

- In Section 4.8 we address the question of whether Comrafs have more modeling power than the previously proposed multivariate IB framework [97]. We provide an example for strict superiority of Comraf models.

- Finally, in Section 4.9 we apply Comrafs to a non-topical document clustering task. We focus on clustering by genre where a lexical modality (e.g. words) are used in conjunction with a stylistic modality (POS n-grams). Similar Comraf models can be applied to document clustering according to other non-topical criteria, such as readability. In Section 5.3 we will extend the non-topical clustering model to a semi-supervised case and test it on clustering by author's sentiment.

Being a valid graphical model, a Comraf takes advantage of modeling abilities of existing graphical models. For example, we can introduce an *observed state* through which some prior knowledge can be represented. The next chapter describes a resulting model.

CHAPTER 5

COMRAFS FOR SEMI-SUPERVISED LEARNING

The Comraf model is a convenient framework for performing semi-supervised clustering [16, 17] (see Section 5.1), transfer learning [17] (see Section 5.2), and interactive clustering [15] (see Section 5.3). Prior to presenting details of particular Comrafs, let us define the concepts of hidden and observed states in the Comraf model. A combinatorial r.v. is *hidden* if it can take any value from its event space. A combinatorial r.v. is *observed* if its value is preset and fixed.

5.1 Semi-supervised clustering with Comrafs

Semi-supervised clustering is a clustering task that takes advantage of labeled examples. Usually, semi-supervised clustering is performed when the number of available labeled examples is not sufficient to construct a good classifier (e.g., the constructed classifier would overfit), or when the the labeled data is noisy or skewed to a few classes. Assuming that *most* of the labeled data is accurate, our goal is to incorporate it into the (unsupervised) Comraf model.

In this thesis, we consider only a uni-labeled case where each labeled data point $x_i|_{i=1}^n$ belongs to one ground truth category $t_j|_{j=1}^k$. We propose an *intrinsic* Comraf approach for incorporating labeled data into clustering (by introducing observed nodes to a Comraf graph), and compare it with existing *seeding* [7] and *constrained optimization* [110] schema.

Intrinsic approach. Comrafs offer an elegant method for incorporating labeled data, which does not require any significant changes in the clustering model proposed

in Chapter 4. First, note that labels define a natural partitioning of the labeled data: for each label t_j let $\tilde{x}_{0j}$ be a subset of $\mathcal{X}$ labeled with t_j, i.e. $\tilde{x}_{0j} = \{x_i | t_i = t_j\}$. We now define a r.v. $\tilde{X}_0$ over the partitioning $x_0^c = \{\tilde{x}_{0j} | j = 1, \ldots, k\}$, and we also define a combinatorial r.v. X_0^c over all the possible partitionings of the set $\mathcal{X}$. Since the partitioning $\tilde{x}_0^c$ is *given* to us, the variable X_0^c is *observed*, with x_0^c being its fixed value. Observed combinatorial random variables appear shaded on a Comraf graph. The objective function from Equation (4.4) and the MPE inference procedure remain unchanged (with the only difference being that there is no need for optimizing the observed nodes): at each ICM iteration the current node is optimized with respect to the *fixed* values of its neighbors, whereas the values of the observed nodes are fixed by definition.

Constrained optimization. Wagstaff and Cardie [110] perform semi-supervised clustering with two types of boolean constraints. The *must-link* constraint ml equals 1 if two equally labeled data points are assigned into different clusters; the *cannot-link* constraint cl equals 1 if two differently labeled data points are assigned into the same cluster. A clustering objective function incorporates the constraints, e.g. in Comrafs (Equation (4.4)) for each combinatorial r.v. X_i^c it is:

$$x_i^{c*} = \arg\max_{x_i^c} \sum_{i':\ (X_i^c, X_{i'}^c) \in \mathbf{E}} I(\tilde{X}_i; \tilde{X}_{i'}) - \sum_{i'} w_{i,i'}\ ml_{i,i'} - \sum_{i'} w_{i,i'}\ cl_{i,i'},$$

where the weights $w_{i,i'}$ are set at $+\infty$, which means that all constraints must be satisfied. Note that in the general case we are free to choose any non-negative weights. In order to fairly compare two semi-supervised methods, for both of them we must use the same underlying clustering algorithm. We use the MDC algorithm (see Section 4.3) in both cases.

Seeding [7] is a method of constructing the *initial* clustering of both labeled and unlabeled data points, for which the must-link and cannot-link constraints are

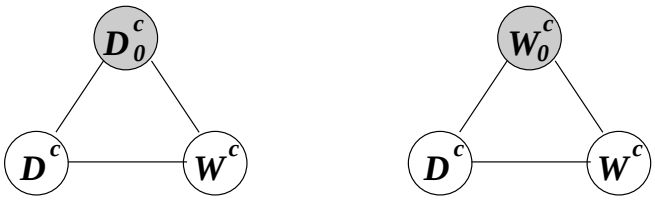


Figure 5.1. Comraf graphs for: (left) semi-supervised clustering; (right) clustering with transfer learning.

satisfied. This method is applied to Comraf clustering by adapting the initialization step of the MDC algorithm (see Algorithm 2): for each node X_i^c we select an initial point in lattice L_i that satisfies the seeding constraints. Note that, in contrast to the constrained optimization scheme described above, in the seeding scheme the clustering objective function remains unchanged, such that the seeding constraints may no longer be satisfied during the course of the MDC algorithm.

5.1.1 Experimentation

Figure 5.1(left) shows a Comraf graph for the intrinsic scheme of semi-supervised clustering. Together with a combinatorial r.v. D^c over document clusterings and a combinatorial r.v. W^c over word clusterings, we introduce an observed node D_0^c, whose value d_0^c is a given partitioning of labeled documents. With a random variable $\tilde{D}_0$ defined over the clusters in d_0^c, our objective derived from Equation (4.3) is:

$$(d^{c*}, w^{c*}) = \arg\max_{d^c, w^c} I(\tilde{D}; \tilde{W}) + I(\tilde{D}; \tilde{D}_0) + I(\tilde{W}; \tilde{D}_0).$$

As mentioned above, the ICM optimization procedure remains unchanged and iterates over nodes D^c and W^c only (the observed node D_0^c shall not be optimized).

It is interesting to note that the seeding approach to the semi-supervised clustering appears to be useless when applied to Comrafs. Despite the sophisticated initialization, the optimization procedure leads to the same local maxima of the objective, as

in the case of trivial initialization. When applied to document clustering, the MDC algorithm with seeding and without seeding demonstrates the same performance. Below we compare our intrinsic Comraf scheme with the constrained optimization only, which is naturally robust to the choice of a particular optimization method.

On the CALO and Enron datasets described in Section 4.6.2, we conduct the following experiment: for each dataset, we uniformly at random select 10%, 20%, or 30% of the data and refer to it as labeled examples while the rest of the data is considered unlabeled. We apply both intrinsic and constrained methods on these three setups and plot the micro-averaged accuracy (calculated on unlabeled data only) vs. the percentage of labeled data used. The results (in terms of clustering accuracy as defined in Section 4.6.1) are shown in Figure 5.2. As we can see from the figure, both methods unsurprisingly improve the unsupervised results, while the intrinsic Comraf method usually outperforms the constrained method.

On the 20NG dataset, we select 10% of data to be labeled. The constrained method obtains $74.8 \pm 0.6\%$ accuracy, while the intrinsic method obtains $78.9 \pm 0.8\%$ accuracy (over 5% and 9% absolute improvement to the unsupervised result, respectively).

The intrinsic scheme is resistant to noise. To show this, we conduct the following experiment: on CALO datasets with the 20%/80% labeled/unlabeled split, we arbitrarily corrupt labels of 10%, 20% and 30% of the labeled data. Figure 5.2(f) shows that clustering accuracy remains almost unchanged for all three datasets.

5.2 Transfer learning with Comrafs

Transfer learning is the problem of applying the knowledge learned in one task to effectively solve another learning task. In this section, we represent the acquired knowledge as a partitioning $\tilde{y}_0^c$ pre-built for data $\mathcal{Y}$ that can be used for constructing a partitioning $\tilde{x}^c$ of data $\mathcal{X}$. We note that the intrinsic scheme for semi-supervised

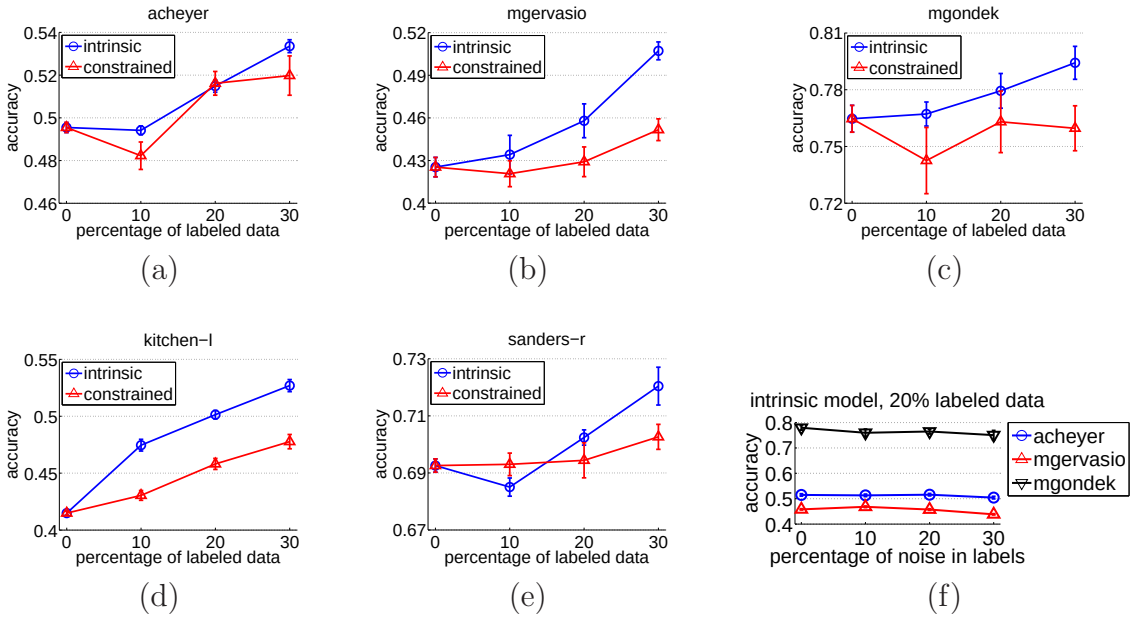


Figure 5.2. Plots (a)-(e): comparing accuracies of the semi-supervised Comraf and the constrained optimization method on five email datasets. Plot (f): the semi-supervised Comraf's resistance to noise in labeled data.

clustering presented in Section 5.1 above allows us to directly use labeled data not only from $\mathcal{X}$ but also from *another* collection $\mathcal{Y}$. Thus, in analogy to the semi-supervised case, we introduce an observed combinatorial r.v. $\tilde{Y}_0^c$ with a fixed value $\tilde{y}_0^c$. During the inference process, we construct $\tilde{x}^{c*}$ that maximizes agreement (in terms of mutual information) with the labeled data $\tilde{y}_0^c$, while applying the same objective function as in Equation (4.3) and the same ICM optimization procedure.

We set up a transfer learning experiment as follows. We notice that in two of the CALO datasets (ACHEYER and MGERVASIO) similar topics are discussed. Our hypothesis is that *known* categories of one dataset can improve the clustering results on another dataset. To test this hypothesis, we first consider one dataset to be labeled, while the other one is unlabeled, and then vice versa. However, since the two datasets do not consist of the *same* documents, we decide to use *word* clusters of the labeled dataset. We first cluster words distributed over categories of the labeled

dataset, as described in [11]. Then we introduce the constructed *word* clustering as an observed node W_0^c into the Comraf graph (see Figure 5.1 right) and perform the ICM inference. Using this scheme we improve the micro-averaged clustering accuracy on MGERVASIO by 3% absolute over unsupervised clustering, but we do not see any change in accuracy on the ACHEYER dataset. This preliminary result demonstrates the usability of Comrafs for transfer learning; other types of Comraf models for transfer learning are emerging.

5.3 Interactive clustering with Comrafs

In *interactive clustering* of text collections, the user is actively involved in the process of clustering documents, their features, or both (see, e.g., [53]). Being thus provided with some level of supervision, the interactive clustering scheme can be viewed as an instance of semi-supervised learning. In Sections 5.1 and 5.2 above, we have shown how to incorporate prior knowledge into the Comraf graph G, while using the same objective or inference algorithm as in unsupervised clustering. Here, we incorporate prior knowledge into our inference *algorithm*, preserving the Comraf graph and the objective (4.3) of the unsupervised case.

In [15], we proposed interactive clustering as a unified framework for clustering document collections according to nearly any criterion of the users choice: documents' style, readability, credibility; authors' age, mood, sentiment, familiarity with the topic etc. (for the beginning of the discussion and an example of clustering by genre, see Section 4.9). The user is first asked to choose modalities (or types of features) suitable for clustering by the desired criterion. In clustering by genres, for example, documents may be represented over sequences of Part-Of-Speech (POS) tags, punctuation marks, stopwords, as well as over general words as captured in the standard BOW representation. The user is next asked to provide a few examples of features (*seed features*) of the chosen types, if such examples are intuitive and can

be obtained without much effort—e.g., when clustering by authors mood, words like 'angry', 'happy', 'upset' might be easily suggested.

The clustering system then represents documents based on the users choice and applies a Comraf clustering method. When seed features are provided, the system iteratively clusters documents represented over the chosen features and then enriches feature sets with other useful features. The user can choose to intervene (or not) after each iteration, in order to fix possible mistakes made by the system on the feature level (no document labeling is required).

In this section, we illustrate the effectiveness of our approach on clustering by author's sentiment [107]. In clustering by sentiment, data categories correspond to different levels of the authors' attitude to the discussed topic (e.g. liked/disliked, satisfied/unsatisfied etc.) The categories can be finer grained (*strongly* liked / *somewhat* liked etc.)—as long as it is possible to distinguish between two adjacent categories. We perform interactive clustering within a bi-modal Comraf framework, where documents and words are clustered simultaneously. The user is involved in the process of clustering words (it is easier for the user to be involved in clustering words than in clustering documents [90]).

5.3.1 Related work

There has been work on interactive *topical* clustering where the user corrects clustering errors on a *document* basis [8], but that effort is more time consuming than feedback on features [90]. Other recent work has had the user select important keywords for (supervised) categorization, thereby leveraging the user's prior knowledge [31, 90]—approaches that are more like that of our framework. Raghavan et al. [90] further support this direction in the finding that users can identify useful features with reasonable accuracy as compared to an oracle. Liu et al. [72] experiment with labeling words instead of documents for text classification, providing the user with a

list of candidate words from which to select potentially good seed words, based on which a training set is constructed from a set of unlabeled documents. A classifier is then constructed given this training set. Liu et al.'s document representation is the standard BOW, which has strong topical flavor, and therefore cannot be used for clustering by arbitrary criteria (for example, our preliminary experiments show that BOW is not appropriate for clustering by author's mood). In addition, Liu et al.'s method involves the user only at the initial step (selecting seed words), limiting the user's control of the classification process.

Although the supervised task of *classification* by sentiment has been widely addressed in the literature (see, e.g. [84] and references therein), *clustering* by sentiment has been very sparsely covered. Turney [107] performs a *binary* clustering of product reviews by authors' sentiment, where only two clusters of documents are constructed: positive reviews and negative reviews. We are not aware of previous work on clustering by sentiment that goes beyond the binary approach. In this section, however, we cluster reviews into four groups, corresponding to the categories of strongly positive, somewhat positive, somewhat negative and strongly negative reviews.

5.3.2 Interactive clustering scenario

Here we provide a step-by-step recipe for clustering documents by a particular criterion that the user has in mind:

1. **Specify the number of clusters:** Learning the natural number of clusters still remains an open problem. We do not attempt to solve it in this thesis, instead the user is asked to specify the desired number of clusters.

2. **Specify feature types:** A list of various *feature types* is provided to the user. Examples of such types are: bag of words or word n-grams, POS tags or POS tag n-grams, punctuation, parse subtrees and other types of syntactic and semantic patterns that can be extracted from text. Such a list can hypothetically include

a large variety of feature types that would respond to everyone's needs. From this list the user is asked to choose one or more types that best serve the particular clustering criterion.

3. **Give examples of features:** For each feature type chosen, the user should attempt to construct (small) sets of seed features that correspond to each category of documents. Sometimes this task is easy: e.g., if the clustering criterion is authors' sentiments, then words such as 'excellent', 'brilliant' etc. would correspond to the category of positive documents, while 'terrible', 'awful' etc. would correspond to the negative category. However, when such sets cannot be easily constructed (e.g. it is non-trivial to come up with good feature examples for clustering by genre—see Section 4.9), the user can skip this step.

4. **Default clustering:** If m feature types are chosen, but no seed features are provided by the user, the standard (unsupervised) clustering scheme is applied (see Chapter 4).

5. **Interactive Clustering:** For the cases when the user has provided seed features for some of the feature types, we propose a new model for Comraf clustering, which combines regular clustering of non-seeded variables with an incremental, bootstrapping procedure for seeded variables:

   (a) Represent documents as distributions over the sets of seed features. Ignore documents with zero probability given the seed features. Cluster the remaining documents using a Comraf clustering method.

   (b) Stop if most documents have been clustered (for details, see Section 5.3.3 below).

   (c) Represent *all* features of the *clustered* documents as distributions over the document clusters. Ignore features that have zero probability given the

clustered documents. Cluster the remaining features using the distributional clustering method.

(d) Select feature clusters that contain the original seed words. Let the user revise the selected clusters: noisy features can be deleted; misplaced features can be relocated; new features can be added. The revised clusters of features are the new sets of seed features. Go to 5(a).

5.3.3 Clustering by sentiment

Following the procedure described in Section 5.3.2 above, after choosing the number of clusters and particular feature types, the user is asked to select a few seed features for each category. For clustering by sentiment, as well as for somewhat similar tasks of clustering by authors' mood or by familiarity with the topic, relevant feature types may be words or word n-grams (i.e. semantic features). However, for other quite close tasks, e.g. clustering by authors' age, not only semantics but also syntax can matter: children, for instance, use certain words more often than adults do; children also tend to use primitive (and sometimes erroneous) syntactic constructions ("me going bye-bye" etc.). In this section, for simplicity, we experiment with word features only.

The task of selecting "sentimental" seed words has two issues. First, it is easier to come up with words that correspond to *extreme* sentimental categories ('spectacular', 'horrible'), but it is difficult to choose seed words for intermediate, mild categories. Nevertheless, as we will see in Section 5.3.6 users usually succeed in accomplishing this task. Second, in our early experiments, users consistently tended to choose words that were out of the vocabulary of a given dataset. Inspired by Liu et al. [72], we decided to provide the users with a word list, to narrow her search only to the dataset vocabulary. Unlike Liu et al. [72], whose task is topical clustering, we cannot automatically predict which words would be relevant. Instead, we employ Zipf's law

and provide the user with a list of words from the interior of the frequency spectrum. We anticipate such a list to contain the most relevant seed words.

We then perform an iterative process of clustering that allows user's involvement in between clustering iterations. We apply a bi-modal Comraf model: we first cluster documents that contain the selected seed words and then we cluster all words of these documents. In the latter step, our seed word groups are enriched with new words that have been clustered together with the original seed words. The user is then asked to edit the new seed word groups, in order to correct possible mistakes made by the system (word removal, relocation and addition is allowed). By this, a clustering iteration is completed and the next iteration can be executed.

Since the seed word groups have been enlarged, we can expect that a set of documents that contain these seed words is now larger as well, so that the clustering process will cover more and more documents from iteration to iteration. The process stops when no more documents are added to the pool. Documents that have never been covered (the ones that contain no seed words from the largest seed word groups) are considered to be clustered incorrectly. An alternative approach to guarantee the algorithm's convergence would be to require enlargement of seed word groups such that at least one document is added to the clustering at each iteration. The algorithm would then stop when the entire dataset is covered. We choose the former approach because (a) we do not want to put additional constraints either on the user or on the Comraf clustering model; (b) in each real-world dataset there can be documents whose sentimental flavor is hard to identify—it would not be beneficial to force such documents into any of the sentimental clusters.

5.3.4 Dataset

We evaluate our interactive clustering system on a dataset of movie reviews. Our dataset consists of 1613 reviews written on *"Harry Potter and the Goblet of Fire*

(2005)" that we downloaded from `IMDB.com` in May 2006.[1] The data was preprocessed exactly as the BNC corpus (Section 4.9.1). We ignore reviews that do not have rating scores assigned by the user. The IMDB's scoring system is from 1 (the worst) to 10 (the best). Based on our extensive experience with `IMDB.com`, we translate these scores into four categories as follows: scores 1 to 4 are translated into the category *strongly disliked* (292 documents), scores 5 to 7 are translated into *somewhat disliked* (454 documents), scores 8 and 9 into *somewhat liked* (447 documents), and score 10 is translated into the category *strongly liked* (420 documents). We do not introduce a neutral category because there are very few neutral reviews on `IMDB.com`.

5.3.5 Experimental setup

On the task of clustering by sentiment, we compare our method's performance with that of k-means and LDA (Section 4.6.3), as well as with the performance of an SVM classifier trained on 22,476 movie reviews. The training data for the SVM consisted of reviews of 46 popular Hollywood movies released in 2005, of the same genre as *Harry Potter*. The reviews and genre labels of movies are obtained from `IMDB.com`. Again, we ignore reviews without user-assigned rating.

To compare our Comraf clustering with other clustering methods, we again use the micro-averaged clustering accuracy, as described in Section 4.6.1. It is not obvious however how to compare Comraf clustering results with SVM classification results. In [16], we show that the clustering accuracy can be directly compared with the (standard) classification accuracy if a constructed clustering is *well-balanced*, meaning

[1] Bo Pang [84] maintains a popular dataset of movie reviews that, unfortunately, does not fully correspond to our task because (a) we want to differentiate the problem of clustering by sentiment from the topical clustering—for this reason our dataset contains reviews written on *one* movie only, so that the *topic* of all the reviews is potentially the same; (b) movie ratings in Bo Pang's dataset are extracted from the reviews' text, which is an error-prone procedure, whereas in our dataset the ratings are assigned by the reviewers using an HTML form which leaves no room for errors.

Doc repres.	k-means	LDA	Comraf	SVM
BOW	28.2	37.0 ± 0.2	40.3 ± 0.8	39.1 ± 0.3
Sentim. list	29.0	40.2 ± 0.5	43.0 ± 0.9	41.3 ± 0.6
Interactive clustering (Oracle)			$\mathbf{47.1 \pm 0.2}$	n/a
Simulated classification (Oracle)			46.3 ± 0.1	

Table 5.1. Clustering by sentiment. Clustering accuracy of Comraf models (both interactive and non-interactive) is compared with clustering accuracy of k-means and LDA, as well as with classification accuracy of SVM. All results are averaged over four independent runs. Standard error of the mean is shown after the $\pm$ sign.

that each category prevails exactly in one cluster. It appears that all our clusterings obtained using the Comraf model are well-balanced.

The system is evaluated on five users who are familiar with the task of document clustering. The users were explained the idea behind interactive clustering and provided a brief description of the dataset. They were given a list of 563 words that appeared in $50 \leq n < 500$ documents in our dataset. The users proceeded as described in Section 5.3.2. Also, we construct an *oracle* as follows: for each category t we select 25 most frequent words that belong to a given list of sentimental words[2] and their distribution over the categories has a peak at t. Unlike human users, the oracle does not provide feedback between clustering iterations. To some extent, the oracle's performance can be considered as an upper bound to results obtained in practice, when a human user is involved.

We perform a *simulated classification* (SC) experiment analogous to the one of Liu et al. [72] (see a description in Section 5.3.1), where the seed words are provided by our oracle. We replace an ad-hoc kNN-like clustering in Liu et al.'s implementation by our effective Comraf clustering, and a Naive Bayes classifier by an SVM.

[2]Our list of 4295 sentimental words was obtained as described in [38].

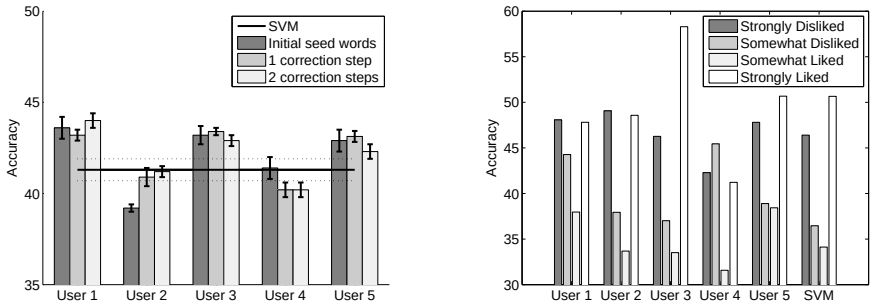


Figure 5.3. Interactive clustering by sentiment. Micro-averaged clustering accuracy over various users: (left) over interactive learning iterations (with original seed words only, after one correction step and after two correction steps). The horizontal line is SVM performance (after feature extraction using a given list of sentimental words, and after training on over 20K documents); (right) over categories of the dataset after two correction steps.

5.3.6 Comparative results

Table 5.1 summarizes our observations. Surprisingly, with BOW features, our Comraf clustering method performs as well as an SVM trained on a large amount of data (Row 1). The good performance of our unsupervised method (with BOW) indicates that the constructed topical clustering sheds some light on reviewers' sentiments, which can occur when the reviewers have a consensus on certain aspects of the movie, e.g. liked the actors but disliked the plot etc.

After feature selection according to our list of sentimental words, the Comraf achieves a significant boost in accuracy surpassing the SVM (Row 2). Using an oracle in our interactive clustering setup (Row 3) improves the performance even further, while the SC result (Row 4) is only slightly (but significantly) inferior. These two results are close because the training set of SC is identical to the clustering constructed at the first iteration of the Comraf algorithm. Since its size appears to be over 3/4 of the entire dataset, there is almost no room for the actual diversity in performance of the two methods.

Figure 5.3 (left) shows the micro-averaged clustering accuracy for each user and each iteration. For three of the five users, selection of the initial seed words is sufficient to obtain significantly higher accuracy than the best result of the SVM. User 2 has significantly lower accuracy than the baseline to begin with, but over the two correction steps is able to provide the necessary feedback so as to obtain an improvement in accuracy, equaling the baseline. We found that User 2 was fairly conservative in her assessment of terms in the beginning marking only 26 terms, while User 1 (the one with the best average performance) marked 58 terms, 23 of which were in common with User 2. User 4 reported that she aggressively removed words at the first correction step, which caused a noticeable drop in the performance.

Figure 5.3 (right) shows the accuracy per class, per user at the end of 3 iterations. User 1 and User 2 have near identical accuracies on the two extreme categories (*strongly liked* and *strongly disliked*), but User 1 has higher accuracies on the intermediate categories, resulting in higher micro-averaged accuracy. It is apparent from this figure that users are able to come up with good features for the two extreme categories, but have difficulties with the intermediate categories. The figure also shows the performance of SVM (with sentiment features). It is interesting to note that the SVM's pattern of behavior is almost identical to the interactive Comraf's.

5.4 Summary

In this chapter, we have shown that Comrafs can be straightforwardly applied to semi-supervised learning, while either adjusting the Comraf graph or the Comraf inference algorithm. As the semi-supervised setup can be viewed as an instance of a supervised setup, we can make a statement that Comrafs are applicable to the entire spectrum of machine learning tasks.

On the task of semi-supervised clustering, we showed that Comraf models outperform a popular constrained optimization method. We also showed that Comraf

models are very robust to the noise in data labels. Our preliminary results demonstrate applicability of the Comraf framework to transfer learning, which has a variety of interesting applications.

We also applied the Comraf framework to non-topical clustering of documents, by introducing the interactive clustering model. We showed that interactive clustering, which is a semi-supervised version of an unsupervised clustering scheme, can potentially outperform one of the best supervised learning methods (SVM), trained on a large amount of labeled data. This result raises an important question that has not been widely addressed in the machine learning literature: for a particular unlabeled dataset, would it be more beneficial to train a supervised model on similar, yet different, data (such as, train a classifier on reviews of movie A and apply it to reviews of movie B), or it would be better to construct a clustering model that takes advantage of some limited knowledge on the unlabeled data, as provided by the user? It appears that the former approach is quite popular. In this chapter we provided evidence that the latter approach can be more effective.

CHAPTER 6

COMRAFS FOR ONE-CLASS CLUSTERING

As we discussed in Section 3.4, each Comraf model is a trinity of a Comraf graph G, an objective function that is factored over G, and an inference procedure for optimizing this objective function. So far, we have experimented with various Comraf graphs (with or without observed nodes) for multi-modal clustering, where the objective is a sum of pairwise Mutual Information terms (3.1), and the inference procedure is a variant of MDC (see Section 4.3). Exploring the variety of Comraf graphs led us to proposing models for email clustering (Section 4.6), clustering scientific papers (Section 4.8), document clustering by genre (Section 4.9), semi-supervised clustering (Section 5.1), and clustering with transfer learning (Section 5.2). In Section 5.3, however, we went beyond this scope and proposed an enhancement to the MDC inference procedure that led to an interactive clustering model. Note that the interactive clustering model exploits the same Comraf graph and objective function as other multi-modal clustering methods we proposed. This chapter goes further in investigating *the role of the objective function* in Comrafs. Specifically, we focus on the problem of constructing an objective function that best suits a particular application.

We address this problem on a representative task of *one-class clustering*, which is the task of identifying the most coherent subset of documents (*the core*) from a given pool of documents. This pool can be generated by a search engine (as a set of documents retrieved on a given query); also, this pool can be an email Inbox, a repository of scientific papers etc. One-class clustering is a technically simpler task than the general (multi-class) clustering: on a given dataset, a *binary* (as opposed

to a k-ary) predicate is constructed that answers the question of whether or not a data instance belongs to the core. This simplicity allows for a theoretical analysis of *optimality* of the one-class clustering method proposed.

Similar to many other unsupervised learning problems, the problem of one-class clustering is generally ill-posed as one can argue that the shortest document in a collection satisfies the criterion of being the most coherent subset. We resolve that issue by introducing a parameter k, which is the number of documents in the core subset. This parameter is analogous to the number of clusters in (multi-class) clustering, the number of outliers [105] or the radius of Bregmanian ball [28] in other formulations of one-class problems.

Note that formally the problem of one-class clustering is a special case of the general, multi-class clustering: one-class clustering is a problem of constructing $n - k + 1$ clusters of n data instances, where one cluster is of size k and all the others are singletons. However, since explicit modeling of singleton clusters appears to be useless, from the practical point of view the two problems become different: methods applicable for one-class clustering are generally unapplicable to multi-class clustering and vice versa. Also note that the problem of one-class clustering is a compliment to an unsupervised formulation of the *outlier detection* problem [1, 105]: once the core cluster is constructed, all the non-core data instances are considered outliers.

Speaking in terms of Comraf models, for one-class clustering we define combinatorial random variables over all the possible *subsets* of a modality (or, in other words, over its *powerset*). Recall that for multi-class clustering we defined combinatorial random variables over all the possible *partitionings* of a modality. Although the Comraf graph's layout appears to be the same for both tasks, the one-class clustering objective function is different from that of multi-class clustering, and so is the inference procedure. In this chapter, we construct step-by-step the objective function

and inference procedure, starting from an artificially simplified case and ending with the real-world application.

Our working assumption throughout this chapter is that the core documents share a (relatively) small lexicon, while the remaining documents (*the noise*) do not have much in common (i.e. they are randomly drawn from the pool of all existing documents written in the English language). Our methods, however, will work equally well in situations when the noise has *some* structure, meaning that some non-core documents share their topics.

We describe the simplest Comraf model with only two modalities: documents and words. Despite its simplicity, this setup allows for three different approaches to one-class clustering of documents:

- Identify the shared lexicon (the subset of *relevant words*), i.e. solve the one-class clustering problem for words. A document will then be considered a part of the core if it contains enough relevant words. We describe this setup in Section 6.2. The fundamental question we answer in that section is whether or not the subset of relevant words *can* be identified in document collections of feasible size. We show analytically that, under some simplifying assumptions, the subset of relevant words can be optimally identified in document collections of log-linear size (in the size of the vocabulary).

- Directly identify the core documents, based on their distributions over words. This setup is in the focus of Section 6.3. In that section, we propose our information-theoretic objective function. We derive a simple uni-modal algorithm for optimizing this objective. We show that the proposed algorithm is optimal under the assumptions imposed in Section 6.2. We then relax these assumptions and adjust our objective function to the real-world case.

- Perform *one-class co-clustering (OCCC)*, while simultaneously identifying the subset of relevant words and the subset of core documents (see Section 6.4). We generalize the algorithm proposed in Section 6.3 to the bi-modal setup. The resulting OCCC algorithm significantly outperforms the uni-modal one.

In Section 6.5, we propose another, probabilistic objective function for our task: the likelihood that a document belongs to the core. Inspired by Huang and Mitchell [53], we apply an EM inference algorithm to the resulting model.

We evaluate our information-theoretic and probabilistic models on two applications: (a) Web appearance disambiguation (see Section 4.7)—our methods outperform the algorithm proposed in [13]; and (b) re-ranking information retrieval results [65, 36]—we significantly improve the accuracy of original Google's ranked lists, as well as of one-class (unsupervised) SVM and one-class Information Bottleneck [28]. Note that our models can also be applied to other real-world tasks, e.g. to spam detection, news filtering, image retrieval, and basically to any task where a common subset of features can be identified in a subset of data instances.

6.1 Related work

Many previously proposed one-class clustering methods (see [105, 28, 50], and references therein) are vector-space methods, where the goal is to find a convex body of small volume that contains as many data instances as possible. Despite that *binary* vector-space methods have proven themselves to be very effective in the text domain, one-class vector-space methods are problematic. In binary methods, the decision boundary is linear (with or without applying the *kernel trick* [29]). In (vector-space) one-class methods, however, the boundaries are essentially *elliptic*, which is unnatural in the highly multidimensional text domain: core documents tend to lie on a lower-dimensional manifold (see [68]), while elliptic boundaries tend to capture too much space around it.

An alternative solution suggested in [13] (and discussed in Section 4.7) is to simulate one-class clustering in text by first applying traditional multi-class clustering, after which one of the clusters in chosen. Intuitively, this approach makes a wrong design choice: structure is artificially forced on the space of non-core documents, which may not have any underlying structure. The models described in this chapter, in contrast, achieve the main goal of one-class clustering—to identify the most coherent subset of objects—without imposing structural or topological constraints.

Our one-class clustering models have interesting cross-links with models applied to other Information Retrieval tasks. For example, a model similar to our information-theoretic one-class clustering, is proposed by Zhou and Croft [115] for *query performance prediction*. Tao and Zhai [104] describe a *pseudo-relevance feedback* model, which is similar to our probabilistic one-class clustering (see discussion in Section 6.6.2). These types of cross-links are common when the models are general enough and relatively simple. In this work we pay particular attention to the simplicity of our models, such that they are feasible for theoretical analysis as well as for efficient implementation.

6.2 One-class clustering of words

We are given a dataset $\mathcal{D}$ of n documents, each of which is represented as a vector of words, with no importance to their order (i.e., *bag-of-words*). We assume that $\mathcal{D}$ has a core $\mathcal{D}^k$ of k documents written on one topic, while the rest of the $(n-k)$ documents are noise. Let $\mathcal{R}$ be the lexicon of relevant words and $\mathcal{G} \supset \mathcal{R}$ be the general lexicon of $\mathcal{D}$ (i.e. all distinct words of $\mathcal{D}$). Let us denote $m = |\mathcal{G}|$ and $m_r = |\mathcal{R}|$ the sizes of the two lexicons, where $m_r \ll m$. Assuming that the core is not too small ($\frac{k}{n} \gg 0$), our intuition is that a word belongs to $\mathcal{R}$ if it is more frequently used in $\mathcal{D}$ than it would be used in general English. For example, many occurrences of the words "reinforcement", "regression", "classifier" in $\mathcal{D}$ indicate that they are relevant, as the

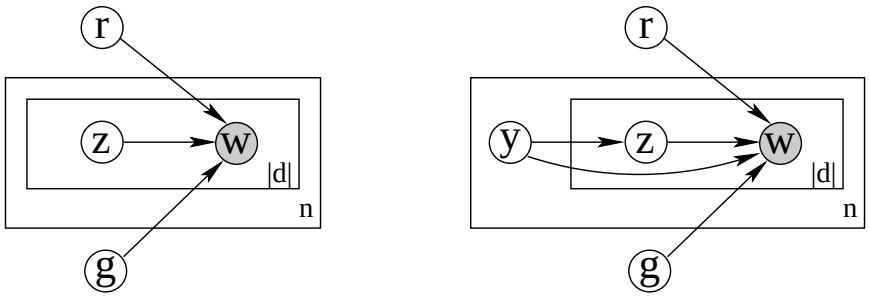


Figure 6.1. (left) The simplest generative model; (right) Latent Topic/Background model (Section 6.5).

probability of observing the same frequency of these words in non-core documents is very low. Our first task is to determine which words belong to $\mathcal{R}$.

We attempt to solve this problem by introducing a simple generative model of documents (see the left panel of Figure 6.1). Given a dataset $\mathcal{D}$ of size n, for each word token in every document, we first decide if it is drawn from a distribution $P_r(W)$ over the the set $\mathcal{R}$ of relevant words, or from a distribution $P_g(W)$ over the set $\mathcal{G}$ of all words in $\mathcal{D}$, and then we choose the word w accordingly. Both $P_r(W)$ and $P_g(W)$ are multinomial, where the former has a much smaller support. Note that in our model, for each word token w, the decision whether it is drawn from $P_r(W)$ or from $P_g(W)$ is made independently of the rest of the model, and thus we can think of the dataset $\mathcal{D}$ as a single document of length $N = n|d|$ (here and in the next section, we assume that all the documents are of the same length $|d|$).

To make the following theoretical analysis easier, let us assume (quite unrealistically) that distributions $P_r(W)$ and $P_g(W)$ are *uniform* rather than multinomial. In Section 6.3.1 we relax this assumption by flattening multinomials using a correction term. Under the uniformity assumption, an algorithm for identifying relevant words is straightforward: obtain a sample of size N and choose words with counts above a certain threshold to be in $\mathcal{R}$ (see an illustration in Figure 6.2 left). The major drawback of this algorithm is that we should know the exact value of the threshold

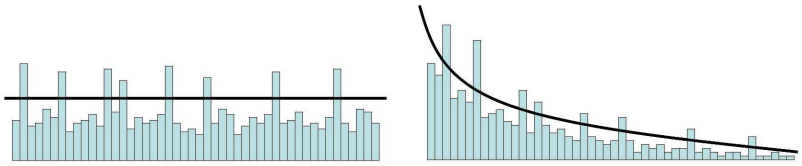


Figure 6.2. An illustration of possible distributions of word counts in one-class clustering: (left) uniform case; (right) multinomial case. Words whose counts are above the threshold are considered relevant. Note that in the multinomial case counts of some relevant words can be lower than counts of non-relevant words.

(an estimation is not enough here). An alternative algorithm would be: obtain a sample of size N, sort words in decreasing order of their counts and choose the first m_r words to be in $\mathcal{R}$. Clearly, the two algorithms are asymptotically equivalent (they identify the same set $\mathcal{R}$ if the sample size N is large enough).

An important question is how large should be the sample size N so that the sets of relevant and non-relevant words will be separable. For instance, if $N = O(m^2)$ samples are required, the algorithm described above will be infeasible in any real-world case. In the following theorem, we prove that a log-linear sample size is enough. Let us first introduce some notation. For a document d_i and a word token w_{ij}, let π be the probability of drawing w_{ij} from the pool of relevant words $\mathcal{R}$, that is: $P(Z_{ij} = 1) = \pi$. Let $p_w = \frac{m_r}{m} = \frac{|\mathcal{R}|}{|\mathcal{G}|}$ be a fraction of relevant words in the dataset's vocabulary.

Theorem 6.2.1 *To determine the set $\mathcal{R}$ with probability $1 - \delta$, we need at most*

$$N = 16\frac{m}{\pi}\ln\frac{m}{\delta} \tag{6.1}$$

samples, under a (weak) constraint of $p_w < 2\pi$.

The proof of this theorem is relatively straightforward—it involves an application of the Chernoff bound and the union bound. We prove this theorem in Appendix A. Now, under the uniformity assumption and conditions imposed in Theorem 6.2.1, we can identify the set $\mathcal{R}$ of relevant words with arbitrarily high probability. The

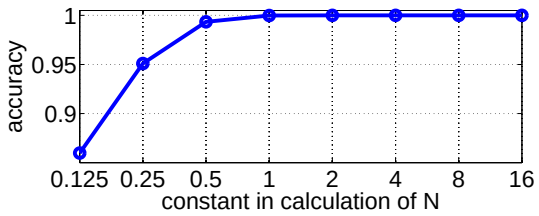


Figure 6.3. The accuracy (as defined in Section 6.6, averaged over 100 independent runs) of identifying $\mathcal{R}$ in a simulation of the generative process, over various values of the constant from Equation (6.1) for the sampling size N. In Equation (6.1), the value of this constant is set to 16. Here we show that the value of 2 is enough in practice.

relevance of a document is then determined by the cumulative relevance of words occurring in the document. Consequently, the core $\mathcal{D}^k$ will consist of k documents, each of which contains more words from $\mathcal{R}$ than any document from $\mathcal{D} \setminus \mathcal{D}^k$.

We simulated the generative process for various values of π and p_w. We saw that in practice many fewer sampling iterations were required for identifying the set $\mathcal{R}$ with 100% accuracy. In Equation (6.1), the constant in calculating N is set to 16. We tuned the value of this constant, and showed that the value of 2 is generally enough to perfectly identify $\mathcal{R}$. Figure 6.3 outlines some results on synthetic data that has similar characteristics to our WAD dataset (see Section 4.7.3): we choose $m = 12000$, $\pi = p_w = 0.2$, and $\delta = 0.01$. For $N = 330,000$, which is the size of the WAD dataset, we obtain 98.5% accuracy. This implies that if words in text datasets were indeed distributed uniformly, the one-class clustering problem would be easy.

6.3 Min-Entropy algorithm for one-class clustering in text

Obviously, the trivial one-class clustering algorithm from Section 6.2 above is applicable only under the restrictive uniformity assumption. Sticking to the uniformity assumption for now, we propose an alternative formal criterion, which in Section 6.3.1 will be adjusted to the practical case. Based on this criterion, we design an algorithm that directly identifies the core, and show that this algorithm is optimal under the

uniformity assumption. Let us define a word entropy of the dataset $\mathcal{D}$ as:

$$H(W) = H_P(P(W)) = -\sum_{w \in \mathcal{G}} P(w) \log P(w) = -\sum_{d \in \mathcal{D}, w \in \mathcal{G}} P(d, w) \log P(w), \quad (6.2)$$

where P is an empirical distribution of words in $\mathcal{D}$: given that a word w occurs $N_{w \in d}$ times in a document d, and N_w times in the entire dataset, we let $P(d, w) = \frac{N_{w \in d}}{N}$ and $P(w) = \sum_d P(d, w) = \frac{N_w}{N}$. Define a *document-word entropy* of a document d as:

$$H_d(W) = -\sum_{w \in \mathcal{G}} P(d, w) \log P(w) = -\sum_{w \in d} P(d, w) \log P(w). \quad (6.3)$$

Note that the word entropy (6.2) is additive: $H(W) = \sum_{d \in \mathcal{D}} H_d(W)$. The document-word entropy $H_d(W)$ captures our intuition of a core document: documents that mainly use frequent words have low $H_d(W)$. To see this, we factorize the joint $P(d, w) = P(d)P(w|d)$, and assume that all documents have a uniform prior $P(d) = \frac{1}{n}$. Thus, $H_d(W)$ is the expectation of $-\log P(w)$ according to the word frequency $P(w|d)$ in d, which is small if d uses a lot of frequent words.

Based on this observation, for each subset $\mathcal{D}^k$ of size k, we define our objective as $\mathcal{D}^k$'s contribution to the word entropy (6.2):

$$H^k(W) = \sum_{d \in \mathcal{D}^k} H_d(W) = -\sum_{d \in \mathcal{D}^k, w \in \mathcal{G}} P(d, w) \log P(w). \quad (6.4)$$

We argue that the most coherent subset $\mathcal{D}^k$ is the one that minimizes this objective. To find the most coherent $\mathcal{D}^k$, we use the following simple, greedy *Min-Entropy* algorithm:

1. Sort documents according to their word entropy portion (6.3), in increasing order.

2. Select the first k documents. Eliminate all the rest.

Since our objective (6.4) is additive in documents, its *global* minimum is found by the above algorithm.

We now show that this algorithm is optimal under the uniformity assumption. Indeed, if the dataset $\mathcal{D}$ is large enough, then according to Theorem 6.2.1 (with high probability) any relevant word w has a lower word-score $-\log P(w)$ than any non-relevant word, because relevant words are more frequent in $\mathcal{D}$. Since we assume that all documents are of the same length ($|d|$ is constant), the Min-Entropy algorithm chooses documents that contain more relevant words than any other document in the dataset. But this is exactly the main property of the core, as discussed in Section 6.2. Therefore, the Min-Entropy algorithm identifies the core. We summarize this observation in the following theorem:

Theorem 6.3.1 *If the dataset $\mathcal{D}$ is large enough, then with high probability over datasets, the Min-Entropy algorithm is optimal for the one-class clustering problem under the uniformity assumption.*

6.3.1 Relaxation of the uniformity assumption

In practice, distributions $P_r(W)$ and $P_g(W)$ are multinomial rather than uniform (see illustration in Figure 6.2 right). We modify the theory presented above to this case by exploiting the fact that entropy of a distribution can be viewed as Kullback-Leibler (KL) divergence between this distribution and a uniform one. In place of the entropy from Equation (6.2), we propose to use KL divergence:

$$KL(P||Q) = \sum_{w \in \mathcal{G}} P(w) \log \frac{P(w)}{Q(w)} = \sum_{d \in \mathcal{D}, w \in \mathcal{G}} P(d, w) \log \frac{P(w)}{Q(w)}, \tag{6.5}$$

where $Q(w)$ is an estimation of the true probability of a word occurrence in the English language. This modification can be thought of as an adjustment of the empirical word

distribution in $\mathcal{D}$ to the uniform one. An algorithm analogous to Min-Entropy aims at finding a subset $\mathcal{D}^k$ that maximizes its portion in (6.5):

$$KL^k(P||Q) = \sum_{d \in \mathcal{D}^k, w \in \mathcal{G}} P(d,w) \log \frac{P(w)}{Q(w)}, \tag{6.6}$$

Thus, we identify the core $\mathcal{D}^k$ as a subset of documents containing many words that occur in $\mathcal{D}$ more frequently than in general English. Following [94, 13], we exploit Web counts of words: we estimate $Q(w)$ as a normalized count of w in the Web. The Web counts are obtained using Google API.

6.4 One-class co-clustering (OCCC)

As discussed in Section 4.1, co-clustering is a special case of multi-modal clustering, where only two interacting modalities are considered. In the text domain, co-clustering usually implies clustering documents D and words W, either sequentially [99], or iteratively [39].

In the one-class clustering case, the co-clustering framework is interpreted as constructing *one* cluster of core documents, together with *one* cluster of relevant words. The co-clustering idea has special importance for one-class clustering, as we want to diminish the influence of non-relevant words on the process of selecting core documents. In many real-world cases, where $|\mathcal{R}| \ll |\mathcal{G}|$, the mass of non-relevant words in the mixture $p(W)$ is dominant, while only relevant words are responsible for a document to be relevant. Reducing this mass is the goal of one-class co-clustering.

By examining Equation (6.6), it is natural to define a score of word relevance as:

$$s(w) = \log \frac{P(w)}{Q(w)}. \tag{6.7}$$

such that our objective function (6.6) is the weighted average of these scores. For co-clustering we propose to replace the objective (6.6) with the following:

$$KL^k(P||Q) = \sum_{d \in \mathcal{D}^k, w \in \mathcal{R}} P'(d,w) \log \frac{P(w)}{Q(w)}, \tag{6.8}$$

where $P'(d,w) = P(d,w)/(\sum_{w \in \mathcal{R}} P(d,w))$ is a joint distribution of documents and (only) relevant words. Because of the re-normalization introduced, it is not obvious how to find the global optimum of the objective (6.8). We thus propose to approximate it using a simple, *sequential* One-Class Co-Clustering (OCCC) algorithm: we first build a cluster of relevant words based on which we build a cluster of core documents,[1] as follows:

1. Sort words according to their scores from Equation (6.7), in decreasing order.
2. Select a subset $\mathcal{R}$ of first m_r words.
3. Represent documents as bags-of-words over $\mathcal{R}$ (delete counts of all words from $\mathcal{G} \setminus \mathcal{R}$).
4. For each document d, calculate its portion in Equation (6.8):

$$KL_d(P||Q) = \sum_{w \in \mathcal{R}} P'(d,w) \log \frac{P(w)}{Q(w)} = \sum_{w \in \mathcal{R} \cap d} P'(d,w) \log \frac{P(w)}{Q(w)}, \tag{6.9}$$

5. Sort documents according to their scores from Equation (6.9), in decreasing order.
6. Select a subset $\mathcal{D}^k$ of the first k documents.

Despite its simplicity, the OCCC algorithm shows excellent results on real-world data (see Section 6.6). The algorithm's complexity is particularly appealing: $O(N)$, where N is the number of word tokens in $\mathcal{D}$.

---

[1] In this simplest algorithm, word clustering is analogous to *feature selection*, in which selected features correspond to only *one class* of the data. In more complex algorithms though, this analogy will be less obvious.

6.4.1 Heuristic for choosing the size of word cluster

The choice of m_r can be crucial. While not proposing a comprehensive method for choosing m_r, we propose a useful heuristic. The distribution of scores $s(w)$ for relevant words can be modeled by a normal distribution with mean $\mu_r \gg 0$ and variance σ_r^2. Analogously, the distribution of word scores for non-relevant words is modeled by a normal distribution with mean $\mu_{nr} = 0$ and variance σ_{nr}^2. We assume that all the words with negative scores are non-relevant. Since the normal distribution is symmetric, we further assume that the number of non-relevant words with negative scores equals the number of non-relevant words with positive scores. Therefore, our estimate of total non-relevant words is twice the number of words with negative scores, and the number of relevant words can thus be estimated as $m_r = m - 2 \cdot$ #{words with negative scores}.

6.5 The Latent Topic/Background (LTB) model

Here we revise our generative model from Section 6.2 and propose another one-class clustering algorithm based on probabilistic inference. Our new generative model is shown in the right panel of Figure 6.1. For each document d_i, Y_i is a Bernoulli random variable where $Y_i = 1$ corresponds to d_i being relevant. For each word token w_{ij}, Z_{ij} is a Bernoulli random variable where $Z_{ij} = 1$ means that w_{ij} is sampled from the multinomial distribution $P_r(W)$ over relevant words, otherwise it is sampled from the general multinomial distribution $P_g(W)$ over all words in $\mathcal{D}$.

Following [53], we admit that not all words in a relevant document should be relevant. In our model, if a document belongs to the core ($Y_i = 1$), for each its word we make a decision (based on Z_{ij}) whether it is sampled from $P_r(W)$ or $P_g(W)$. However, if a document does not belong to the core ($Y_i = 0$), each its word is sampled from $P_g(W)$, i.e. $P(Z_{ij} = 0|Y_i = 0) = 1$.

We use the Expectation-Maximization (EM) algorithm to learn parameters of our model from the dataset. We now describe the model parameters Θ. First, the probability of a document belonging to the core is denoted by $P(Y_i = 1) = \frac{k}{n} = p_d$ (this parameter is fixed and will not be inferred from data). Second, for each document d_i, we maintain a probability of each its word being relevant (given that the document is relevant), $P(Z_{ij} = 1|Y_i = 1) = \pi_i$ for $i = 1, \ldots, n$. Third, for each word $w_l|_{l=1}^m$ we let $P(w_l|Z_l = 1) = p_r(w_l)$ and $P(w_l|Z_l = 0) = p_g(w_l)$. The overall number of parameters is $n + 2m + 1$, one of which (p_d) is preset. The dataset likelihood is then:

$$P(\mathcal{D}) = \prod_{i=1}^{n} [p_d \; P(d_i|Y_i = 1) + (1 - p_d)P(d_i|Y_i = 0)] =$$

$$= \prod_{i=1}^{n} \left[p_d \prod_{j=1}^{|d_i|} [\pi_i p_r(w_{ij}) + (1 - \pi_i)p_g(w_{ij})] + (1 - p_d) \prod_{j=1}^{|d_i|} p_g(w_{ij}) \right].$$

At each iteration t of the EM algorithm, we first perform the E-step, where we compute the posterior distribution of hidden variables $\{Y_i\}$ and $\{Z_{ij}\}$ given the current parameter values Θ^t and the data $\mathcal{D}$. Then, at the M-step, we compute the new parameter values Θ^{t+1} that maximize the model log-likelihood given Θ^t, $\mathcal{D}$ and the posterior distribution.

The initialization step is crucial for the EM algorithm. Our pilot experimentation showed that if distributions $P_r(W)$ and $P_g(W)$ are initialized as uniform, the EM results are close to random. Therefore, we borrow an idea from our OCCC model (Section 6.4) and initialize word probabilities proportional to their relevance scores from Equation (6.7). Initialization of π_i parameters, which are the ratio of relevant words in relevant documents, is a problem analogous to determining the word cluster size in OCCC (see Section 6.4.1). We do not propose the optimal way to initialize π_i parameters, however, as we show later in Section 6.6, the EM algorithm appears to be quite robust to the choice of π_i, namely, $\pi_i = 0.5$ (or close to that) leads to a good result.

Input:
$\mathcal{D}$ – the dataset
$s(w_l)$ – score for each word $w_l|_{l=1}^m$, from Equation (6.7)
T – number of EM iterations
Output: Posteriors $P(Y_i = 1|d_i, \Theta^T)$ for each document $d_i|_{i=1}^n$

Initialization:
for each document d_i **initialize** π_i^1
for each word w_l **initialize** $p_r^1(w_l) = \frac{1}{S_r}\exp(s(w_l))$; $p_g^1(w_l) = \frac{1}{S_g}\exp(-s(w_l))$,
where S_r and S_g are normalization factors
Main loop:
For each $t = 1, \ldots, T$ **do**
E-step:
for each document d_i **compute** $\alpha_i^t = P(Y_i = 1|d_i, \Theta^t)$
for each word token w_{ij} **compute** $\beta_{ij}^t = P(Z_{ij} = 1|Y_i = 1, w_{ij}, \Theta^t)$
M-step:
for each document d_i **update** $\pi^{t+1} = \frac{1}{|d_i|}\sum_j \beta_{ij}^t$
for each word w_l **update**

$$p_r^{t+1}(w_l) = \frac{\sum_i \alpha_i^t \sum_j \delta(w_{ij} = w_l)\ \beta_{ij}^t}{\sum_i \alpha_i^t \sum_j \beta_{ij}^t}; \qquad p_g^{t+1}(w_l) = \frac{N_w - \sum_i \alpha_i^t \sum_j \delta(w_{ij} = w_l)\ \beta_{ij}^t}{N - \sum_i \alpha_i^t \sum_j \beta_{ij}^t}$$

Algorithm 4: EM algorithm for one-class clustering using the LTB model.

The EM procedure is sketched in Algorithm 4. We omit minor details, see Appendix B for more detailed description of the algorithm. After T iterations, we sort the documents according to α_i in decreasing order and choose the first k documents to be the core. The complexity of our implementation of Algorithm 4 is $O(TN)$. To avoid overfitting, we set T to be a small number: in our experiments we fix $T = 5$.

6.6 Experimentation with OCCC and LTB

To define our evaluation criteria, let C be the constructed cluster and let C_r be its portion consisting of documents that actually belong to the core. Precision is then defined as Prec $= |C_r|/|C|$, recall as Rec $= |C_r|/k$ and F-measure as (2 Prec Rec)/(Prec+Rec). In all our experiments we fix $|C| = k$, such that precision equals recall and is then called *one-class clustering accuracy*, or just *accuracy*.

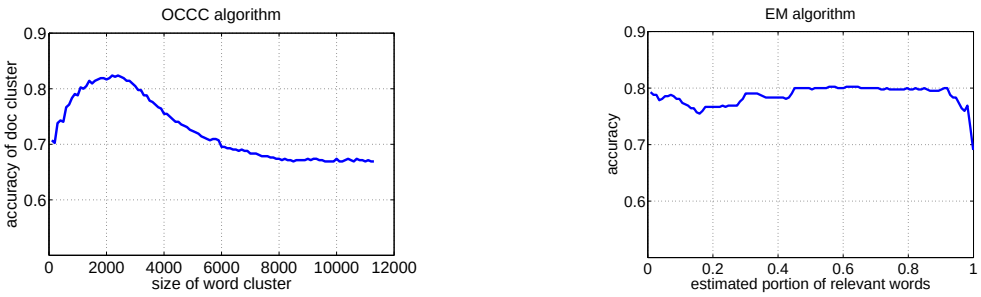


Figure 6.4. Web appearance disambiguation. (left) OCCC accuracy as a function of the word cluster size; (right) accuracy of LTB (with the underlying EM algorithm) over various initializations of π_i parameters: LTB shows a more robust behavior than OCCC, however LTB's maximal result (80.2%) is slightly inferior to the OCCC's (82.4%).

6.6.1 Web appearance disambiguation

The Web appearance disambiguation (WAD) task is described in Section 4.7. WAD is a classic one-class clustering task, that was solved in that section using a *simulated* one-class clustering method: multiple clusters are constructed, out of which one cluster is then selected. Here we propose a more effective solution.

We test our methods on the WAD dataset (Section 4.7.3). The dataset consists of the 1085 pages, out of which 420 are relevant, so we apply our algorithms with $k = 420$. At a preprocessing step, we binarize document vectors and remove low frequency words (both in terms of $P(w)$ and $Q(w)$). The results are summarized in Figure 6.4. On its left panel, the x-axis corresponds to the hypothetic number of relevant words, and the y-axis to accuracy. The best OCCC performance is obtained with $m_r = 2200$ words: 82.4% accuracy, while the F-measure reported in Section 4.7.5 is 78.4% (on a cluster with less than 420 documents—its recall is only 71.3%).

As can be seen from the left panel of Figure 6.4, the OCCC performance is robust: accuracy above 80% is obtained with a word cluster of any size in the 1000-3000 range. The heuristic from Section 6.4.1 suggests a cluster size of 1000. The right panel of Figure 6.4 shows the LTB accuracy over various initialization values of the π_i parameter (the fraction of relevant words in core documents). We can infer from

#	OCCC	LTB	#	OCCC	LTB	#	OCCC	LTB
1	cheyer	artificial	8	mlittman	proceedings	15	gorfu	kaelbling
2	kachites	learning	9	hardts	computational	16	billmark	andrew
3	quickreview	cs	10	meuleau	reinforcement	17	pomdps	conference
4	adddoc	intelligence	11	dipasquo	papers	18	ml95	markov
5	aaai98	machine	12	shakshuki	cmu	19	agentus	stanford
6	kaelbling	edu	13	xevil	aaai	20	megacanje	models
7	mviews	algorithms	14	sangkyu	workshop			

Table 6.1. Most highly ranked words by OCCC and LTB, on the WAD dataset.

this plot that LTB is even more robust to parameter initialization than OCCC: any but very large (i.e. $\pi_i \approx 1$) values can be chosen.

Finally, Table 6.1 lists the top 20 words according to the models learned by OCCC and by LTB. The OCCC algorithm sorts words according to their score $s(w)$, such that words that often occur in the dataset but rarely in the Web, are on the top of the list. These are mostly last names or login names of researchers, venues etc. The EM algorithm of LTB is given the OCCC's word rank list as an input to initialize $p_r^1(w)$ and $p_g^1(w)$, which are then updated at each M-step. In the LTB column, words are sorted by $p_r^5(w)$. The high quality of the LTB list is due to *explaining away* in our generative model (via the Y_i nodes). Still, OCCC (marginally) outperforms LTB on this dataset: the maximal result obtained by OCCC is 82.4% accuracy, while LTB obtains 80.2% accuracy.

6.6.2 Re-ranking Web retrieval results

Modern search engines are usually successful in identifying relevant documents for a given general-type query. However, in most cases some of the top-ranked documents have only marginal relation to the query. For example, querying Google for *Beatles*, many top-ranked documents indeed talk about the quartet, however, one can see a document about the Apple Corps vs. Apple Computer trial (which is certainly not *about* Beatles), and some other clearly non-relevant documents.

QUERY	GOOGLE	OC-SVM	OC-IB	OCCC	LTB
Godfather	0.444	0.407	0.400	0.852	**0.926**
Bunker Hill	0.487	0.590	0.821	0.897	**0.923**
Beatles	0.400	0.457	0.571	0.629	**0.771**

Table 6.2. Re-ranking Web retrieval results: We compare one-class clustering accuracy of our OCCC (with heuristic from Section 6.4.1) and LTB (initialized with $\pi_i = 0.5$) models with the accuracy of the original Google rank lists, of one-class SVM (OC-SVM) and of one-class Information Bottleneck (OC-IB) [28] with l^2-norm.

In this section, we leverage the high quality of Web retrieval results and attempt to improve them even further. Our assumption is that relevant documents are topically close to each other, while non-relevant documents can be on any topic. We notice that as soon as *a few* relevant documents appear among the n top-ranked results, we can apply our one-class clustering methods to the task of *re-ranking* those results, where the goal is to move relevant documents up in the ranked list, while moving non-relevant ones down the list. In one-class clustering, we identify the most coherent subset (i.e. the core) from a set of n documents. Assuming that core documents are relevant, while non-core documents are non-relevant, we re-organize the ranked list such that core documents are now located above non-core ones, while preserving the initial ordering within both the core and non-core subsets.

Note that the problem of one-class clustering for re-ranking Web retrieval results is similar to the problem of pseudo-relevance feedback (see, e.g. [104]). However, the two problems are still fundamentally different. In pseudo-relevance feedback, one assumes that the first k documents in a ranked list are relevant, and re-ranks the rest of the ranked list based on that assumption. In one-class clustering, in contrast, we make a weaker assumption that the k relevant documents *exist* within the first n documents in a ranked list. Our task is then to discover those k documents and place them on the top of the ranked list.

We test the resulting system on three small datasets that we created for this chapter. Each of them contains 100 first Google hits retrieved on a certain query, labeled as relevant / non-relevant with regards to the major meaning of the query. These queries are:

- ***Godfather.*** While the word Godfather is ambiguous, a query *Godfather* most probably refers to the popular movie/book[2]—other readings are considered non-relevant. Among the set of 100 documents, 27 were annotated as relevant.

- ***"Bunker Hill".*** The phrase *Bunker Hill* is not ambiguous, and a user who types such a query is presumably interested in information about the Bunker Hill battle and/or monument. However, some Bunker Hill mentions are not directly related to the historical event, e.g. *Bunker Hill Community College* or *Bunker Hill Presbyterian Church*. This dataset contains 39 relevant documents.

- ***Beatles.*** The obvious reading of the query *Beatles* is the name of the legendary quartet. All the 100 first Google hits refer to the quartet, however only 35 of them provide information *about* the quartet, such as their biography or discography, while this is (almost) certainly the type of information a user expects to retrieve on query *Beatles*.

We compare our methods with two previously proposed one-class clustering techniques: an unsupervised one-class SVM and a one-class Information Bottleneck (see [28] for details on those methods). Our results are shown in Table 6.2; together with the two baselines, we list the accuracies of the original Google's ranked lists, where the first k documents are considered the core, while the rest of $n-k$ documents are considered the noise. Our methods clearly outperform the baselines, while LTB shows better performance than OCCC.

[2]According to `imdb.com`, *The Godfather* is the world's most popular film to date.

6.6.3 Detecting the topic of the week

As we discussed in this chapter's introduction, the real-world data rarely consists of a clean core and uniformly distributed noise. Usually, the noise has some structure, namely, it may contain coherent components. With this respect, one-class clustering can be used to detect the *largest* coherent component in a dataset, which is an integral part of many applications. In this section, we solve the problem of automatically detecting the *topic of the week (TW)* in a newswire stream, i.e. detecting all articles in a weekly news roundup that refer to the most broadly discussed event.

The TW detection task can be considered as a subtask of *Topic Detection and Tracking (TDT)* [2], and is closely related to:

- **Generating topic overviews** [103]. A *topic overview* is a set of keywords that best describe the discussed topic. Using the one-class clustering terminology, such set is the cluster of relevant words. In our OCCC approach, we generate both a subset of core documents and a subset of relevant words. In LTB, we rank documents and words according to their likelihood of belonging to the core.

- **Discovering thematic changes** [103, 52]. Major topics (represented both as subsets of documents and as their descriptive words) are changing with time. In our work, we deal with those changes by discretizing the timeline into weeks. A topic that was most broadly discussed one week, may or may not remain so the next week.

- **Quantifying trends** [44]. The trend quantification task aims at discovering *how large* a certain topic is, without necessarily mapping documents to topics. In TW detection, however, the task is to discover *which* topic is the largest one. Also, trend quantification is an intrinsically supervised task, while TW detection can be formulated both in terms of supervised and unsupervised learning.

We evaluate the TW detection task on the TDT-5 dataset[3], which consists of 250 news events spread over a time period of half a year, and 9,812 documents In English, Arabic and Chinese (translated to English), annotated by their relationship to those events.[4] The largest event in TDT-5 dataset (#55106, titled *"Bombing in Riyadh, Saudi Arabia"*) has 1,144 relevant documents, while 66 out of the 250 events have only one relevant document each. We split the dataset to 26 weekly chunks (to have 26 full weeks, we delete all the documents dated with the last day in the dataset, which decreases the dataset's size to 9,781 documents). Each chunk contains from 138 to 1292 documents. Over each chunk, we applied our one-class clustering methods in four setups:

- **OCCC with the m_r heuristic** (from Section 6.4.1).
- **OCCC with optimal m_r.** We unfairly choose the number m_r of relevant words such that the resulting accuracy is maximal. This setup can be considered as the upper limit of the OCCC's performance, which can be hypothetically achieved if a better heuristic for choosing m_r is proposed.
- **LTB initialized with $\pi_i = 0.5$.** As we show in Sections 6.6.1 and 6.6.2 above, if π_i parameters are initialized with 0.5, the LTD model shows good results.
- **LTB initialized with $\pi_i = p_d$.** We notice a significant deviation in the core's size among our 26 datasets. Quite naturally, the number of relevant words in a dataset depends on the number of core documents. For example, if the core is only 10% of a dataset, it is unrealistic to assume that 50% of all words are relevant. In this setup, we condition the ratio of relevant words on the ratio of core documents.

[3] `http://projects.ldc.upenn.edu/TDT5/`

[4] We take into account only labeled documents, while ignoring unlabeled documents that can be found in the TDT-5 data.

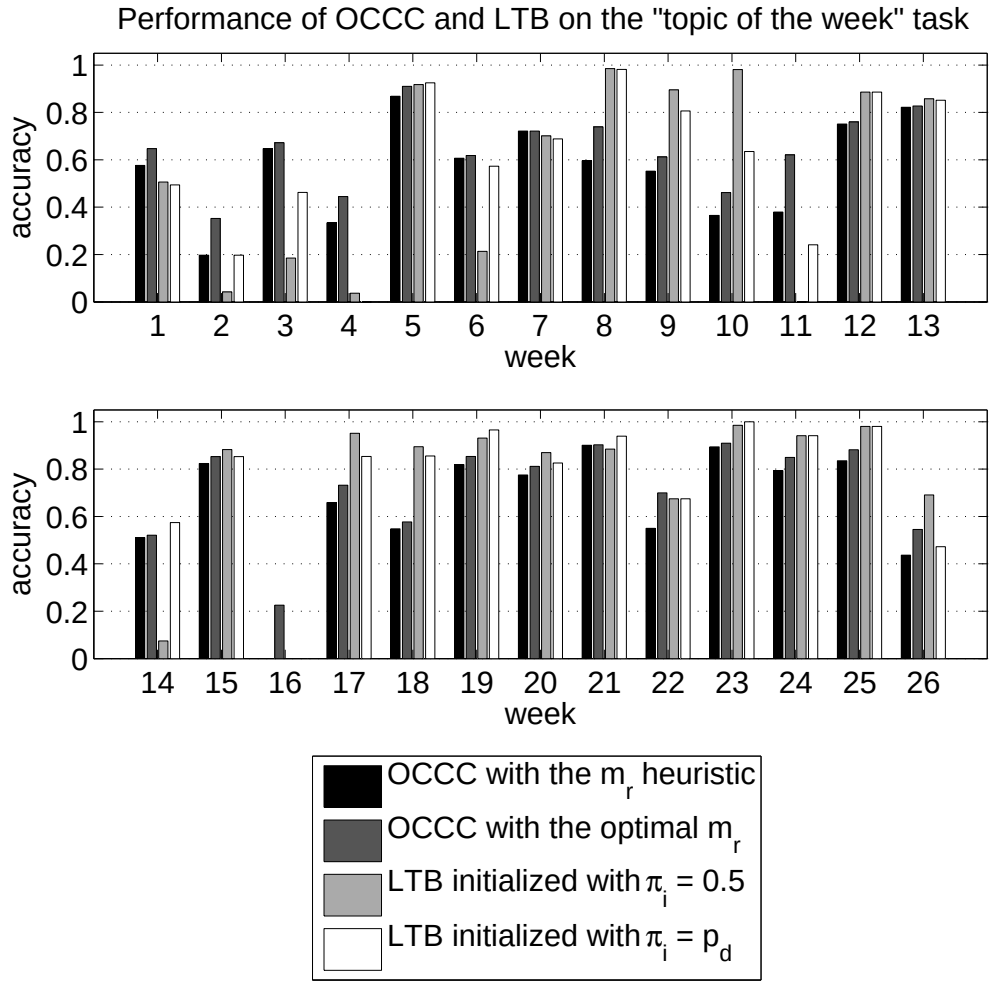


Figure 6.5. One-class clustering results on the "topic of the week" detection task.

One-class clustering accuracies per week are shown in Figure 6.5. These results reveal very interesting observations. First, OCCC methods tend to outperform LTB only on datasets where the results are quite low in general (less than 60% accuracy). Specifically, on weeks 2, 4, 11 and 16 the LTB models demonstrates extremely poor performance. While investigating this phenomenon, we discovered that in two of the four cases LTB was able to construct very clean core clusters, however, those clusters corresponded to the second largest topic rather than to the largest one. For example, on the week-4 data, topic #55077 (*"River ferry sinks on Bangladeshi river"*) was

Method	Accuracy
OCCC with the m_r heuristic	$61.4 \pm 4.5\%$
OCCC with optimal m_r	$68.3 \pm 3.6\%$
LTB initialized with $\pi_i = 0.5$	$65.3 \pm 7.3\%$
LTB initialized with $\pi_i = p_d$	$68.0 \pm 5.9\%$

Table 6.3. One-class clustering accuracy on the "topic of the week" detection task. The accuracies are macro-averaged over the 26 weekly data chunks. Standard error of the mean is presented after the ± sign.

discovered as the largest and the most coherent one. In that dataset, topic #55077 is represented by 20 documents, while topic #55063 (*"SARS Quarantined medics in Taiwan protest"*) is represented by 27 documents, such that topic #55077 is the second largest one. Another interesting observation is that the (completely unsupervised) LTB model can obtain very high results on some of the data chunks. For example, on weeks 5, 8, 19, 21, 23, 24, 25 LTB's accuracy is above 90%, with a striking 100% on week-23.

The one-class clustering accuracies, macro-averaged over the 26 weekly chunks, are presented in Table 6.3. As we can see, both LTB models outperform the OCCC variation where the m_r heuristic is applied. Moreover, even the optimal choice of m_r does not cause OCCC to perform significantly better than LTB. The dataset-dependent initialization of LTB's π_i parameters ($\pi_i = p_d$) appears to be preferable over the dataset-independent one ($\pi_i = 0.5$).

6.7 Summary

We have addressed the problem of inducing objective functions in Comraf models. For the task of one-class clustering, we proposed an information-theoretic and a probabilistic objective functions, as well as algorithms for their optimization. The proposed algorithms are very simple, very efficient and still surprisingly effective. More sophisticated algorithms (e.g. better optimization of the objective function in OCCC) are

emerging. Also, since the Comraf framework allows straightforward generalization of OCCC to one-class clustering with many modalities, it will be interesting to see whether one-class clustering results can be improved by adding more modalities, such as author names or hyperlinks.

Our evaluation of one-class clustering models on the re-ranking task is preliminary. It gives positive signals in the Web search case, where queries are of the general type and unlikely to be ambiguous. Also, one-class clustering is likely to be useful in Topic Detection and Tracking. However, our pilot experimentation in the ad-hoc retrieval domain shows rather negative results. In ad-hoc retrieval, our main assumption that the noise has no or little structure is generally wrong. For example, querying TREC 1 and 2 data for *acid rain*, the majority of 1000 retrieved documents are actually weather reports, most probably because all the other documents in the collection are even less relevant. Since one-class clustering methods do not take the query into account, and since the weather reports' subset *is* the largest and the most coherent one in the set of retrieved documents, our re-ranking hurts the ranking results on that query. Evaluating one-class clustering methods on other related tasks is the subject of our future work.

CHAPTER 7

IMAGE CLUSTERING WITH COMRAFS

In this chapter, we revise the Comraf clustering mechanism, proposed in Chapter 4. Based on the concept of *observed* combinatorial random variables (discussed in Chapter 5), we adapt the Comraf model to the case where the data consists of both sparse modalities (which need to be clustered) and dense modalities (not to be clustered). We also generalize the Comraf clustering objective function, making it more flexible and adjustable to a variety of real-world tasks. These two innovations finalize the development of the Comraf framework toward giving a comprehensive recipe for modeling with Comrafs, which we present in Chapter 8.

We focus here on multi-modal clustering of image collections, particularly of those where images are associated with textual captions.[1] Besides the caption words' modality, we consider visual modalities, both global (such as colors, texture) and local (regions, blobs). For details, see Section 7.4 below. Image clustering can be a useful component in a retrieval system [26], it can also be a stand-alone application, for example, for constructing semantic groups of image retrieval results [108], or for browsing image collections [5]. Unfortunately, existing uni-modal clustering methods often demonstrate poor performance on the image clustering task. In this chapter, we show that by employing the multi-modal learning paradigm we can significantly improve image clustering results.

Multi-modal clustering of images has an important difference when compared to multi-modal clustering of documents. Document features, such as words, POS tags

[1] A preliminary version of this work [12] was published at CVPR 2007.

etc. are situated in a discrete, finite space. Two textual features can be either identical or not. Some visual features, in contrast, are unique. These are local features, such as interest points [91], image regions [56] etc. An affinity metric should be defined to estimate similarity of those features. We find at least two disadvantages in working in the affinity space. First, the choice of the affinity metric is often arbitrary. Second, the affinity metric is defined for each *pair* of data points, which makes the computational complexity of related clustering algorithms quadratic in the best case. In this thesis, we aim at avoiding the explicit definition of the affinity metric (see Section 7.4.2).

7.1 Related work

The idea of clustering images using both low-level image features and surrounding text (i.e. grouping together visually similar and semantically related images) has attracted close attention of the research community. Barnard et al. [5] propose a generative hierarchical model for image clustering, in which every node generates words and blobs based on the given probability distributions for that node. Higher level nodes generate more general terms and lower level nodes generate more specific terms. The EM algorithm is used to fit the model. This approach can handle only two feature types (words, blobs); to handle more types, the model and the learning procedure must be revised.

Cai et al. [25] cluster Web image search results using visual, textual and link analysis. They extract text relevant to the image using a vision-based page segmentation algorithm. First, only text and hyperlink data is used to cluster images. The resulting clusters are clustered again using low-level image features. Loeff et al. [73] apply a similar approach: they calculate a histogram of gradient magnitude of the pixel values from every *interest point* and then cluster images using these local features with global color histograms and surrounding text. Both Cai et al. and Loeff et al. use spectral clustering methods (where the affinity scores for every pair of data instances

of every modality must be calculated), which are computationally infeasible in many other multi-modal applications.

Bipartite spectral graph partitioning [34] is useful for co-clustering two modalities such as documents and words. Gao et al. [47] extend this method to handle one more modality. In their tripartite graph model, nodes are arranged in three layers: words, images and image features. To handle more modalities, Gao et al. [48] propose another method that is most closely related to our work: they organize modalities in a *star structure* of interrelationships, where a central modality is connected to all the others. They treat this problem as fusion of multiple pairwise co-clustering problems. Each sub-problem is solved using the bipartite graph partitioning method.

Our approach has a few advantages over the others. First, our method has no practical limitation in the number of modalities as long as the pairwise interaction data is available—the addition of a modality increases the computational complexity only linearly. Second, our model can cluster multiple modalities while taking into account other modalities, which do not have to be clustered. Third, our information-theoretic clustering method does not rely on hard-to-obtain affinity matrices of individual modalities. Instead, easily computable contingency tables of interacting modalities are used. Overall, we propose a general framework for clustering multimedia collections, which can be straightforwardly applied to video data, sound tracks, hypertext etc. as well as to any of their combinations.

7.2 Multi-modal clustering objective, revisited

In Section 4.1, we proposed an objective function for multi-modal clustering as the sum of pairwise Mutual Information between interacting clusterings:

$$\mathbf{x}^{\mathbf{c}*} = \arg\max_{\mathbf{x}^{\mathbf{c}}} P(\mathbf{x}^{\mathbf{c}}) = \arg\max_{\mathbf{x}^{\mathbf{c}}} \sum_{(X_i^c, X_{i'}^c) \in \mathbf{E}} I(\tilde{X}_i; \tilde{X}_{i'}),$$

subject to $|\tilde{X}_i| = k_i$, where $i = 1, \ldots, m$. In Section 3.3 we discussed one disadvantage of a global objective function like that: Mutual Information terms can significantly vary in their magnitude, dependently on the support size of corresponding variables. Summing these terms together can cause an undesired effect of artificial preference of some interactions over the others. A natural generalization of this objective would be to consider a *weighted* linear combination of pairwise Mutual Information terms:

$$\mathbf{x}^{\mathbf{c}*} = \arg\max_{\mathbf{x}^{\mathbf{c}}} \sum_{(X_i^c, X_{i'}^c) \in \mathbf{E}} \beta_{ii'} I(\tilde{X}_i; \tilde{X}_{i'}), \tag{7.1}$$

where the weights $\beta_{ii'}$ are chosen using some domain knowledge. An obvious choice of the weights is such that all the Mutual Information terms are to be brought to the same scale. Another factor for choosing the weights is to make them correspond to various importance levels of various interactions. For example, if images are clustered based on their captions and their color histograms, the images/captions interaction can have a heavier weight than the weight of the images/colors interaction.

In some cases, weights $\beta_{ii'}$ can be adjusted during the course of an inference algorithm, in an *annealing* framework. Also, the weights can be learned using a model learning procedure. Both these extensions are left for our future work.

7.3 Comraf*: a lightweight version of the Comraf model

In previous chapters we made it obvious that, in most real-world situations, a practitioner is interested in clustering only one modality (images, in our case), which we call here a *target* modality. This implies that not every modality *has* to be clustered: if a representation of a modality is dense enough, clustering it may cause an underestimation of the joint (an effect known as *oversmoothing*), which may hurt clustering results of the target modality. For example, if images are distributed over

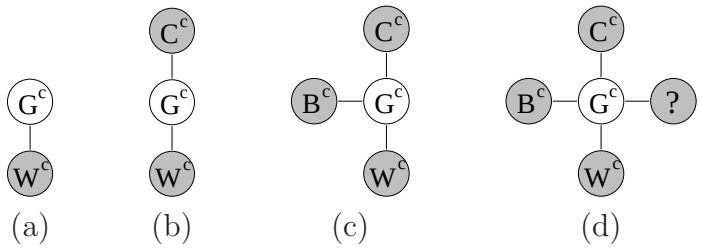


Figure 7.1. Comraf* models: (a) for images G^c and words W^c from their captions; (b) for images, words and colors C^c; (c) for images, words, colors and blobs B^c; (d) straightforward generalization to any number of modalities.

256 colors, it makes no sense to simultaneously cluster images and colors because the distributions are already dense enough.

In this section, we propose a special case of Comraf models, in which only the target modality is clustered, while the representations of all the other modalities are assumed to be dense enough. Each unclustered modality is associated with an *observed* combinatorial random variable. Recall that a combinatorial random variable is defined over all the possible clusterings of a given set. In case of unclustered modalities, the observed value of a corresponding combinatorial random variable is a clustering of *all singleton* clusters. For example, given a set $\{red, green, blue\}$, the observed value of a corresponding combinatorial random variable is $\{\{red\}, \{green\}, \{blue\}\}$.

Each observed combinatorial random variable of an unclustered modality is connected by an edge with a *hidden* combinatorial random variable of the target modality. Observed nodes are not connected to each other because they are statistically independent by definition. Hence, the resulting topology of the Comraf model is an *asterisk* with the target modality in the center. We call such a model *Comraf**. Examples of Comraf* graphs are given in Figure 7.1. Even though only one modality is clustered in Comraf*, it is still a model for *multi-modal* clustering, as multiple modalities are involved in the clustering process.

Recall that in Chapter 4 we considered Comraf models for multi-modal clustering where *each* modality should be clustered. In Comraf* models only *one* modality is clustered. The general Comraf model, however, takes care of any number of dense *and* sparse random variables. In Section 7.4.2 we present a Comraf model for simultaneously clustering images and their local features, while incorporating other (unclustered) modalities. Since the simultaneous clustering can be computationally hard, we also show how to reduce the computational burden by translating such a Comraf model into a series of Comraf* models, each of which is then optimized separately.

7.3.1 Inference in Comraf*

In Comraf*, where all the edges are attached to X_0^c and all the leaves are observed combinatorial random variables, Equation (7.1) is transformed into:

$$x_0^{c*} = \arg\max_{\mathbf{x^c}} \sum_{i=1}^{m-1} \beta_i I(\tilde{X}_0; \tilde{X}_i) = \arg\max_{\mathbf{x^c}} \sum_{i=1}^{m-1} \beta_i I(\tilde{X}_0; X_i), \tag{7.2}$$

since $\tilde{X}_i = X_i$ for the unclustered modalities. As always, we have the $|\tilde{X}_0| = k$ constraint.

To compute the weighted sum of pairwise mutual information from Equation (7.2), the following procedure is used. The input of the procedure is an (empirical) joint distribution $P(X_0, X_i)$ of the underlying data of each interacting pair (X_0^c, X_i^c). For a given partitioning x_0^c, the distribution $P(\tilde{X}_0, X_i)$ is computed using the cumulative rule $P(\tilde{x}_0; x_i) = \sum_{x_0 \in \tilde{x}_0} P(x_0, x_i)$. Marginals $P(\tilde{X}_0)$ and $P(X_i)$ are obtained through the marginalization $P(\tilde{x}_0) = \sum_{x_i} P(\tilde{x}_0, x_i)$ and $P(x_i) = \sum_{\tilde{x}_0} P(\tilde{x}_0, x_i)$. Now we have all the ingredients to calculate the mutual information:

$$I(\tilde{X}_0; X_i) = \sum_{\tilde{x}_0, x_i} P(\tilde{x}_0, x_i) \log \frac{P(\tilde{x}_0, x_i)}{P(\tilde{x}_0)P(x_i)}.$$

To perform an inference in Comraf*, we apply a version of our MDC algorithm (see Section 4.3), where either top-down or bottom-up clustering procedures is used for clustering X_0. In the top-down procedure, we start with one cluster that contains all the values of X_0 and split it until the required number of clusters is obtained (while interleaving with the optimization routine). In bottom-up clustering, we start with all singleton clusters and merge them until, again, reaching the required number of clusters.

The computational complexity of the top-down algorithm is $O(l|X_0|\sum_{j=1}^{m-1}|X_i|)$, and of the bottom-up algorithm $O(l|X_0|^2\sum_{j=1}^{m-1}|X_i|)$, where l is a (fixed) number of clustering iterations. Note that an arbitrary number of leaves (unclustered modalities) can be incorporated into the Comraf* model, while adding new modalities increases the complexity only linearly.

7.4 Modalities of an image collection

In this work, along with images, we consider three other modalities. The first one is words from image captions. We remove stopwords and apply a simple 's'-stemming (removal of plural suffixes). A joint probability of an image g and a word w is $P(g,w)=\frac{N_{w\in g}}{|W|}$, where $N_{w\in g}$ is the number of occurrences of w in g's caption, $|W|$ is the total number of words. Another modality is colors appearing in images. The joint probability distribution of colors and images is obtained from color histograms, as a number of pixels of color c in image g divided by the total number of pixels in all images. The third modality is blobs, as described below.

7.4.1 Rectangular blobs

Blobs (or *visual terms*) are a special type of image content representation based on a *fixed vocabulary*. To generate blobs, images are first segmented into regions, which are then clustered across all images. Blobs are the resulting region clusters. Each

image is mapped onto the set of blobs which leads to in a representation analogous to the bag-of-words (BOW) in text processing.

Barnard and Forsyth [6] and Duygulu et al. [37] segment images into semantically coherent regions using Blobworld and Normalized-Cuts algorithms, respectively. Unfortunately, these algorithms do not always produce segmentations accurate enough for further use. Jeon and Manmatha [56] and Feng et al. [41] use a rectangular grid to segment images and report better results on an image retrieval task. We apply the same set of blobs as in [41], built using the following procedure. Images are first segmented to regions using a 6×4 grid. Then, for each region, a feature vector is constructed that contains texture and color information: Gabor texture filters with 4 orientations and 3 scales are used to construct 12 dimensional texture features; the mean, standard deviation and skewness of RGB and LAB components are computed to build 18 dimensional color features. The resulting 30 dimensional feature vectors are clustered using k-means.

7.4.2 Blobs constructed by Comraf models

As discussed in Section 7.4.1 above, a clustering process is involved in constructing blobs from rectangular regions, represented by color and texture features. Naturally, since Comrafs are models for multi-modal clustering, an intrinsic Comraf model can be used for simultaneously clustering images and their regions. Co-clustering of images and features has been recently described in literature [89], however, Comrafs have an additional power over co-clustering methods: Comrafs can incorporate multiple modalities, both sparse (that are to be clustered) and dense (that are not).

Figure 7.2 (left) shows a Comraf model for clustering images G simultaneously with their regions R, taking into account color C and texture T information of the regions, as well as the colors and caption words W of the images. Obviously, more edges and nodes can be added to the model, depending on the data's availability.

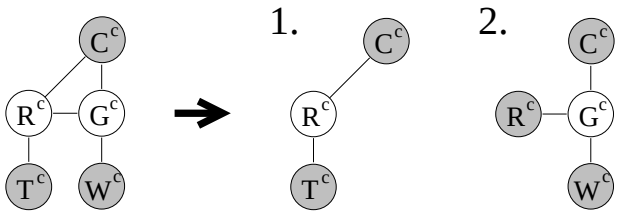


Figure 7.2. (left) A Comraf model for simultaneously clustering images G^c and their rectangular regions R^c, while taking into account words W^c from image captions, colors C^c and texture data T^c; (right) a translation of this model into a two-step Comraf*: the first Comraf* is for clustering regions into blobs, whereas the second Comraf* is for clustering images based on these blobs.

In Section 7.3.1 we mentioned that the input of a Comraf inference procedure is a set of pairwise probability tables $P(X_i, X_{i'})$ for each edge in the Comraf graph. An interesting case is the (G^c, R^c) edge between image and region combinatorial random variables in Figure 7.2 (left). Unlike colors and caption words, each region is unique, so for each region r and each image g, their joint probability is $P(r, g) = \frac{1}{|R|}$ if $r \in g$, and 0 otherwise (where $|R|$ is the total number of regions in the dataset). Such a probability mass function is useless for clustering regions, because only regions that belong to the same image can be clustered together. A possible way to resolve this problem would be to estimate this probability by giving a portion of its mass to $P(r, g)$ even if $r \notin g$. Such an estimation can be made based on computing an affinity metric between regions of various images, which is computationally hard: $O(|R|^2|r|)$, where $|r|$ is the size of any region.

Comrafs offer an elegant solution to this problem: since regions are clustered *not only* based on images, but also based on colors and texture, neither of which has this problem, we still can use our objective function from Equation (7.1). As long as images are clustered in parallel with regions, Equation (7.1) allows grouping together regions that belong to the same image *cluster*, as desired. Therefore, we apply the Comraf model from Figure 7.2 (left) as it is. We choose to cluster images bottom-up

and regions top-down. In our objective (7.1), we cope with the fact that $I(\tilde{R};\tilde{G})$ is two orders of magnitude larger than $I(\tilde{R};\tilde{C})$ and $I(\tilde{R};\tilde{T})$, by setting the weights of the latter two terms to 100.

The simultaneous clustering of images and regions is a time consuming process: its complexity is $O(|R|\ |G|\ (|C|+|T|+|W|))$. We propose a light-weight version of this model, in which inference is done in *two steps*: first, regions are clustered based on their color and texture features, and then images are clustered based on colors, caption words and region clusters. Such a model is equivalent to two Comraf* models applied one after another, as presented in Figure 7.2 (right). This model's complexity is plausible: $O(|R|\ (|C|+|T|)+|G|\ (|C|+|\tilde{R}|+|W|))$, where $|\tilde{R}|$ is the number of region clusters. Moreover, in Section 7.5.2 we show that the performance of the two-step Comraf* is plausible as well: on one of our datasets, it obtains clustering accuracy comparable to the one of a general Comraf. Generalizing the two-step setting, it is easy to see that any Comraf can be translated into a series of Comraf* models.

7.5 Experimentation

We experiment with a variety of particular Comraf* models (see examples in Figure 7.1), as well as with the general Comraf models from Figure 7.2. The experiments are conducted using our open-source Comraf clustering tool.[2] In all our models, images are clustered agglomeratively. All our results are averaged over 10 independent runs, with the standard error reported. As a baseline, we use the k-means algorithm (`SimpleKMeans` implementation of WEKA[3]), where images are represented as BOW of their captions. Also, our 2-node Comraf* model is equivalent to the hard-clustering version of Information Bottleneck (IB) [106] (see Section 4.1 for discussion), hence

[2] `http://sourceforge.net/projects/comraf`

[3] `http://cs.waikato.ac.nz/ml/weka`

Category	# of images	Category	# of images
Birds	152	Christianity	191
Desert	172	Islam	96
Flowers	165	Judaism	187
Trees	190	Personalities	188
Food	187	Symbols	130
Housing	165	OVERALL:	1823

Table 7.1. Categories (and their sizes) of the IsraelImages dataset.

we use it as our baseline as well. For evaluation of our clustering results, we use micro-averaged *accuracy* (Section 4.6.1) of the constructed image clustering.

7.5.1 Datasets

We demonstrate the performance of our clustering methods on two datasets: a subset of the benchmark Corel dataset and a new multimedia dataset, which we refer to as *IsraelImages*, collected by us especially for this work.

The Corel subset[4] has already been used in various previous research projects [37, 55, 41]. The dataset consists of 5,000 images from 50 Corel Stock Photo CDs. Each CD contains 100 images on the same topic, such as "Sunrises and Sunsets", "Mountains of America" and "Wild Animals". Every image has a caption and an annotation. The caption is a brief description of the scene and the annotation is a list of objects that appear in the image. An example of an image caption is "Man And Boy Fishing Mountain Lake", while "Tree People Mountain Water" is an annotation for this image. Overall 371 words are used to annotate the collection. The original dataset has 4,500 training images and 500 test images. Since our model does not require training, we use 4,500 training images for our experiments and save the remaining 500 images for future use.

[4] `http://kobus.ca/research/data/eccv_2002`

Method	Accuracy
k-means: images over caption words	22.0%
IB: images/caption words	$44.2 \pm 1.0\%$
IB: images/colors	$24.4 \pm 0.2\%$
Comraf*: images/words/colors	$54.2 \pm 0.9\%$
General Comraf: Figure 7.2 (left)	$68.6 \pm 1.0\%$
Two-step Comraf*: Figure 7.2 (right)	$\mathbf{69.0 \pm 0.6}\%$

Table 7.2. Micro-averaged clustering accuracy on IsraelImages. All IB/Comraf results are averaged over 10 independent runs with the standard error of the mean reported after the '±' sign.

The second dataset consists of 1823 images downloaded from `IsraelImages.com`. The images reflect different aspects of Israel scenery and/or society and are grouped into 11 categories (see Table 7.1). Each image is 375 by 250 pixels and has a 1 to 18 word long caption. This dataset is available to the research community.[5]

7.5.2 Comparative results

Our results on the IsraelImages dataset are reported in Table 7.2. Adding the color modality to the caption BOW improves the clustering result by 10% (on an absolute scale), whereas adding the regions (in a 2-step Comraf* scheme) leads to an additional 15% improvement. These findings demonstrate the value of multi-modal setting in image clustering. The general Comraf model from Figure 7.2 (left) is not able to outperform the 2-step Comraf*. This is probably due to the fact that color and texture information is more important for clustering regions than the correspondence between regions and image clusters.

We also experiment with various levels of color granularity in a 3-node Comraf* setting (from Figure 7.1b)—the results are presented in Figure 7.3 (left). As can be seen, if the color information is detailed enough (above 216 colors), the difference in the results is statistically insignificant. Figure 7.3 (center) shows the results of the

[5] `http://www.cs.umass.edu/~ronb/image_clustering.html`

Method	Accuracy
k-means: images over caption words	22.0%
IB: images/caption words	$46.6 \pm 0.5\%$
IB: images/colors	$22.5 \pm 0.2\%$
IB: images/blobs (see Section 7.4.1)	$24.7 \pm 0.3\%$
Comraf*: images/words/colors	$55.3 \pm 0.5\%$
Comraf*: images/words/blobs	$59.4 \pm 0.5\%$
Comraf*: images/words/colors/blobs	$60.1 \pm 0.3\%$
Two-step Comraf*: Figure 7.2 (right)	$\mathbf{61.2 \pm 0.4}\%$
IB: images/annotation words	$58.6 \pm 0.3\%$

Table 7.3. Micro-averaged clustering accuracy on Corel. All IB/Comraf results are averaged over 10 independent runs with the standard error of the mean reported after the '±' sign.

2-step Comraf* over various numbers of colors for clustering regions. Generally, less colors are needed for clustering regions than for clustering images: 216 colors appear to be too many.

A summary of our results on the Corel dataset is presented in Table 7.3. It shows surprisingly similar trends as for IsraelImages. On a 3-node setup with caption words and blobs we obtain 59.4% accuracy, which is especially impressive given that a random assignment of images into 50 clusters would lead to 2% accuracy (our result is 30 times above random). Adding the color modality improves this result only insignificantly (as expected, since blobs already incorporate the color information, among with texture). The success of 3-node and 4-node Comraf* clustering models is also supported by the fact that they outperform a 2-node *supervised* clustering model, in which images are clustered with respect to their annotations assigned by human experts.

The 2-step Comraf* shows some further (minor) improvement over the 1-step Comraf* models. Here, in contrast to IsraelImages, 8 colors are enough for clustering regions, and adding more colors causes a significant drop in the performance. We suspect that the Corel dataset is "too simple": it contains many images that are

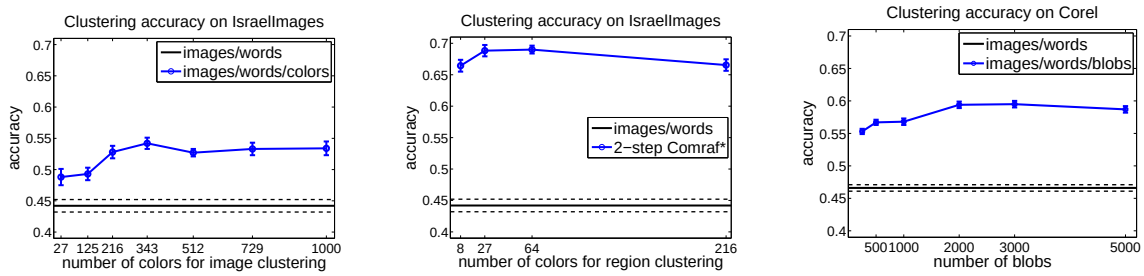


Figure 7.3. Experimentation with various numbers of: (left) colors on IsraelImages in a 3-node images/words/colors Comraf*; (center) colors for clustering regions in the 2-step Comraf* on IsraelImages; (right) blobs on Corel in a 3-node images/words/blobs Comraf*. Our baseline is the 2-node images/words clustering result. Left and right graphs show the same trend: after reaching a certain number of colors (256) or blobs (2000), the results vary only insignificantly. The central graph, however, shows that too many colors for clustering regions can hurt.

almost identical to each other, therefore more advanced clustering models lead to no (or minor) gain over the simpler ones.

Analogously to our IsraelImages experiment with various sizes of color sets, we test various numbers of blobs on Corel. In previous work [37, 55], the number of blobs is set to 500, to (roughly) correspond to the number of annotation keywords. Here we show that 500 blobs are not enough for clustering: when moving from 1000 to 2000 blobs, a significant boost in the system's performance can be seen.

Figures 7.4 and 7.5 are illustrations of the quality of multi-modal setup: unrelated groups of images are mixed together when the clustering is based only on caption words, whereas they are nicely separated when a visual modality is added.

7.6 Summary

In this chapter, we have introduced the Comraf framework for clustering multimedia collections. We have also proposed a family of lightweight Comraf models called Comraf*, which demonstrate excellent performance on clustering two real-world image collections. To further improve the image clustering results, a semi-supervised

Comraf setting (see Chapter 5) can be used, in which a few labeled examples are taken into account in the clustering process. We plan to experiment with this setting in our future work.

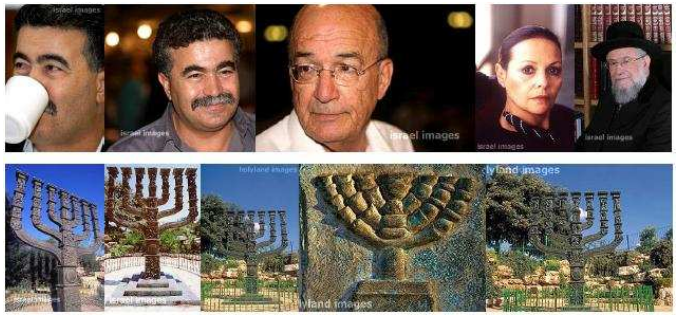
Designing general Comraf models for image clustering (in the flavor of the model shown in Figure 7.2 left) is an ongoing process. Extensive experimentation will lead to discovering the optimal Comraf setting for clustering multimedia collections.

(a) Clustering results using only caption words, Corel dataset

(b) Clustering results using words and blobs, Corel dataset

Figure 7.4. Corel dataset. The first row shows clustering results using only words. Swimmers and swimming tigers are clustered together because they share common terms like "water" and "swim". The second and the third rows show clustering results using both words and blobs. The swimmers and the swimming tigers are now in two different clusters with other similar images.

(a) Clustering results using only caption words, IsraelImages dataset



(b) Clustering results using words and color histograms, IsraelImages dataset

Figure 7.5. IsraelImages dataset. People portraits and pictures of the menorah monument are clustered together using caption words because they have a word 'Knesset' (the Israeli parliament) in common: the individuals are Knesset members, while the menorah monument is placed in front of the Knesset building. The problem is resolved after the color modality is added.

CHAPTER 8

CONCLUSION

In this thesis we have introduced Combinatorial Markov Random Fields (Comrafs)—a novel, generic framework for statistical modeling that consists of three basic components:

1. An undirected graph with nodes being statistical objects of "rich" structure and edges being interactions between those objects;

2. An objective function (either probabilistic or non-probabilistic) factored over the graph;

3. A method for optimizing the objective.

We have applied the Comraf framework to *multi-modal learning*, which is a learning problem in the environment where multiple views (or *modalities*) of the data are available. Based on the material presented in previous chapters, we can give an ultimate recipe for solving multi-modal learning problems with Comrafs:

1. Come up with a few modalities for a particular dataset. In most cases, it is easy to come up with two modalities: one for data instances, another for their features. Once a few data types or feature types are available, they can be represented as modalities. Note that a modality is a *set* over which a probability distribution can be defined. Comrafs are unlikely to be useful in cases where data instances are represented as feature vectors, where each feature is intrinsically different from the others (e.g. where feature vectors consist of four

features: *color*, *size*, *temperature*, and *price*, such that it is difficult to define a probability distribution over this feature set).

2. Decide which modalities interact with each other. This decision should be made upon availability of contingency information for each pair of modalities, as well as based on domain knowledge (e.g. whether or not it is *natural* to see these modalities interacting). For example, say we are given three modalities *images*, their *colors*, and their *caption words*. Captions definitely interact with images, as well as colors interact with images. However, we can assume that captions do not interact with colors, as it is a very rare case where captions directly describe colors in an image. Note that a decision about presence / absence of interactions is analogous to defining conditional independencies in other types of graphical models: the number of interactions should be kept as low as possible in order to keep the model tractable.

3. For a particular learning task, decide which modalities should be optimized and which should be *observed*. Observed modalities usually provide some level of supervision to the model: using observed modalities, prior knowledge can be represented. Also, a modality can be observed if its size is very small, such that a distribution defined over this modality is statistically dense.

4. Represent those modalities that are to be optimized as hidden *combinatorial random variables*, and those that are not as observed combinatorial random variables. A combinatorial r.v. can be defined over a set of possible partitionings, subsets, partial orderings of a modality, or over other types of combinatorial sets, according to the particular problem being solved.

5. Represent each combinatorial r.v. as a node in a graph in which undirected edges correspond to interactions. We now have finished building the *Comraf graph*.

6. Represent the learning task as optimization of an objective function that is defined over nodes and edges in the Comraf graph. Choose the objective function that best suits the task. The choice of objective function can be made based on previously published work in the field (as in Chapter 4), or based on theoretical analysis (as in Chapter 6), as well as based on some pilot research or other considerations.

7. Since optimizing the objective function simultaneously over the entire Comraf graph appears to be intractable, propose a method for traversing the Comraf graph in order to perform *iterative* optimization of the objective. In this thesis, we have discussed two such methods: Iterative Conditional Modes (ICM) and Clique-wise Optimization (CWO)—see Section 3.3. ICM can be considered a global optimization method, as the objective is optimized at each node conditionally on the rest of the model. In contrast, CWO is a local optimization method, as the objective is optimized over each clique independently of the rest of the model.

8. At each node / clique, apply a combinatorial optimization method for optimizing the objective. In Section 4.3, we proposed two simple and greedy combinatorial optimization methods (sequential and shuffled), both of which explore the local neighborhood of an initial configuration. More sophisticated methods, such as Branch and Bound, can be used as well.

9. As the global optimum of the objective function is unlikely to be found, propose a stopping criterion of the optimization procedure. In our experimentations with multi-modal clustering, we halted the optimization procedure as soon as the desired number of clusters was achieved.

We applied the proposed framework to multi-modal clustering (Chapter 4), semi-supervised learning (Chapter 5), and one-class clustering (Chapter 6). Both text and

image domains (Chapter 7) were explored. Three important issues have not been addressed in this thesis—we leave them for our future work:

- **Multi-modal ranking**, which is another application of the Comraf framework. In the multi-modal ranking problem, one simultaneously ranks a number of modalities, given rankings of the other modalities. One example of multi-modal ranking comes from the data mining / collaborative filtering area: given a ranking of movies, the task is to simultaneously rank its directors and actors who starred in those movies. The goal of such system would be to adequately measure popularity of celebrities. Another example comes from image retrieval: given a ranked list of documents, retrieved on a certain query, the task is to simultaneously rank images in these documents and their local features (blobs or interest points). Our intuition here is that the simultaneous ranking would improve the quality of image ranked lists. Note that the layout of Comraf graphs for multi-modal ranking is the same as for multi-modal clustering. However, the objective function and optimization method should be specific for the multi-modal ranking task.

- **Scalability issue in Comrafs.** Unfortunately, the current version of our MDC implementation for multi-modal clustering is very slow. Given that each optimization iteration is repeated ten times (i.e. ten random restarts), a straightforward enhancement would be to perform those random restarts in parallel on ten machines. Another possible enhancement would be to limit the search length in the shuffled version of MDC, or to parallelize the shuffling steps using the MapReduce paradigm [33].

- **Model learning in Comrafs.** It turns out that the main factor for achieving good clustering results with Comraf models is the good choice of modalities and

their interactions. It is desired to construct a system that could a priori decide whether the available modalities would be helpful or harmful.

In addition to being a useful framework for multi-modal learning, Comrafs can go beyond it: Comraf nodes do not necessarily have to represent data modalities. Also, random variables of rich structure may not necessarily be of the combinatorial nature, so Comrafs have good potential for a generalization into a new framework. We call it *Non-Bayesian Networks (NBN)*. Development of such a framework is also the subject of our future work. An interesting question yet to be answered is how to model NBN's nodes (which are structurally rich) in a finer-grained manner. A possible answer is to use a lower-level NBN at each node of the (upper-level) NBN, by which we build a *telescopic model*. Constructing such a model would resemble designing an object-oriented software system, which has a direct connection with the power framework of Object-Oriented Bayesian Networks [61]. Developing Object-Oriented Non-Bayesian Networks would be the final goal of this research.

To conclude, the contributions of this thesis are:

1. Proposing Comrafs—a novel framework for statistical modeling that brings together two research fields: graphical models and combinatorial optimization.

2. Applying this framework to a variety of problems in multi-modal learning, such as multi-modal clustering, semi-supervised clustering, interactive clustering, one-class clustering etc.

3. Proposing model layouts, objective functions and optimization procedures for each of these problems.

4. Showing empirical advantage of Comrafs over previous state-of-the-art methods on various real-world tasks, such as Web appearance disambiguation, document clustering by genre and author's sentiment, organization of image galleries etc.

BIBLIOGRAPHY

[1] Aggarwal, C. C., and Yu, P. S. Outlier detection for high dimensional data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data* (2001), pp. 37–46.

[2] Allan, J., Ed. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, 2002.

[3] Argamon, S., Koppel, M., Fine, J., and Shimoni, A. R. Gender, genre, and writing style in formal written texts. *Text 23*, 3 (2003).

[4] Bagga, A., and Baldwin, B. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of COLING-ACL-17* (1998), pp. 79–85.

[5] Barnard, K., Duygulu, P., and Forsyth, D. A. Clustering art. In *Proceedings of CVPR* (2001), pp. 434–441.

[6] Barnard, K., and Forsyth, D. A. Learning the semantics of words and pictures. In *Proceedings of ICCV-8* (2001), pp. 408–415.

[7] Basu, S., Banerjee, A., and Mooney, R. J. Semi-supervised clustering by seeding. In *Proceedings of ICML-19* (2002), pp. 19–26.

[8] Basu, S., Bilenko, M., and Mooney, R. J. A probabilistic framework for semi-supervised clustering. In *Proceedings of SIGKDD-10* (2004), pp. 59–68.

[9] Bekkerman, R., Eguchi, K., and Allan, J. Unsupervised non-topical classification of documents. Tech. Rep. IR-472, Center of Intelligent Information Retrieval, UMass Amherst, 2006.

[10] Bekkerman, R., El-Yaniv, R., and McCallum, A. Multi-way distributional clustering via pairwise interactions. In *Proceedings of ICML-22* (2005), pp. 41–48.

[11] Bekkerman, R., El-Yaniv, R., Tishby, N., and Winter, Y. Distributional word clusters vs. words for text categorization. *Journal of Machine Learning Research 3* (2003), 1183–1208.

[12] Bekkerman, R., and Jeon, J. Multi-modal clustering for multimedia collections. In *Proceedings of CVPR-07, the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2007).

[13] Bekkerman, R., and McCallum, A. Disambiguating web appearances of people in a social network. In *Proceedings of WWW-14* (2005).

[14] Bekkerman, R., McCallum, A., and Huang, G. Automatic categorization of email into folders: benchmark experiments on Enron and SRI corpora. Tech. Rep. IR-418, CIIR, UMass Amherst, 2004.

[15] Bekkerman, R., Raghavan, H., Allan, J., and Eguchi, K. Interactive clustering of text collections according to a user-specified criterion. In *Proceedings of IJCAI-20* (2007).

[16] Bekkerman, R., and Sahami, M. Semi-supervised clustering using combinatorial MRFs. In *Proceedings of ICML-23 Workshop on Learning in Structured Output Spaces* (2006).

[17] Bekkerman, R., Sahami, M., and Learned-Miller, E. Combinatorial Markov Random Fields. In *Proceedings of ECML-17* (2006).

[18] Bekkerman, R., Zilberstein, S, and Allan, J. Web page clustering using heuristic search in the web graph. In *Proceedings of IJCAI-20* (2007).

[19] Besag, J. Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society 36*, 2 (1974), 192–236.

[20] Besag, J. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society 48*, 3 (1986).

[21] Bickel, S., and Scheffer, T. Multi-view clustering. In *Proceedings of ICDM'04, the 4th IEEE International Conference on Data Mining* (2004).

[22] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet allocation. *Journal of Machine Learning Research 3* (2003), 993–1022.

[23] Bouvrie, J. Multi-source contingency clustering. Master's thesis, EECS, MIT, 2004.

[24] Burnard, L. User reference guide for the British National Corpus. Tech. rep., Oxford University Computing Services, 2000.

[25] Cai, D., He, X., Li, Z., Ma, W.-Y., and Wen, J.-R. Hierarchical clustering of WWW image search results using visual, textual and link information. In *Proceedings of the 12th international conference on Multimedia* (2004), pp. 952–959.

[26] Chen, Y., Wang, J. Z., and Krovetz, R. Clue: Cluster-based retrieval of images by unsupervised learning. *IEEE Transactions on Image Processing 14*, 8 (2005), 1187–1201.

[27] Chickering, D. M. Optimal structure identification with greedy search. *Journal of Machine Learning Research 3* (2003), 507–554.

[28] Crammer, K., and Chechik, G. A needle in a haystack: local one-class optimization. In *Proceedings of ICML-21* (2004).

[29] Cristianini, Nello, and Shawe-Taylor, John. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[30] D. Aldous, D. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII 1983*. Springer, 1985, pp. 1–198.

[31] Dayanik, A., Lewis, D. D., Madigan, D., Menkov, V., and Genkin, A. Constructing informative prior distributions from domain knowledge in text classification. In *Proceedings of SIGIR-29* (2006).

[32] de Sa, V.R. *Unsupervised Classification Learning from Cross-Modal Environm ental Structure*. PhD thesis, University of Rochester, 1994.

[33] Dean, J., and Ghemawat, S. Mapreduce: simplified data processing on large clusters. In *Proceedings of Symposium on Operating System Design and Implementation* (2004).

[34] Dhillon, I. S. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of SIGKDD-7* (2001), pp. 269–274.

[35] Dhillon, I. S., Mallela, S., and Modha, D. S. Information-theoretic co-clustering. In *Proceedings of SIGKDD-9* (2003), pp. 89–98.

[36] Diaz, F. Regularizing ad hoc retrieval scores. In *Proceedings of CIKM-05, the 14th ACM international conference on Information and knowledge management* (2005), pp. 672–679.

[37] Duygulu, P., Barnard, K., de Freitas, N., and Forsyth, D. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of ECCV-7* (2002).

[38] Eguchi, K., and Lavrenko, V. Sentiment retrieval using generative models. In *Proceedings of EMNLP* (2006).

[39] El-Yaniv, R., and Souroujon, O. Iterative double clustering for unsupervised and semi-supervised learning. In *Advances in Neural Information Processing Systems (NIPS-14)* (2001).

[40] Fayyad, U. M., Reina, C., and Bradley, P. S. Initialization of iterative refinement clustering algorithms. In *Proceedings of SIGKDD-4* (1998), pp. 194–198.

[41] Feng, S. L., Manmatha, R., and Lavrenko, V. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of CVPR* (2004), pp. 1002–1009.

[42] Finn, A., Kushmerick, N., and Smyth, B. Genre classification and domain transfer for information filtering. *Advances in Information Retrieval LNCS 2291* (2002).

[43] Fleischman, M. B., and Hovy, E. Multi-document person name resolution. In *Proceedings of ACL-42, Reference Resolution Workshop* (2004).

[44] Forman, G. Quantifying trends accurately despite classifier error and class imbalance. In *Proceedings of SIGKDD-12* (2006), pp. 157–166.

[45] Freeman, W. T., Pasztor, E. C., and Carmichael, O. T. Learning low-level vision. *IJCV 40*, 1 (2000), 25–47.

[46] Friedman, N., Mosenzon, O, Slonim, N., and Tishby, N. Multivariate information bottleneck. In *Proceedings of UAI-17* (2001).

[47] Gao, B., Liu, T.-Y., Qin, T., Zheng, X., Cheng, Q.-S., and Ma, W.-Y. Web image clustering by consistent utilization of visual features and surrounding texts. In *Proceedings of the 13th international conference on Multimedia* (2005).

[48] Gao, B., Liu, T.-Y., Zheng, X., Cheng, Q.-S., and Ma, W.-Y. Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering. In *Proceeding of SIGKDD-11* (2005), pp. 41–50.

[49] Gooi, C. H., and Allan, J. Cross-document coreference on a large scale corpus. In *Proceedings of HLT/NAACL* (2004).

[50] Gupta, G., and Ghosh, J. Robust one-class clustering using hybrid global and local search. In *Proceedings of ICML-22* (2005), pp. 273–280.

[51] Han, H., Giles, L., Zha, H., Li, C., and Tsioutsiouliklis, K. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of JCDL-4* (2004).

[52] Havre, S., Hetzler, E., Whitney, P., and Nowell, L. Themeriver: Visualizing thematic changes in large document collections. *IEEE Transactions on Visualization and Computer Graphics 8*, 1 (2002), 9–20.

[53] Huang, Y., and Mitchell, T. Text clustering with extended user feedback. In *Proceedings of SIGIR-29* (2006), pp. 413–420.

[54] Jakulin, A., and Bratko, I. Testing the significance of attribute interactions. In *Proceedings of ICML-21* (2004).

[55] Jeon, J., Lavrenko, V., and Manmatha, R. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of SIGIR-26* (2003), pp. 119–126.

[56] Jeon, J., and Manmatha, R. Using maximum entropy for automatic image annotation. In *Proceedings of the 5th International Conference on Image and Video Retrieval* (2004), pp. 24–32.

[57] Jordan, M. I. Graphical models. *Statistical Science 19* (2004), 140–155.

[58] Jordan, M. I., and Weiss, Y. *Graphical models: Probabilistic inference.* MIT Press, 2002. In M. Arbib (ed.), The Handbook of Brain Theory and Neural Networks.

[59] Karlgren, J., and Cutting, D. R. Recognizing text genres with simple metrics using discriminant analysis. In *Proceedings of the 15th International Conference on Computational Linguistics* (1994), pp. 1071–1075.

[60] Kessler, B., Nunberg, G., and Schütze, H. Automatic detection of text genre. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics* (1997), pp. 32–38.

[61] Koller, D., and Pfeffer, A. Object-oriented Bayesian networks. In *Proceedings of UAI-13* (1997), pp. 302–313.

[62] Koontz, W. L. G., Narendra, P. M., and Fukunaga, K. A branch and bound clustering algorithm. *IEEE Transactions on Computers 24* (1975).

[63] Koppel, M., and Shimoni, A. R. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing 17*, 4 (2002), 401–412.

[64] Kumaran, G., Jones, R., and Madani, O. Biasing web search results for topic familiarity. In *Proceedings of the ACM 14th Conference on Information and Knowledge Management* (2005), pp. 271–272.

[65] Kurland, O., and Lee, L. Pagerank without hyperlinks: structural re-ranking using links induced by language models. In *Proceedings of the 28th annual international ACM SIGIR conference* (2005), pp. 306–313.

[66] Lafferty, J., McCallum, A., and Pereira, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML-18* (2001), pp. 282–289.

[67] Land, A. H., and Doig, A. G. An automatic method for solving discrete programming problems. *Econometrica 28* (1960), 497–520.

[68] Lebanon, G. *Riemannian Geometry and Statistical Machine Learning.* PhD thesis, CMU, 2005.

[69] LeCun, Y., and Huang, F. J. Loss functions for discriminative training of energy-based models. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (2005), pp. 206–213.

[70] Lee, D. Y.-W. Genres, registers, text types, domains and styles: clarifying the concepts and navigating a path through the bnc jungle. *Language Learning & Technology 5*, 3 (2001), 37–72.

[71] Lee, Y.-B., and Myaeng, S. H. Text genre classification with genre-revealing and subject-revealing features. In *Proceedings of the 25th International ACM SIGIR Conference* (2002), pp. 145–150.

[72] Liu, B., Li, X., Lee, W. S., and Yu, P. S. Text classification by labeling words. In *Proceedings of AAAI-19* (2004), pp. 425–430.

[73] Loeff, N., Alm, C. O., and Forsyth, D. A. Discriminating image senses by clustering with multimodal features. In *Proceedings of COLING/ACL* (2006), pp. 547–554.

[74] Mann, G. S., and Yarowsky, D. Unsupervised personal name disambiguation. In *Proceedings of CoNLL-7* (2003), pp. 33–40.

[75] Marinescu, R., and Dechter, R. AND/OR branch-and-bound for graphical models. In *Proceedings of IJCAI-19* (2005).

[76] Matsuo, Y., Mori, J., Hamasaki, M., Ishida, K., Nishimura, T., Takeda, H., Hasida, K., and Ishizuka, M. Polyphonet: an advanced social network extraction system from the web. In *Proceedings of WWW-15, the 15th international conference on World Wide Web* (2006), pp. 397–406.

[77] Matthews, R. A. J., and Merriam, T. V. N. Neural computation in stylometry I: An application to the works of Shakespeare and Fletcher. *Literary and Linguistic Computing 8*, 4 (1993), 203–209.

[78] McCallum, A., Corrada-Emmanuel, A., and Wang, X. Topic and role discovery in social networks. In *Proceedings of IJCAI-19* (2005), pp. 786–791.

[79] McCallum, A., Pal, C., Druck, G., and Wang, X. Multi-conditional learning: Generative/discriminative training for clustering and classification. In *Proceedings of AAAI-21* (2006).

[80] McGurk, H., and MacDonald, J. Hearing lips and seeing voices. *Nature 264*, 5588 (1976), 746–748.

[81] Metzler, D., and Croft, W. B. A Markov random field model for term dependencies. In *Proceedings of SIGIR-28* (2005).

[82] Mishne, G. Experiments with mood classification in blog posts. In *Proceedings of the 1st Workshop on Stylistic Analysis of Text for Information Access* (2005).

[83] Ng, V., and Cardie, C. Improving machine learning approaches to coreference resolution. In *Proceedings of ACL-40* (2002).

[84] Pang, B., and Lee, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL-43* (2005), pp. 115–124.

[85] Pang, B., Lee, L., and Vaithyanathan, S. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing* (2002), pp. 79–86.

[86] Papadimitriou, C. H., and Steiglitz, K. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, 1982.

[87] Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[88] Pereira, F., Tishby, N., and Lee, L. Distributional clustering of English words. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics* (1993), pp. 183–190.

[89] Qiu, G. Image and feature co-clustering. In *Proceedings of ICPR-17* (2004), pp. 991–994.

[90] Raghavan, H., Madani, O., and Jones, R. InterActive feature selection. In *Proceedings of IJCAI-19* (2005), pp. 841–846.

[91] Schmid, C., Mohr, R., and Bauckhage, C. Comparing and evaluating interest points. In *Proceedings of the Sixth International Conference on Computer Vision* (1998).

[92] Seki, Y., Eguchi, K., and Kando, N. Analysis of multi-document viewpoint summarization using multi-dimensional genres. In *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications* (2004), pp. 150–153.

[93] Sha, F., and Pereira, F. Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL* (2003).

[94] Shamma, D. A., Owsley, S., Hammond, K. J., Bradshaw, S., and Budzik, J. Network arts: exposing cultural reality. In *Proceedings of WWW-04, the 13th international World Wide Web conference, Alternate track* (2004), pp. 41–47.

[95] Slonim, N. *The Information Bottleneck: Theory and Applications*. PhD thesis, The Hebrew University, 2003.

[96] Slonim, N., Friedman, N., and Tishby, N. Unsupervised document classification using sequential information maximization. In *Proceedings of SIGIR-25* (2002).

[97] Slonim, N., Friedman, N., and Tishby, N. Multivariate information bottleneck. *Neural Computation 18*, 8 (2006), 1739–1789.

[98] Slonim, N., and Tishby, N. Agglomerative information bottleneck. In *Advances in Neural Information Processing Systems 12 (NIPS)* (2000), pp. 617–623.

[99] Slonim, N., and Tishby, N. Document clustering using word clusters via the information bottleneck method. In *Proceedings of SIGIR-23* (2000), pp. 208–215.

[100] Stamatatos, E., Fakotakis, N., and Kokkinakis, G. K. Text genre detection using common word frequencies. In *Proceedings of the 18th International Conference on Computational Linguistics* (2000), pp. 808–814.

[101] Still, S., and Bialek, W. How many clusters? an information-theoretic perspective. *Neural Computation 16*, 12 (2004), 2483–2506.

[102] Sutton, C., and McCallum, A. Piecewise training of undirected models. In *Proceedings of UAI-21* (2005).

[103] Swan, R., and Allan, J. Automatic generation of overview timelines. In *Proceedings of SIGIR-23* (2000), pp. 49–56.

[104] Tao, T., and Zhai, C. A two-stage mixture model for pseudo feedback. In *Proceedings of the 27th annual international ACM SIGIR conference* (2004), pp. 486–487.

[105] Tax, D. M. J., and Duin, R. P. W. Outliers and data descriptions. In *Proceedings of the 7th Annual Conference of the Advanced School for Computing and Imaging* (2001), pp. 234–241.

[106] Tishby, N., Pereira, F., and Bialek, W. The information bottleneck method, 1999. Invited paper to the 37th Annual Allerton Conference on Communication, Control, and Computing.

[107] Turney, P. D. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (2002), pp. 417–424.

[108] Uematsu, Y., Kataoka, R., and Takeno, H. Clustering presentation of web image retrieval results using textual information and image features. In *Proceedings of the 24th IASTED international conference on Internet and multimedia systems and applications* (2006), pp. 217–222.

[109] Vapnik, V. N. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

[110] Wagstaff, K., and Cardie, C. Clustering with instance-level constraints. In *Proceedings of ICML-17* (2000).

[111] Wan, X., Gao, J., Li, M., and Ding, B. Person resolution in person search results: Webhawk. In *Proceedings of CIKM-14, the 14th ACM international conference on Information and knowledge management* (2005), pp. 163–170.

[112] Witten, I. H., and Frank, E. *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, 2005.

[113] Yang, K.-H., Chiou, K.-Y., Lee, H.-M., and Ho, J.-M. Web appearance disambiguation of personal names based on network motif. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* (2006), pp. 386–389.

[114] Yeung, R.W. A new outlook of Shannon's information measures. *IEEE transactions on information theory 37*, 3 (1991).

[115] Zhou, Y., and Croft, W. B. Query performance prediction in web search environments. In *Proceedings of the 30th Annual International ACM SIGIR Conference* (2007).

[116] Zhu, X. Semi-supervised learning literature survey. Tech. Rep. TR 1530, University of Wisconsin-Madison, 2005.

APPENDIX A

PROOF OF THEOREM 6.2.1

First, note that since both distributions P_r and P_g are uniform, then $P(W_{ij} = w|Z_{ij} = 1) = \frac{1}{m_r}$ and $P(W_{ij} = w|Z_{ij} = 0) = \frac{1}{m}$.

Let us now compute the marginal $P(W_{ij} = w)$. For a relevant word w_r, let us denote it $P(W_{ij} = w_r) = p_r$:

$$p_r = P(W_{ij} = w_r) = P(W_{ij} = w_r|Z_{ij} = 1)P(Z_{ij} = 1) + P(W_{ij} = w_r|Z_{ij} = 0)P(Z_{ij} = 0) = \frac{1}{m_r}\pi + \frac{1}{m}(1 - \pi) \tag{A.1}$$

For a non-relevant word w_n, denote $P(W_{ij} = w_n) = p_n$:

$$p_n = P(W_{ij} = w_n) = P(W_{ij} = w_n|Z_{ij} = 1)P(Z_{ij} = 1) + P(W_{ij} = w_n|Z_{ij} = 0)P(Z_{ij} = 0) = 0 \cdot \pi + \frac{1}{m}(1 - \pi) = \frac{1}{m}(1 - \pi)$$

We assume that the difference between these two probabilities is substantial, that is $p_r - p_n = \pi/m_r >> 0$. Let τ be their arithmetic mean:

$$\tau = \frac{1}{2}(p_r + p_n). \tag{A.2}$$

For each word w, we introduce a random variable X_w of its count (the number of its occurrences in the dataset), which is distributed *binomially*: if w is relevant, then $X_w \sim Bi(p_r, N)$ and its mean is p_rN; if w is non-relevant, then $X_w \sim Bi(p_n, N)$ with mean p_nN. We are interested in bounding the probability that $X_w \leq \tau N$ for relevant words, and that $X_w \geq \tau N$ for non-relevant words.

Using Chernoff bound for a relevant word w, we have:

$$P(X_w \leq \tau N) \leq \exp\left(-N\frac{(p_r - \tau)^2}{2p_r}\right) \leq \epsilon. \tag{A.3}$$

For a non-relevant word w we have:

$$P(X_w \geq \tau N) \leq \exp\left(-N\frac{(p_n - \tau)^2}{3p_n}\right) \leq \epsilon. \tag{A.4}$$

Solving (A.3) and (A.4) simultaneously with respect to N, we have:

$$N \geq \max\left(\frac{2p_r \ln\frac{1}{\epsilon}}{(p_r - \tau)^2}, \frac{3p_n \ln\frac{1}{\epsilon}}{(p_n - \tau)^2}\right),$$

and then substituting τ from (A.2):

$$N \geq \max\left(\frac{8p_r \ln\frac{1}{\epsilon}}{(p_r-p_n)^2}, \frac{12p_n \ln\frac{1}{\epsilon}}{(p_r-p_n)^2}\right) = \frac{8p_r \ln\frac{1}{\epsilon}}{(p_r-p_n)^2},$$

where we use the given constraint that $p_w < 2\pi$ (and thus $3p_n < 2p_r$, so the first term is always greater than the second one). Substituting $p_r - p_n = \frac{\pi}{p_w m}$, and applying the definition of p_r from (A.1), we get:

$$\begin{aligned}\frac{8p_r \ln\frac{1}{\epsilon}}{(p_r-p_n)^2} &= 8\frac{\pi + p_w - \pi p_w}{p_w m} \cdot \frac{p_w^2 m^2}{\pi^2} \ln\frac{1}{\epsilon} \\ &\leq 8\frac{p_w m}{\pi^2} \ln\frac{1}{\epsilon} \leq 16\frac{m}{\pi}\ln\frac{1}{\epsilon},\end{aligned}$$

where we used the fact that $\pi + p_w - \pi p_w \leq 1$ and that $p_w < 2\pi$. Finally, we choose the value of N to be the minimum among all the possible choices:

$$N = 16\frac{m}{\pi}\ln\frac{1}{\epsilon}.$$

Putting it all together: What is the probability that there exists a word w which was not detected correctly? Using the union bound we get:

$$P\left(\cup_{w\in\mathcal{R}}(X_w \leq \tau N) \bigcup \cup_{w\notin\mathcal{R}}(X_w \geq \tau N)\right) \leq \epsilon m = \delta,$$

so $\epsilon = \frac{\delta}{m}$, and then

$$N = 16\frac{m}{\pi}\ln\frac{m}{\delta},$$

which is log-linear in m. □

APPENDIX B

DETAILS OF EM ALGORITHM FOR ONE-CLASS CLUSTERING

Given the graphical model from Figure 6.1, the joint distribution is:

$$P(\{Y\},\{Z\},\{w\}) = \prod_i P(Y_i) \prod_j [P(Z_{ij}|Y_i)P(w_{ij}|Z_{ij})] \tag{B.1}$$

Note that we represent a document d_i as its Bag-Of-Words: $d_i \triangleq \{w_{i1}, w_{i2}, \ldots, w_{i|d_i|}\}$. Let us now define EM parameters Θ:

$$\begin{aligned}
P(Y_i = 1) &= p_d && \text{(B.2)}\\
\text{For each document } d_i|_{i-1}^n: \quad P(Z_{ij} = 1|Y_i = 1) &= \pi_i && \text{(B.3)}\\
P(Z_{ij} = 1|Y_i = 0) &= 0 && \text{(B.4)}\\
\text{For each word } w_l|_{l=1}^m: \quad P(w_l|Z_l = 1) &= p_r(w_l) && \text{(B.5)}\\
P(w_l|Z_l = 0) &= p_g(w_l) && \text{(B.6)}
\end{aligned}$$

Using this notation, the marginal distribution of a document is written as:

$$\begin{aligned}
P(d_i) &= \sum_{Y_i} P(Y_i) \sum_{Z_{i1}} P(Z_{i1}|Y_i)P(w_{i1}|Z_{i1}) \sum_{Z_{i2}} P(Z_{i2}|Y_i)P(w_{i2}|Z_{i2}) \ldots \\
&\quad \sum_{Z_{i|d_i|}} P(Z_{i|d_i|}|Y_i)P(w_{i|d_i|}|Z_{i|d_i|}) \\
&= \sum_{Y_i} P(Y_i) \prod_{j=1}^{|d_i|} (P(Z_{ij} = 1|Y_i)P(w_{ij}|Z_{ij} = 1) + P(Z_{ij} = 0|Y_i)P(w_{ij}|Z_{ij} = 0)) \\
&= p_d \prod_{j=1}^{|d_i|} (\pi_i \; p_r(w_{ij}) + (1 - \pi_i) \; p_g(w_{ij})) + (1 - p_d) \prod_{j=1}^{|d_i|} p_g(w_{ij}) \qquad \text{(B.7)}
\end{aligned}$$

E-step

Given the current set of parameters Θ^k at iteration k, for each document d_i and each word w_{ij} in d_i, we compute the posteriors:

$$\tilde{P}^k(Y_i = 1|d_i, \Theta^k) = \frac{P(d_i|Y_i = 1, \Theta^k)P(Y_i = 1|\Theta^k)}{P(d_i|\Theta^k)}$$

$$= \frac{p_d \prod_{j=1}^{|d_i|} \left(\pi_i^k \; p_r^k(w_{ij}) + (1-\pi_i^k) \; p_g^k(w_{ij})\right)}{\underbrace{p_d \prod_{j=1}^{|d_i|} \left(\pi_i^k \; p_r^k(w_{ij}) + (1-\pi_i^k) \; p_g^k(w_{ij})\right) + (1-p_d) \prod_{j=1}^{|d_i|} p_g^k(w_{ij})}_{\text{denote } \alpha_i^k}}$$

$$\begin{aligned}
\tilde{P}^k(Y_i = 1, Z_{ij} = 1 | d_i, \Theta^k) &= \tilde{P}^k(Y_i = 1 | d_i, \Theta^k) P(Z_{ij} = 1 | Y_i = 1, w_{ij}, \Theta^k) \\
&= \tilde{P}^k(Y_i = 1 | d_i, \Theta^k) \times \\
&\quad \frac{P(w_{ij}, Z_{ij} = 1 | Y_i = 1, \Theta^k)}{P(w_{ij}, Z_{ij} = 1 | Y_i = 1, \Theta^k) + P(w_{ij}, Z_{ij} = 0 | Y_i = 1, \Theta^k)} \\
&= \tilde{P}^k(Y_i = 1 | d_i, \Theta^k) \underbrace{\frac{\pi_i^k \; p_r^k(w_{ij})}{\pi_i^k \; p_r^k(w_{ij}) + (1-\pi_i^k) \; p_g^k(w_{ij})}}_{\text{denote } \beta_{ij}^k} \\
&\triangleq \alpha_i^k \; \beta_{ij}^k \\
\tilde{P}^k(Y_i = 1, Z_{ij} = 0 | d_i, \Theta^k) &= \tilde{P}^k(Y_i = 1 | d_i, \Theta^k) \underbrace{\frac{(1-\pi_i^k) \; p_g^k(w_{ij})}{\pi_i^k \; p_r^k(w_{ij}) + (1-\pi_i^k) \; p_g^k(w_{ij})}}_{\text{this term is } (1-\beta_{ij}^k)} \\
&\triangleq \alpha_i^k \; (1 - \beta_{ij}^k) \\
\tilde{P}^k(Z_{ij} = 1 | d_i, \Theta^k) &= \tilde{P}^k(Y_i = 1, Z_{ij} = 1 | d_i, \Theta^k) \triangleq \alpha_i^k \; \beta_{ij}^k \\
\tilde{P}^k(Z_{ij} = 0 | d_i, \Theta^k) &= 1 - \tilde{P}^k(Z_{ij} = 1 | d_i, \Theta^k) \triangleq 1 - \alpha_i^k \; \beta_{ij}^k
\end{aligned}$$

M-step

We maximize:

$$\begin{aligned}
Q(\Theta^{k+1} | \Theta^k) &= \sum_i E[\log(P(Y_i, \{Z_{ij}\}, \{w_{ij}\} | \Theta^{k+1}) | \tilde{P}^k] \\
&= \sum_i E\left[\log\left(P(Y_i | \Theta^{k+1}) \prod_j P(Z_{ij} | Y_i, \Theta^{k+1}) \prod_j P(w_{ij} | Z_{ij}, \Theta^{k+1})\right) | \tilde{P}^k\right] \\
&= \underbrace{\sum_i E\left[\log\left(P(Y_i | \Theta^{k+1})\right) | \tilde{P}^k\right]}_{\text{denote } A} + \underbrace{\sum_{i,j} E\left[\log\left(P(Z_{ij} | Y_i, \Theta^{k+1})\right) | \tilde{P}^k\right]}_{\text{denote } B} \\
&\quad + \underbrace{\sum_{i,j} E\left[\log\left(P(w_{ij} | Z_{ij}, \Theta^{k+1})\right) | \tilde{P}^k\right]}_{\text{denote } C}
\end{aligned}$$

The A portion should not be optimized, because p_d is a constant in our setting.

$$B = \sum_{i,j} E\left[\log\left(P(Z_{ij} | Y_i, \Theta^{k+1})\right) | \tilde{P}^k\right]$$

$$
\begin{aligned}
&= \sum_{i,j} \tilde{P}^k(Y_i = 1, Z_{ij} = 1|d_i) \log\left(P(Z_{ij} = 1|Y_i = 1, \Theta^{k+1})\right) \\
&\quad + \sum_{i,j} \tilde{P}^k(Y_i = 1, Z_{ij} = 0|d_i) \log\left(P(Z_{ij} = 0|Y_i = 1, \Theta^{k+1})\right) \\
&\quad + \sum_{i,j} \tilde{P}^k(Y_i = 0, Z_{ij} = 1|d_i) \log\left(P(Z_{ij} = 1|Y_i = 0, \Theta^{k+1})\right) \\
&\quad + \sum_{i,j} \tilde{P}^k(Y_i = 0, Z_{ij} = 0|d_i) \log\left(P(Z_{ij} = 0|Y_i = 0, \Theta^{k+1})\right) \\
&= \sum_{i,j} \alpha_i^k \ \beta_{ij}^k \log\left(\pi_i^{k+1}\right) \\
&\quad + \sum_{i,j} \alpha_i^k \ (1 - \beta_{ij}^k) \log\left(1 - \pi_i^{k+1}\right) \\
&\quad + \sum_{i,j} 0 \log(0) + \sum_{i,j} 1 \log(1)
\end{aligned}
$$

$$
\begin{aligned}
C &= \sum_{i,j} E\left[\log\left(P(w_{ij}|Z_{ij}, \Theta^{k+1})\right)|\tilde{P}^k\right] \\
&= \sum_{i,j} \tilde{P}(Z_{ij} = 1|d_i) \log\left(P(w_{ij}|Z_{ij} = 1, \Theta^{k+1})\right) \\
&\quad + \sum_{i,j} \tilde{P}(Z_{ij} = 0|d_i) \log\left(P(w_{ij}|Z_{ij} = 0, \Theta^{k+1})\right) \\
&= \sum_{i,j} \alpha_i^k \ \beta_{ij}^k \log\left(p_r^{k+1}(w_{ij})\right) \\
&\quad + \sum_{i,j} (1 - \alpha_i^k \ \beta_{ij}^k) \log\left(p_g^{k+1}(w_{ij})\right)
\end{aligned}
$$

Now we compute derivatives of $Q(\Theta^{k+1}|\Theta^k)$ with respect to π_i^{k+1}, $p_r^{k+1}(w_l)$, $p_g^{k+1}(w_l)$ and find their values. First, let us find the optimal value of π_i^{k+1}.

$$
\begin{aligned}
\frac{\partial Q}{\partial \pi_i^{k+1}} &= \frac{\partial B}{\partial \pi_i^{k+1}} = \frac{\partial}{\partial \pi_i^{k+1}} \left[\sum_j \alpha_i^k \ \beta_{ij}^k \log \pi_i^{k+1} + \sum_j \alpha_i^k \ (1 - \beta_{ij}^k) \log(1 - \pi_i^{k+1}) \right] \\
&= \alpha_i^k \left[\sum_j \beta_{ij}^k \frac{1}{\pi_i^{k+1}} - \sum_j (1 - \beta_{ij}^k) \frac{1}{1 - \pi_i^{k+1}} \right] = 0 \\
\pi_i^{k+1} &= \frac{\sum_j \beta_{ij}^k}{\sum_j \beta_{ij}^k + \sum_j (1 - \beta_{ij}^k)} = \frac{1}{|d_i|} \sum_j \beta_{ij}^k
\end{aligned}
$$

Second, let us find the optimal value of $p_r^{k+1}(w_l)$:

$$
\frac{\partial Q}{\partial p_r^{k+1}(w_l)} = \frac{\partial C}{\partial p_r^{k+1}(w_l)}
$$

$$
\begin{aligned}
&= \frac{\partial}{\partial p_r^{k+1}(w_l)} \left[\sum_{i,j} \alpha_i^k \ \beta_{ij}^k \log\left(p_r^{k+1}(w_{ij})\right) + \lambda \left(1 - \sum_l^m p_r^{k+1}(w_l)\right)\right] \\
&= \frac{\partial}{\partial p_r^{k+1}(w_l)} \left[\sum_{i,j} \delta(w_{ij} = w_l) \ \alpha_i^k \ \beta_{ij}^k \log\left(p_r^{k+1}(w_l)\right) + \lambda \left(1 - \sum_l^m p_r^{k+1}(w_l)\right)\right] \\
&= \frac{\sum_{i,j} \delta(w_{ij} = w_l) \ \alpha_i^k \ \beta_{ij}^k}{p_r^{k+1}(w_l)} - \lambda = 0 \\
p_r^{k+1}(w_l) &= \frac{1}{\lambda} \sum_{i,j} \delta(w_{ij} = w_l) \ \alpha_i^k \ \beta_{ij}^k \\
1 &= \sum_l^m p_r^{k+1}(w_l) = \sum_l^m \frac{1}{\lambda} \sum_{i,j} \delta(w_{ij} = w_l) \ \alpha_i^k \ \beta_{ij}^k = \frac{1}{\lambda} \sum_{i,j} \alpha_i^k \ \beta_{ij}^k \\
\lambda &= \sum_{i,j} \alpha_i^k \ \beta_{ij}^k \\
p_r^{k+1}(w_l) &= \frac{\sum_{i,j} \delta(w_{ij} = w_l) \ \alpha_i^k \ \beta_{ij}^k}{\sum_{i,j} \alpha_i^k \ \beta_{ij}^k} = \frac{\sum_i \alpha_i^k \sum_j \delta(w_{ij} = w_l) \ \beta_{ij}^k}{\sum_i \alpha_i^k \sum_j \beta_{ij}^k}
\end{aligned}
$$

Finally, let us find the optimal value of $p_g^{k+1}(w_l)$:

$$
\begin{aligned}
\frac{\partial Q}{\partial p_g^{k+1}(w_l)} &= \frac{\partial C}{\partial p_g^{k+1}(w_l)} \\
&= \frac{\partial}{\partial p_g^{k+1}(w_l)} \left[\sum_{i,j} (1 - \alpha_i^k \ \beta_{ij}^k) \log\left(p_g^{k+1}(w_{ij})\right) + \lambda \left(1 - \sum_l^m p_g^{k+1}(w_l)\right)\right] \\
&= \frac{\partial}{\partial p_g^{k+1}(w_l)} \left[\sum_{i,j} \delta(w_{ij} = w_l)(1 - \alpha_i^k \ \beta_{ij}^k) \log\left(p_g^{k+1}(w_l)\right) \right. \\
&\quad \left. + \lambda \left(1 - \sum_l^m p_g^{k+1}(w_l)\right)\right] \\
&= \frac{\sum_{i,j} \delta(w_{ij} = w_l)(1 - \alpha_i^k \ \beta_{ij}^k)}{p_g^{k+1}(w_l)} - \lambda = 0 \\
p_g^{k+1}(w_l) &= \frac{\sum_{i,j} \delta(w_{ij} = w_l)(1 - \alpha_i^k \ \beta_{ij}^k)}{\sum_{i,j}(1 - \alpha_i^k \ \beta_{ij}^k)} = \frac{N_w - \sum_i \alpha_i^k \sum_j \delta(w_{ij} = w_l) \ \beta_{ij}^k}{N - \sum_i \alpha_i^k \sum_j \beta_{ij}^k}
\end{aligned}
$$

EM algorithm

To compute α_i and β_{ij} efficiently, let us use the following relations:

$$
\begin{aligned}
\alpha_i &= \frac{p_d \prod_{j=1}^{|d_i|} (\pi_i \ p_r(w_{ij}) + (1 - \pi_i) \ p_g(w_{ij}))}{p_d \prod_{j=1}^{|d_i|} (\pi_i \ p_r(w_{ij}) + (1 - \pi_i) \ p_g(w_{ij})) + (1 - p_d) \prod_{j=1}^{|d_i|} p_g(w_{ij})} \\
&= \frac{1}{1 + \frac{1-p_d}{p_d} \prod_j \frac{p_g(w_{ij})}{\pi_i \ p_r(w_{ij}) + (1-\pi_i) \ p_g(w_{ij})}} = \frac{1}{1 + \frac{1-p_d}{p_d} \prod_j \frac{1}{\pi_i \frac{p_r(w_{ij})}{p_g(w_{ij})} + 1 - \pi_i}}
\end{aligned}
$$

$$= \frac{1}{1 + \frac{\frac{1-p_d}{p_d}}{\prod_j \left(\frac{\pi_i \ p_r(w_{ij})}{(1-\pi_i) \ p_g(w_{ij})} + 1 \right)^{(1-\pi_i)}}}$$

$$\beta_{ij} = \frac{\pi_i \ p_r(w_{ij})}{\pi_i \ p_r(w_{ij}) + (1-\pi_i) \ p_g(w_{ij})} = \frac{1}{1 + \frac{1-\pi_i}{\pi_i} \ \frac{p_g(w_{ij})}{p_r(w_{ij})}} = \frac{1}{1 + \frac{1}{\frac{\pi_i \ p_r(w_{ij})}{(1-\pi_i) \ p_g(w_{ij})}}}$$

Let us denote $\gamma_{ij}^k = \frac{\pi_i^k \ p_r^k(w_{ij})}{(1-\pi_i^k) \ p_g^k(w_{ij})}$. Then we have

$$\alpha_i^k = \frac{1}{1 + \frac{\frac{1-p_d}{p_d}}{\prod_j (\gamma_{ij}^k + 1)(1-\pi_i^k)}} \tag{B.8}$$

$$\beta_{ij}^k = \frac{1}{1 + \frac{1}{\gamma_{ij}^k}} \tag{B.9}$$

Algorithm:

1. Initialization:
   (a) For each document d_i: $\pi_i^0 \leftarrow p_w$.
   (b) For each word w_l: $p_r^0(w_l) \leftarrow \frac{score(w_l)}{\sum_{l'} score(w_{l'})}$, and $p_g^0(w_l) \leftarrow \frac{\frac{1}{score(w_l)}}{\sum_{l'} \frac{1}{score(w_{l'})}}$.
2. For each document d_i:
   (a) For each word w_{ij} calculate γ_{ij}^k, and then β_{ij}^k using (B.9).
   (b) Accumulate $\prod_j (\gamma_{ij}^k + 1)(1-\pi_i)$. Calculate α_i^k using (B.8).
   (c) Accumulate $\sum_j \beta_{ij}^k$. Calculate $\pi_i^{k+1} \leftarrow \frac{1}{|d_i|} \sum_j \beta_{ij}$.
3. Over all documents, accumulate $\psi^k \leftarrow \sum_i \alpha_i^k \sum_j \beta_{ij}^k$.
4. Rank documents in decreasing order of α_i^k. Stop if the ranking has not changed since the previous iteration.[1]
5. For each word w_l
   (a) Over all documents, accumulate $\varrho_l^k \leftarrow \sum_i \alpha_i^k \sum_j \delta(w_{ij} = w_l) \ \beta_{ij}^k$.
   (b) Calculate $p_r^{k+1}(w_l) \leftarrow \frac{\varrho_l^k}{\psi^k}$, and $p_g^{k+1}(w_l) = \frac{N_{w_l} - \varrho_l^k}{N - \psi^k}$.
6. $k \leftarrow k+1$. Go to 2.

[1] Alternatively, the algorithm can be terminated after a predefined number of EM iterations.