Collective Document Classification using Explicit and Implicit Inter-document Relationships

A thesis presented
by
Clinton Burford
to
The Department of Computing and Information Systems
in total fulfillment of the requirements
for the degree of
Doctor of Philosophy

The University of Melbourne
Melbourne, Australia
May 2013
©2013 - Clinton Burford

All rights reserved.
Abstract

Information systems are transforming the ways in which people generate, store and share information. One consequence of this change is a massive increase in the quantity of digital content the average person needs to deal with. A large part of the information systems challenge is about finding intelligent ways to help users locate and analyse this information. One tool that is available to build systems to address this challenge is automatic document classification.

A document classifier is a statistical model for predicting a label for an input document that is represented as a set of features. The potential usefulness of such a generalised system for categorising documents based on their contents is very great. There are direct applications for systems that can answer complex document categorisation questions like: Is this product review generally positive or negative? Document classification systems can also become critical parts of most complex systems that need input documents to be selected based on complex criteria.

This thesis addresses the question of how document classifiers can exploit information about the relationships between documents being classified. Normally, document classifiers work on a single document at a time: once the classifier has been trained from a set of labelled examples, it can then be used to label single input documents as required. Collective document classifiers learn a classifier that can be applied to a group of related documents. The inter-document relationships in the group are used to improve labelling performance beyond what is possible when considering documents in isolation.

Work on collective document classifiers is based on the observation that some types of documents have features which are either ambiguous or not present in training data,
but which have the special characteristic of indicating relationships between the labels of documents. Most often, an inter-document relationship indicates that two documents have the same label, but it may also indicate that they have different labels. In either case, classifiers gain an advantage if they can consider inter-document features.

Inter-document features can be explicit, as when a document cites or quotes another, or implicit, as when documents exist in semantically related groups in which stylistic, structural or semantic similarities are informative, or when they are related by a spatial or temporal structure.

In the first part of this thesis I survey the state-of-the-art in collective document classification and explore approaches for adding collective behaviour to standard document classifiers. I present an experimental evaluation of these techniques for use with explicit inter-document relationships. In the second part I develop techniques for extracting implicit inter-document relationships.

In total, the work in this thesis assesses and extends the capabilities of collective document classifiers. Its contribution is in four main parts: (1) I introduce an approach that gives better than state of the art performance for collective classification of political debate transcripts; (2) I provide a comparative overview of collective document classification techniques to assist practitioners in choosing an algorithm for collective document classification tasks; (3) I demonstrate effective and novel approaches for generating collective classifiers from standard classifiers; and (4) I introduce a technique for inferring inter-document relationships based on matching phrases and show that these relationships can be used to improve overall document classification performance.
Declaration

This is to certify that

i. the thesis comprises only my original work towards the PhD

ii. due acknowledgement has been made in the text to all other material used

iii. the thesis is less than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices

(Clinton Burford)
## Contents

| Title Page                                      | i                |
| Abstract                                        | iii              |
| Declaration                                     | v                |
| Table of Contents                               | vi               |
| List of Figures                                 | ix               |
| List of Tables                                  | xi               |
| Citations to Previously Published Work          | xii              |
| Acknowledgments                                 | xiii             |
| Dedication                                      | xiv              |

### 1 Introduction

1.1 Aim and Contribution                         | 4                |
1.2 Thesis Structure                             | 8                |

### 2 Literature Review

2.1 Introduction                                 | 11               |

#### 2.2 Document Classification

2.2.1 Document Classification Terminology        | 13               |
2.2.2 Practical Considerations for Document Classification | 13 |
2.2.3 Support Vector Machines                    | 14               |

#### 2.3 Collective Document Classification

2.3.1 Definition, Notation and Terminology        | 15               |
2.3.2 Collective Document Classification with Explicit Links | 17 |
2.3.3 Collective Document Classification with Implicit Links | 22 |

#### 2.4 Collective Classification Algorithms

2.4.1 Iterative Classifiers                       | 26               |
2.4.2 Feature Models for Iterative Classifiers    | 28               |
2.4.3 Markov Random Fields                        | 32               |
2.4.4 Classifying with Markov Random Fields       | 35               |
2.4.5 Learning with Markov Random Fields          | 36               |
2.4.6 The Minimum-Cut Technique                  | 39               |
2.4.7 Dual Classifiers                            | 40               |
4.8 Experiments ................................................................. 88
  4.8.1 Relationship Classifier Performance ......................... 90
  4.8.2 Local Classifier Performance ................................ 91
  4.8.3 Collective Classifier Performance without Tuning ........ 91
  4.8.4 Collective Classifier Performance with Tuning ............ 93
4.9 Conclusions ............................................................. 95
  4.9.1 Differences over Earlier Work ................................ 99
4.10 Future Work .......................................................... 101

5 Collective Document Classification with Implicit Relationships 103
  5.1 Introduction .......................................................... 103
  5.2 Similarity-based Inter-Document Relationships .................. 105
    5.2.1 Preliminary Experiment 1: Finding Label Agreement with Shared Words 106
    5.2.2 Analysis of Shared Words .................................... 108
    5.2.3 Preliminary Experiment 2: Similarity-based Inter-document Relationship Detection 110
    5.2.4 Preliminary Experiment 3: Similarity-based Local Classification .... 115
  5.3 Dual Classifier Approach ........................................... 119
    5.3.1 Base classification ............................................. 119
    5.3.2 Normalisation .................................................. 120
    5.3.3 Decoding ....................................................... 121
    5.3.4 Tuning .......................................................... 121
  5.4 Iterative Classifier Approach ...................................... 122
    5.4.1 Base Classification ............................................. 123
    5.4.2 Relational Features .......................................... 123
  5.5 Experiments .......................................................... 124
    5.5.1 Local Classifier Performance ................................ 125
    5.5.2 Collective Classifier Performance ......................... 126
    5.5.3 Dual Classifier Dampening Response ....................... 129
  5.6 Relationship to Prior Work ........................................ 130
  5.7 Conclusions .......................................................... 136
    5.7.1 A Theoretical Difficulty with Graph-based Methods ........ 138
  5.8 Future Work .......................................................... 139

6 Conclusions and Future Work ........................................ 142
# List of Figures

1.1 A comparison of content-only and collective document classification approaches. ............................................. 5

2.1 Example in-sample and out-of-sample settings for a collective classification problem. ........................................ 18

2.2 A toy binary collective document classification training problem. ................................................................. 27

2.3 Iterative classifier approach. ....................................................................................................................... 30

2.4 A toy binary collective document classification problem. .............................................................................. 31

2.5 A toy binary collective document classification problem with clique potentials. ........................................... 34

2.6 A toy binary collective document classification problem framed as a flow graph. ........................................... 41

2.7 The dual classifier approach to collective classification. .................................................................................. 43

3.1 Sample by-name references from the ConVote Corpus. ...................................................................................... 59

3.2 Tokenised sample text from a ConVote debate. ................................................................................................. 63

3.3 Cumulative distribution of by-name references for ConVote speeches. .......................................................... 64

3.4 Tokenised Bitterlemons Corpus sample text. .................................................................................................. 67

4.1 The Thomas et al. dual classifier. .................................................................................................................... 73

4.2 The Markov random field dual classifier with probabilistic SVM normalisation. .......................................... 83

4.3 Iterative classifier approach to ConVote floor-debate classification. .............................................................. 86

4.4 Count and context window iterative classifier features for three inter-linked sample speeches. ....................... 89

5.1 Proportion of result set that is \texttt{SAME} when ConVote instances are paired based on shared high-idf terms. 108

5.2 Proportion of result set that is \texttt{SAME} when Bitterlemons instances are paired based on shared high-idf terms. 109

5.3 Tokenised sample text from ConVote showing examples of high-idf word overlap predicting \texttt{SAME} relationships between pairs of speakers. 111
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4</td>
<td>Tokenised sample text from Bitterlemons showing examples of high-idf word overlap predicting SAME_LABEL relationships between pairs of contributors.</td>
<td>112</td>
</tr>
<tr>
<td>5.5</td>
<td>Precision versus recall for ConVote relationship detection using cosine similarity.</td>
<td>115</td>
</tr>
<tr>
<td>5.6</td>
<td>Precision versus recall for Bitterlemons relationship detection using cosine similarity.</td>
<td>116</td>
</tr>
<tr>
<td>5.7</td>
<td>Dual classifier with similarity-based links.</td>
<td>122</td>
</tr>
<tr>
<td>5.8</td>
<td>Iterative classifier approach with similarity-based relational features.</td>
<td>124</td>
</tr>
<tr>
<td>5.9</td>
<td>Dual classifier performance against dampening factor for the ConVote Corpus with similarity-based links and the minimum-cut algorithm.</td>
<td>130</td>
</tr>
<tr>
<td>5.10</td>
<td>Dual classifier performance against dampening factor for the ConVote Corpus with similarity-based links and the loopy belief algorithm.</td>
<td>131</td>
</tr>
<tr>
<td>5.11</td>
<td>Dual classifier performance against dampening factor for the ConVote Corpus with similarity-based links and the mean-field algorithm.</td>
<td>132</td>
</tr>
<tr>
<td>5.12</td>
<td>Dual classifier performance against dampening factor for the Bitterlemons Corpus with similarity-based links and the minimum-cut algorithm.</td>
<td>132</td>
</tr>
<tr>
<td>5.13</td>
<td>Dual classifier performance against dampening factor for the Bitterlemons Corpus with similarity-based links and the loopy belief algorithm.</td>
<td>133</td>
</tr>
<tr>
<td>5.14</td>
<td>Dual classifier performance against dampening factor for the Bitterlemons Corpus with similarity-based links and the mean-field algorithm.</td>
<td>133</td>
</tr>
<tr>
<td>5.15</td>
<td>An example of incorrect handling of similarity-based links by graph-based models.</td>
<td>140</td>
</tr>
</tbody>
</table>
List of Tables

2.1 Collective classification performance on the Cora Corpus. ............... 48
2.2 Collective classification performance on the CiteSeer Corpus. .......... 49
2.3 Collective classification performance on the ConVote Corpus. .......... 52

3.1 ConVote statistics. .................................................... 58
3.2 Corpus statistics for the modified ConVote Corpus. ...................... 61
3.3 Corpus statistics for the Bitterlemons Corpus. ............................ 68

4.1 Relationship classifier performance for all relationships and for relations- ships classified as DIFFERENT LABEL and SAME LABEL. ................. 91
4.2 Local classifier performance for count features, binary context window fea- tures and frequency count context window features. ........................ 92
4.3 Collective classification performance with no tuned parameters. ........ 94
4.4 Collective classification performance with tuned parameters. ............. 96

5.1 Local classifier performance for average neighbour similarity features on the ConVote Corpus. .................................................. 118
5.2 Local classifier performance for average neighbour similarity features on the Bitterlemons Corpus. ........................................... 118
5.3 Local classifier performance on the ConVote Corpus using similarity-based features. .............................................................. 126
5.4 Local classifier performance on the Bitterlemons Corpus using similarity- based features. .............................................................. 126
5.5 Collective classification performance on the ConVote Corpus. .......... 128
5.6 Collective classification performance on the Bitterlemons Corpus. .... 128
5.7 Collective classification performance on the Movies Corpus. .............. 136
Citations to Previously Published Work

The basis of the work presented in Chapter 4 of this thesis was first published in:


This was predominantly my own work, with minor contributions from the co-authors. Additional parts of the work presented in Chapter 4 were first published in:


Note that I changed the spelling of my surname between the publication of these two papers and the writing of this thesis.
Acknowledgments

I am deeply grateful to Steven Bird and Tim Baldwin for their service to me as Ph.D. supervisors. All of the strengths of this work owe something to their excellence as researchers and educators; all of the imperfections are my own.

I thank Adrian Pearce and Laurence Park for providing feedback on this thesis at various stages and assisting with administrative oversight during their stints as members of my supervisory committee.

Other valuable comments on works that became the basis for this thesis were provided by David Newman and Ramamohanarao Kotagiri.

I am also indebted to Veselin Stoyanov for making his ERMA code available and providing email advice about its use.

My work on this thesis was assisted by travel grants for attendance at the ACL 2009 and ACL 2011 conferences, provided by the University of Melbourne Department of Computer Science and Google Australia respectively.
Dedicated to Basil S. Brown.
Chapter 1

Introduction

This thesis is about the timeless human challenge of finding, reading, and understanding documents, and about new, automatic methods for speeding up the process. It speaks to a context where information technology is enabling the publication, linking, and accumulation of vast and ever growing quantities of human communication. At the highest level, it asks the question: how can automatic systems help members of the new information society make sense of all that they write and say?

The vision of artificial intelligence as applied to information processing systems is that our tools will be able to process written and spoken language in ways that remove or reduce the need for human effort. In many regards this vision is being realised. Compared to a few decades ago, information retrieval systems do a significantly better job of understanding queries and returning relevant documents (Carpineto and Romano 2012). Likewise, machine translation systems produce output that is measurably closer to that produced by human translators (Lopez 2008). New domains like email spam detection yield ever more sophisticated solutions (Caruana and Li 2008). These are just three prominent examples of progress in the field.

The vision extends beyond just saving time and labour by automating existing tasks. There are also tasks that involve computers ingesting millions of documents and suggesting conclusions that could be difficult to draw otherwise. Real-life systems in the fields of epidemiology and counter-terrorism show the promise of this area[1].

[1]Palantir and i2 are prominent analytical systems with features like this (http://www.palantir.com/,
The consistently decreasing cost of computational resources is the main enabler of such progress. After computing power, which is the most basic concern for all information processing systems, the disciplines of information retrieval and natural language processing are perhaps the most directly applicable and important computer science disciplines. Within natural language processing, document classification is a foundational task. It asks: can we build automated systems to usefully categorise documents based on their content?

Document classification systems provide an excellent measure of our ability to perform natural language processing because it is easy to frame document classification tasks that embody human levels of understanding. A computer system is clearly exhibiting meaningful artificial intelligence if it can answer questions like, What is the broad topic area in which this document belongs? or Is the content of this document primarily subjective or objective? or Is this document written from a predominantly left-wing or right-wing political perspective?

Document classification is also a useful facility in itself. A system that can answer questions like the three above is not just scientifically interesting, it is also practical. There are clear and direct applications for document classification systems that can perform with sufficient speed and accuracy. A document classifier can be the basis for a search system that is specialised for a topic domain or for a political or emotional perspective. A document classifier can also be the basis for an e-rulemaking system that groups public email submissions into meaningful categories for processing by a human expert. Spam detection is a type of document classification. These are just three of a great many applications for document classifiers.

Like any human faculty that is useful in itself, document classification can also function as a foundation for other capabilities. The performance of learning systems depends on high quality inputs, which document classifiers can provide if they are able to distinguish useful candidate input documents from noisy ones. As high-level processors that work on documents as single units, document classifiers can provide valuable pre-processing for lower-level systems that need to extract information at a paragraph, sentence, phrase or word level.

http://www.i2group.com/).
The specific topic of this thesis is collective document classification. That is, the branch of document classification concerned with categorising documents not individually, but as parts of an inter-connected whole. The objective is the same as with standard document classification, but the means of inference are different. A collective document classifier considers the contents of documents as well as the relationships between them.

As an example, consider the process of classifying each entry in a collection of political blog articles as having a left-wing or right-wing perspective. To a human reader, a political blog post will normally indicate its perspective by the stance it takes on an issue or towards a public figure. There may also be other stylistic (e.g. preference for particular grammatical forms) or incidental features (e.g. spelling mistakes) of the language used that are indicative.

In the normal case, a document classifier will process documents individually, classifying each based on the presence or absence of these cues. This is known as content-only classification. The collective element is incorporated when the classifier begins to use the links between posts as an additional source of information. For example, a post that is difficult to classify based on content alone may become simpler to classify when it is observed that it links to and criticises another post that is clearly from the left.

Figure 1.1 gives a diagrammatic illustration of the difference between content-only and collective approaches to document classification. In the content-only case, shown in the top half of the figure, the five documents are classified individually based on their content alone. If we were to wipe any number of the documents from the test set, the classifications of those remaining would be unaltered. The box around the collective case, shown in the bottom half of the figure, indicates that the classification problem should be seen as an indivisible whole. This is the because the labels assigned are going to be based not only on the contents of the documents themselves, but also on the relationships between documents. These relationships are indicated by dotted lines connecting pairs of documents.

This emphasis on collective or networked approaches is highly apposite. The Web is, by definition, an interlinked document corpus. Social networking services, also by definition, provide an additional layer of links between people that can be exploited by collective approaches. They also facilitate the sharing of links between content producers and consumers, an act which in itself creates more inter-document relationships. Collective
Chapter 1: Introduction

document classification is important in this light because of the way it is designed to take advantage of network structures. As we will see in the next chapter (Section 2.3.2), researchers have already demonstrated that hyperlinks and social networking relationships can be used to produce document classifiers that perform better than non-collective alternatives.

There has also been work to show that links that are useful for collective classification can be inferred automatically in cases where explicit links are absent (Section 2.3.3). These implicit links have the advantage of being applicable in a wider range of cases. The idea can be made more concrete by extending the earlier example relating to classifying the political perspective of blog posts. Where there are no hyperlinks between posts, an implicit link might be made between two writers who use similar, characteristic sets of words or phrases when describing an entity or idea. Even if the bias of the words or phrases themselves is unknown, the link may be useful as a hint that the two posts come from the same political perspective.

This thesis aims to provide a new and significant contribution to the body of knowledge about collective document classification. As indicated by the thesis title, I aim to distinguish approaches that apply to explicit and implicit inter-document relationships. In the section that follows I will break this goal into sub-goals, and provide an overview of how these are addressed.

1.1 Aim and Contribution

In this section I consider the contribution of this thesis to the body of knowledge about collective document classification in seven parts. In each I describe a high-level goal and describe how that goal has been met. The parts are arranged in a natural conceptual order and not according to any ranking of importance.

Provide a review of collective document classification techniques and applications

The general aim of a literature review is to prepare the reader to understand the novel content of the thesis and to correctly place the research in its context. This chapter can also serve as a general introduction to the field that can be useful even for those
Figure 1.1: A comparison of content-only and collective document classification approaches. The arrows between documents represent relationships that influence classification. The dotted line around the collective problem represents the fact that document classification is performed jointly, rather than individually.
who do not intend to read the rest of the thesis. My aim for the literature review in this thesis (Chapter 2) is to allow a new researcher to obtain the grounding necessary to begin his or her own work in collective document classification.

I also aim to clarify the discussion of collective classification by introducing new categories and concepts. My distinction between explicit and implicit inter-document relationships provide a layer of conceptual simplicity that has been lacking from the discussion of different types of inter-document relationships (Section 2.3). I also introduce the concept of a dual classifier, a missing element in the established collective classifier taxonomy of globally formulated classifiers and iterative classifiers (Section 2.4).

Provide a new experimental comparison of collective techniques

As I will discuss in Chapter 2, there are a limited range of tasks on which multiple collective classification algorithms have been trialled in identical experimental configurations. Practitioners rely on this kind of balanced comparison to be able to choose the appropriate algorithms for their tasks. Currently, there is limited guidance on which algorithms are best suited to which tasks. Chapters 4 and 5 contribute to this problem with experiments that show the relative performance of different algorithms using corpora on which they have not previously been tested. Chapter 4 experiments with the ConVote Corpus of congressional floor-debate transcripts, a corpus which has two collective benchmarks but has not been the subject of an exhaustive comparison.

Show better than state-of-the art performance

This thesis aims to validate one or more of its classifier implementations by obtaining better than state-of-the-art performance on one or more test corpora. This objective represents a clear and quantifiable success that will provide a good reason for some potential readers to look at the work. Several of the techniques I trial on the ConVote Corpus in Chapter 4 advance the state-of-the-art for performance on that corpus.

Provide a balanced evaluation of feature models for representing explicit links
As with collective classification algorithms, there is a need for more research into the relative performance of different feature models for link representation. Chapter 2 describes a range of approaches that have been the subject of limited comparative analysis. Chapter 4 improves the situation by providing a balanced comparison using experiments on the ConVote Corpus. It also evaluates several new feature models.

**Develop a new similarity-based method for implicit link construction**

The body of knowledge relating to the use of implicit links for collective document classification is much smaller than the corresponding body of knowledge for explicit links. This is partly a reflection of the fact that explicit links offer a simpler path to significant performance increases. Nevertheless, this thesis aims to show that there is significant scope for further development of implicit techniques. The literature review (Chapter 2) demonstrates what may be possible by pointing to a handful of promising but not fully realised ideas from the literature. Chapter 5 is centred around experiments that show the efficacy of simple techniques for deriving inter-document links based on measures of similarity. The chapter includes a series of experiments that justify the similarity-based approach by showing word and phrase level interactions in two test corpora. These experiments are also the first to show that there is significant benefit to be obtained by using features that may not have been seen in training data.

**Conduct an experimental comparison of collective techniques that use implicit links**

This thesis aims to fill a gap in the research by providing the first comparison of collective classification algorithms applied to implicit inter-document relationships. Because implicit links have special characteristics that can strongly differentiate them from explicit links, it is necessary to choose algorithms based on an evaluation that is formulated specifically for the purpose. I provide this in Chapter 5 which represents the first use of the Bitterlemons Corpus of opinion articles related to the Arab-Israeli conflict for collective classification.

To sum up, this goal of this thesis is to advance the science of collective document classification. I focus on a distinction between explicit and implicit inter-document relationships.
I provide a review of the existing literature that motivates this distinction and stands on its own as an introduction to the field in general. The two major sub-goals of this thesis are to improve the science of collective document classification using explicit inter-document relationships and to do likewise for collective document classification using implicit inter-document relationships. In the first major experimental chapter of this thesis I provide a new comparison of collective classification algorithms and explicit link feature models based on a standard corpus. Several of the approaches I develop yield better that state-of-the-art performance on the task. In the second experimental chapter I introduce a new approach for deriving implicit inter-document relationships based on measures of similarity. I provide an experimental comparison of collective classification algorithms for this task and demonstrate that a number of these give significant performance gains compared to content-only classification.

1.2 Thesis Structure

This thesis is organised into six chapters. This chapter and Chapters 2 and 3 provide introductory material and present the background to the work. Chapters 4 and 5 present the experiments and detailed analysis that constitute the main body of the work. Chapter 6 presents conclusions and discusses possible future work. A detailed breakdown of the remaining five chapters follows:

Chapter 2: Literature Review

In this chapter I review the existing work that is the context for the original research presented in this thesis. My aim is to prepare the reader to understand how and why my methods work and how to view my research in comparison with related research. In the early part of Chapter 2 I give a general overview of document classification, which is the overall research area in which this thesis is situated. From there, collective document classification is described and the concepts of explicit and implicit inter-document relationships are formally introduced. A detailed description of how these different types of links have been used for collective classification is also given. Finally, I describe three categories of collective classifier: (1) iterative classifiers; (2)
global methods; and (3) dual classifiers. In the final section I compare these approaches in terms of their relative performance and computational complexity.

Chapter 3: Corpora

In this chapter I introduce the corpora that will be the basis for the experimental work in this thesis. I begin by defining a set of corpus selection criteria. Two corpora are then introduced: (1) ConVote, a collection of transcripts of debates from the U.S. congress; and (2) Bitterlemons, a set of articles giving perspectives on the Arab-Israeli conflict. I discuss the origins and structure of the corpora, and show that they meet the selection criteria. I also describe the real-world document classification task that each corpus is intended to simulate and give a brief analysis of the fitness of each for that purpose.

Chapter 4: Collective Document Classification using Explicit Inter-Document Relationships

I begin Chapter 4 with a detailed description and critical analysis of the collective dual classifier classification approach that was originally applied to the ConVote Corpus. I then describe a range of alternative approaches, some of which are novel, based on dual classifier, iterative classifier and global algorithms. I motivate these in terms of how they address the limitations of the previous approach. The main body of the chapter is an experimental comparison of the original and alternative approaches. This serves two ends: (1) it provides a view of relative algorithm performance that can assist the selection of an approach for similar tasks; and (2) it shows that several of the approaches surpass all previously published methods for the task. In the final part of Chapter 4 I outline avenues for future research.

Chapter 5: Collective Document Classification using Implicit Inter-Document Relationships

In this chapter I conduct experiments in deriving useful inter-document relationships from the text of the ConVote and Bitterlemons corpora. I demonstrate that relationships can be accurately predicted using features based on matching phrases, i.e. relationships between pairs of documents that can be detected based on the mutual
use of particular $n$-grams. I go on to show experimentally that these relationships can be used to build collective classifiers that outperform standard classifiers. As with the previous chapter, I provide an experimental comparison of several different collective techniques. This opens the way for recommendations about the best algorithms for similar tasks.

Chapter 6: Conclusions and Future Work

I summarise the findings of this thesis and present recommendations for future research including: (1) applying the techniques in this thesis to new tasks; (2) improving collective classification algorithms; and (3) improving techniques for detecting inter-document relationships.
Chapter 2

Literature Review

2.1 Introduction

In this chapter I review the existing work that provides the context for the original research presented in this thesis. My aim is to prepare the reader to understand: (1) how and why my methods work; and (2) how to view my research in comparison with related research.

In Section 2.2 I give a general overview of document classification, which is the overall research area in which this thesis is situated. In Section 2.3 I describe collective document classification and introduce the concepts of explicit and implicit inter-document relationships. I also give a detailed description of how these different types of links have been used for collective classification. Section 2.4 describes three categories of collective classifier: (1) iterative classifiers; (2) global methods; and (3) dual classifiers. In the final Section (Section 2.5) I compare these three approaches in terms of their relative performance and computational complexity.

2.2 Document Classification

At the most basic level, this thesis is about document classification, so a brief introduction to that field will be helpful. Document classification, also known as text categorisation, text classification, and automated text classification, is the automatic classification of doc-
Systems for document classification fall into one of two main categories: (1) automated systems that use machine learning techniques to learn a classifier from a set of pre-classified documents; and (2) rule-based systems that use a knowledge engineering approach, where domain experts manually define the classifier. It is the former approach that is the main subject of this thesis.

Automated document classification was studied seriously in the 1960s, but took until the 1990s to grow into a major sub-field of the information systems discipline. This growth was encouraged by the increasing availability of documents in electronic form and a steady decrease in the cost of computational resources.

Compared with rule-based systems, the advantages of using machine learning techniques are: accuracy approaching that of human experts; considerably reduced cost in terms of human effort; and easy translation of effort from one target domain to another (Sebastiani 2002).

The range of potential applications of document classification is as large as the number of useful document categories that can be imagined. Early work focused on a set of newswire articles from the Reuters News Service, which were classified into a set of commerce-related topic categories such as CORN, CRUDE, EARN, and TRADE (e.g. Apté et al. 1994, Joachims 1998, Lam and Ho 1998).

More recent and varied applications include classifying the mood of blog posts (Mishne 2005), distinguishing satirical from genuine news articles (Burfoot and Baldwin 2009), distinguishing between authors (Escalante et al. 2011), and detecting vandalism on Wikipedia (Harpalani et al. 2011). Since the early 2000s there has been an explosion of interest in sentiment classification, the task of classifying the opinion content of a document as either positive or negative (Pang and Lee 2008).

Document classification assumes we are given a tuple \( T = (D, X, Y, L) \) where \( D \) is a set of documents, each \( x_i \in X \) is an attribute vector for document \( d_i \in D \), each \( y_i \in Y \) is a label variable for \( d_i \), and \( L \) is a set of labels. The task is to infer the values for \( Y \).

Constraints control the selection of labels from \( L \) to be assigned to each \( Y_i \) variable.\(^\text{1}\)

\(^{1}\)For a brief discussion of unsupervised document classification see Section 2.2.1
Multi-label document classification has the constraint that each $Y_i$ may be a set of labels of any size. Single-label document classification requires that exactly one label be selected for each document. Binary document classification is a special case of single-label classification, where the classifier must choose between two labels, $l$ and the complement $\bar{l}$.

Binary document classification is the most represented in the literature because: (1) it is conceptually the simplest; and (2) single-label and multi-label document classification can be approached as special cases of binary classification by framing the classification problem as a sequence of binary tasks (e.g., Fujino et al. 2008, Duan and Keerthi 2005).

2.2.1 Document Classification Terminology

Strictly speaking, the type of document classification referred to here is supervised document classification (Jackson and Moulinier 2002, Manning et al. 2008). This is to distinguish it from so-called unsupervised document classification, which does not use labelled examples. Unsupervised classifiers try to group documents into logically distinct groups, or clusters, without reference to a pre-defined category definition. They are particularly useful in cases where labelled training data is difficult or impossible to obtain (Jain et al. 1999).

Semi-supervised document classifiers represent the middle ground between the supervised and unsupervised options. They rely on a small amount of unlabelled training data and try to make up the difference by using unlabelled examples or outside data from sources like ontologies and large text databases (Abney 2007).

This thesis will focus on supervised document classification only, because the techniques used to approach these three distinct types of document classification are disparate.

2.2.2 Practical Considerations for Document Classification

Indexing is the process of converting a document from its original form into a form that can be processed by a machine learner. Normally, documents are represented by vectors of features (also known as term weights), where the value of the feature represents the extent to which the term contributes to the semantics of the document.
By far the most common indexing strategy is the bag-of-words model, which interprets individual words as separate terms. [Sebastiani (2002)] summarises a number of early works which trialled more sophisticated representations and found inferior performance to bag-of-words. This effect comes from the problem of data sparseness, where more sophisticated terms are less statistically useful because they tend to occur more rarely across the corpus.

A side-effect of many indexing strategies, including bag-of-words, is that large numbers of terms are produced. For the majority of machine learning algorithms this has two negative effects: (1) the time required for classifier induction becomes impractically large; and (2) the probability of overfitting is increased. Overfitting refers to the phenomenon where a classifier is tuned to characteristics that are particular to the training data rather than the category in general. This results in very good performance when classifying instances from the training set, but much poorer performance when classifying any other instances [Jackson and Moulinier 2002; Manning et al. 2008]. One solution is dimensionality reduction, where an algorithm is used to systematically reduce the feature space [Blei et al. 2003, Tan et al. 2006].

### 2.2.3 Support Vector Machines

The support vector machine (SVM) classification method is currently the dominant approach to document classification. In simple terms, an SVM is a hyperplane that separates positive and negative training examples with a maximum margin. The hyperplane, or “decision surface” is determined by a small set of the training examples, which are referred to as support vectors. The output of an SVM classifier is a signed number representing the distance between the sample and the decision plane. By convention this is simplified to -1 representing the negative class and 1 representing the positive class.

In their basic form SVMs use a linear kernel, but they can also be made to learn more complex decision functions through the use of alternative kernel functions. SVMs were applied to text classification by [Joachims (1998)] who demonstrated that polynomial and radial basis function (RBF) kernel SVMs provided superior performance to popular alternatives such as naive Bayes, Rocchio, and k-nearest neighbour on a range of standard test corpora. Other work has shown that linear kernel SVMs provide similar benefits (e.g.

Joachims identified two key characteristics of SVMs that made them suitable for document classification:

1. SVMs are able to handle the very large feature spaces characteristic of document classification without overfitting and without the need for dimensionality reduction. The benefit of this facility is more than just convenience: an information theoretic analysis using the Reuters Corpus showed that even the most lowly ranked features still carried information about document labels. This result indicates that dimensionality reduction for document classification is the lesser of two evils, where potentially useful information is thrown away to avoid overfitting.

2. There is theoretical and empirical evidence that SVMs perform well with the sparse document vectors that are characteristic of document classification tasks.

SVM implementations have a theoretically motivated set of default parameters that obviate the need for computationally expensive parameter tuning. They also have good computational efficiency. Joachims (2006) introduced an algorithm that could be used to train linear SVMs on large datasets in linear time. The algorithm has a theoretical and empirical training time that is $O(sn)$ for training sets with $n$ instances, $N$ features, and $s \ll N$ non-zero features in each instance.

Performance when evaluating new instances is comparable to other methods and insignificant when compared to the cost of training. In one test, the average time to classify a test document using a linear SVM was approximately 2 milliseconds (Dumais et al. 1998).

### 2.3 Collective Document Classification

In this section I provide a formal definition of collective document classification. I then flesh out the definition by giving a detailed overview of existing collective document classification tasks, focusing on the types of documents classified and the nature of the inter-document relationships used.
Automated document classification techniques take advantage of the correlations between the features of documents and their unobserved labels. Collective document classification expands this scope of correlation to include document “neighbourhoods”. A document neighbourhood is the set of other documents that are related to that document in some specified way. Collective document classifiers can take advantage of two types of correlations relating to document neighbourhoods (Sen et al. 2008):

1. The correlations between the label of a document and the observed attributes, including labels, of neighbour documents.
2. The correlations between the label of a document and the unobserved labels of neighbour documents.

2.3.1 Definition, Notation and Terminology

Recall from Section 2.2 that standard document classification assumes we are given a tuple $T = (D, X, Y, L)$ where $D$ is a set of documents, each $\vec{x}_i \in X$ is an attribute vector for document $d_i \in D$, each $Y_i \in Y$ is a label variable for $d_i$, and $L$ is a set of labels. The task is to infer the values for $Y$.

Collective document classification augments this tuple with a set of edges, $E$, so we are now given a graph $G = (D, E, X, Y, L)$. Assume also that we are given a set of known values $Y^K$ for documents $D^K \subset D$, so that $Y^K = \{Y_i|d_i \in D^K\}$. The task of collective document classification is to infer $Y^U$, the values of $Y_i$ for the remaining documents with “unknown” labels ($D^U = D - D^K$) (McDowell et al. 2009).

For convenience of notation, assume that $N$ is a neighbourhood function giving the link structure of the graph, where $N_i \subseteq D \setminus \{d_i\}$. Learning takes place on a training graph, $G_{Tr}$.

Note that for the remainder of this document I will use the terms link, edge and relationship interchangeably to refer to the edge information $E$ used in collective classification. Note also that despite the fact that all of the classification algorithms addressed in this thesis are single-label approaches (i.e. one label per document), I will continue to use an uppercase $Y_i$ to represent the label variable for the sake of consistency.

Collective document classification can be applied in two distinct settings:
1. **Out-of-sample setting.** In the out-of-sample setting, the set of known labels, $Y^K$ is empty. This setting is most common when working in domains where the available data is naturally broken down into disjoint graphs.

2. **In-sample setting.** In this setting an arbitrary proportion of the labels are known. This setting is most applicable to domains that have a single, interconnected document graph.

Figure [2.1] shows an example of both settings. In both cases, assume that the classifier has already been trained on a set of labelled examples. For the in-sample example, the labels for the first two documents in the graph are given, and it is the job of the classifier to deduce the labels for the remaining three. In the out-of-sample example none of the labels are given and the classifier must deduce labels for all five documents.

I now move on to a detailed overview of established tasks in collective document classification. This section is effectively a survey of collective document classification tasks, with an emphasis on the types of links used.

### 2.3.2 Collective Document Classification with Explicit Links

The significant majority of experiments in collective classification have focused on document domains where the links between documents are explicit, i.e. where one document links to, refers to or quotes another. There are four major types of explicit links represented in the literature: (1) links in hypertext documents; (2) citations in academic papers; (3) links in social publishing sites; and (4) by-name references.

Early collective document classification work focused on classifying hypertext documents, with hyperlinks providing the graphical structure. [Chakrabarti et al. (1998)](chakrabarti1998) take link analysis ideas from information retrieval and apply them to supervised document classification in a manner consistent with our definition. They use two test corpora: (1) a set of patents from the IBM Patent Server; and (2) a set of web pages from the Yahoo! directory. Both corpora have multi-level hierarchies based on topic. [Chakrabarti et al.](chakrabarti1998) use techniques that make use of the neighbourhood structure created by links within the test corpora in three different ways:
Figure 2.1: Example in-sample and out-of-sample settings for a collective classification problem. The labels for the first two documents in the in-sample example are given. No labels are given in the out-of-sample setting.
1. **Inference based on given neighbour labels.** This approach embodies the view that there is a correlation between the label of a document and the labels of the document that it links to or which link to it. It assumes that the classifier is given reliable labels for the neighbourhood documents. This is not a collective approach, because it does not consider relationships between unlabelled documents.

2. **Inference based on words in neighbour documents.** This approach embodies the view that there is a correlation between the label of a document and the words in the documents it links to or which link to it. It assumes that the classifier is not given labels for the neighbour documents. This is the out-of-sample setting.

3. **Inference based on words in neighbour documents and some given neighbour labels.** This approach is a combination of the first two approaches. It assumes that only a small number of neighbour labels are known. This is the in-sample setting.

  Slattery and Craven (1998) experiment on a corpus of university web pages, with the aim of labelling pages in one of the categories Student, Course, Faculty or Project. The network structure is derived from hyperlinks and an out-of-sample setting is used. This corpus has become something of a standard for collective classifier evaluation (Taskar et al. 2003b; Macskassy and Provost 2007; McDowell et al. 2009). Joachims et al. (2001) work on the same corpus but use link structure based on co-citation. This means that a pair of documents is related if they link to one or more of the same documents, regardless of whether or not one links to the other. There are a range of other works dealing with this corpus (Taskar et al. 2002; Taskar et al. 2003a; McDowell et al. 2007a; McDowell et al. 2007b; McDowell et al. 2010; Crane and McDowell 2012).

  Oh et al. (2000) use a corpus of articles from a web-based encyclopedia broken into a large number of topic-based categories. Their method uses the standard approach of relating documents where one contains a hyperlink to the other. Like Chakrabarti et al. (1998), they experiment with taking differing proportions of neighbour labels as given. The link structures in the two corpora are assumed to be strongly influenced by **homophily**, i.e. the tendency for links to occur between documents with the same labels.
Yang et al. (2002) give a survey of work in hyperlink-based collective classification. They also conduct experiments of their own using a subset of the university web page corpus described above and the Hoover Corpus, a collection of company homepages labelled according to industry categories, e.g. Sporting Goods, Oil and Gas, Computer Software and Services. Yang et al. identify three “regularities” by which the internal link structure of a corpus can assist classification of a document:

1. **Encyclopedia.** The neighbours of a document all have the same class as that document. This kind of regularity might be expected when performing topic-classification on a corpus of encyclopedia articles, because articles tend to link to others from similar topics.

2. **Co-referencing.** The neighbours of a document all have the same class, different to the document’s class. For example, a department home page might have links to courses.

3. **Pre-classified.** A single document points only to all documents of a given class. A common example of this pattern is an online bibliography that tracks all of the published research specific to a given topic.

Separate classifiers are built to take advantage of each type of information, with neighbour labels taken as hidden. This emphasis on different types of links is particularly significant because of the tendency of other work to focus only on the encyclopedia regularity, where links are taken as an indication that the source and target documents have the same class.

After hypertext documents, academic papers have had the most attention as targets for collective classification. The Cora and CiteSeer corpora have been used by researchers to show the ability of collective classifiers to use the information contained in citation networks to improve topic-based classification of academic papers (Giles et al. 1998; McCallum et al. 2000). The Cora Corpus consists of a number of machine learning papers divided up into seven sub-categories (e.g. Neural Networks, Probabilistic Methods, Rule Learning). CiteSeer is a collection of computer science papers divided into six categories (e.g. Artificial Intelligence, Agents, Information Retrieval).
Lu and Getoor (2003a) demonstrate models for incorporating citation information into a collective classifier, which they evaluate on the Cora Corpus.

Angelova and Weikum (2006) build a classifier for their own corpus of academic papers, similar to Cora, for the out-of-sample setting. They use co-authorship instead of citations for links, creating a relationship between papers that have one or more authors in common.

A series of works by Luke McDowell and collaborators feature in-sample collective classification on the Cora and CiteSeer corpora (McDowell et al. 2007a; McDowell et al. 2007b; McDowell et al. 2009; McDowell et al. 2010; McDowell and Aha 2012). This task is also undertaken by Sen et al. (2008).

After hypertext and academic citations, social network links have been the most significant target domain for collective document classification experiments. Jiang et al. (2011) build a collective classifier to determine the sentiment of tweets (posts) on the Twitter social networking service, where the objective is to determine whether the tweet author is expressing a generally positive or negative opinion about the topic of their post. A set of five hand-chosen topics is used: “Obama”, “Google”, “iPad”, “Lakers”, “Lady Gaga”. A range of non-collective techniques specific to the field of sentiment analysis are first applied, then final classifications are computed collectively by placing the tweets in a graph, where links are added between pairs of tweets that satisfy one of their criteria specific to Twitter:

1. One of the tweets is a direct reply to the other.

2. Both of the tweets are by the same author.

3. One of the tweets is a retweet (attributed re-publication) of the other.

Tan et al. (2011) attempt the task of user-level Twitter sentiment classification on a similar list of hand-chosen topics. They demonstrate a statistical correlation between two types of Twitter relationships and user-level sentiment agreement, which is then used to build a collective model:

1. One of the tweet authors has sent a direct message to the other.

2. One of the tweet authors follows (subscribes to tweets by) the other.
By-name references are the final category of explicit links represented in the literature on collective document classification. A by-name reference can be understood as an informal form of a citation. Instead of providing detailed bibliographical information to give an unambiguous pointer to a source work, a by-name reference consists simply of a reference to another document by some common name. The identity of the document can normally be determined by interpreting the by-name reference in context.

Work on by-name references has focussed on the ConV ote Corpus of U.S. Congressional Debate transcripts (Thomas et al. 2006; Bansal et al. 2008; Burfoot 2008; Burfoot et al. 2011; Stoyanov and Eisner 2012). These are transcripts of spoken debates in which members of Congress consider the merits of specific articles of legislation. The object of the task is to assess the contribution of members to determine whether they will vote for or against the legislation in question. By-name references occur when members refer to each other in the course of debate, either by full name or title. The task is framed in the out-of-sample setting, with distinct sets of debates used for training and testing. I deal with this work in detail in Chapter 4.

2.3.3 Collective Document Classification with Implicit Links

There is a smaller body of collective document classification research that has focussed on tasks where the links between documents are not explicit. Before collective techniques can be applied to such tasks, inter-document relationships need to be inferred using some non-trivial process. There are two basic ways to do this:

1. Infer inter-document relationships using the concept of proximity. Many document domains have a built-in spatial or temporal dimension that allows documents to be related in terms of how close or far apart they are. Proximity-based inter-document relationship detection works on the premise that the labels of the most proximate documents are likely to be related.

2. Infer inter-document relationships using the concept of similarity. If documents do not have obvious spatial relationships, they can still be related using measures of similarity. The intuition here is simple: the more similar two documents are, the
more likely it is that they will have the same label.

Agrawal et al. (2003) develop a technique for classifying the political orientation of posts in online newsgroup discussions. They perform binary classification in the topic areas “abortion”, “immigration” and “gun control”. A simple measure of proximity is induced from the thread structure of the newsgroups. Posts are considered to be proximate if one occurs after the other within a thread. Analysis is presented to show that these relationships very commonly indicate label disagreement.

Goldberg et al. (2007) conduct a similar experiment using a set of posts to the online political discussion site, Politics.com (Mullen and Malouf 2006). They add links between posts where one quotes content from the other in the immediate context of two question marks or exclamation marks, or two or more consecutive words in caps. These links are shown to correlate with label disagreement.

Pang and Lee (2004) use the concept of “coherence” to build a collective sentence subjectivity classifier. The classifier is designed to judge whether or not a given sentence contains primarily subjective or objective content. Links are added between sentences based on how far apart they are in the source document, implementing the intuition that sentences that are closer to each other are more likely to have the same label.

In one of several experiments, McDowell et al. (2009) build a classifier to label written descriptions of terrorist events with one of six event-type categories (Bombing, Kidnapping, Arson, etc.). The links between documents are based on the reported geographic location of the event and represent the intuition that particular types of events are characteristic to locations.

A final example of collective classification making use of proximity-based relationships comes again from the field of sentiment analysis. Somasundaran et al. (2009a) use collective techniques to perform opinion polarity classification on annotated transcripts of meetings from the AMI Corpus (Carletta et al. 2005). The corpus includes seven multi-party meetings with complex scenarios, e.g. discussing and designing a new TV remote. Discourse-level links are used to relate opinion statements, so that final classifications take into account not just the contents of the opinion statements, but their relationships with other, related opinion statements.
Given the diversity of these six papers, it might be reasonable to argue that the concept of proximity-based links is too vague to be useful. Nevertheless, this grouping serves the second useful function of separating the papers from the remainder in this section, all of which use are similarity-based.

Agarwal and Bhattacharyya (2005) develop an out-of-sample binary movie review collective classifier that judges whether the review has a generally negative or generally negative sentiment. They use a fully-connected link graph, with link weights that are proportional to the similarity of the two reviews. The measure of similarity is based on a simple count of overlapping terms from the bag-of-words representations of the reviews.

Takamura et al. (2007) build an out-of-sample ternary collective classifier for Japanese adjective/noun pairs that judges whether the pair has a negative, positive or neutral semantic orientation. Each of the adjectives for a given noun is classified together, with links added between each pair of adjectives where one has the other in its dictionary gloss.

Transductive semi-supervised document classification can be understood as a form of similarity-based in-sample collective document classification. Where a standard (inductive) learner induces a generalised model based on the training examples, a transductive learner reasons directly from a set of labelled and unlabelled examples to produce a labelling for the unlabelled examples (Abney 2007). For example, a transductive binary document classifier may cluster the labelled and unlabelled (test) examples into two clusters to guide label assignments. Collective classification techniques also rely on relationships between labelled and unlabelled examples, though, as will be shown later in this chapter, they are not necessarily transductive.

In a crucial early paper, Blum and Chawla (2001) demonstrate techniques for transductive semi-supervised classification using a range of standard machine learning datasets. They trial measures of similarity based on hamming distance, euclidean distance, and information gain.

Joachims (2003) attacks the same problem using cosine similarity (Manning et al. 2008) and evaluates on a range of tasks including topic classification of newswire articles and the university web page type detection task described in the previous section.

Goldberg and Zhu (2006) build a transductive semi-supervised document classifier for the movie review sentiment classification corpus described above. Instead of binary clas-
sification, they attempt the rating-inference problem, where the goal is to learn a numeric rating for the review such as the number of “stars”. They trial four similarity measures: (1) cosine similarity between bag-of-words document vectors; (2) cosine similarity between document vectors consisting only of terms with high mutual information; (3) cosine similarity between bag-of-words document vectors weighted based on mutual information; and (4) positive sentence percentage (PSP). PSP is a similarity measure of sentiment content originally proposed by Pang and Lee (2005). It assumes the availability of a sentiment classifier that can label each sentence in a document as expressing neutral sentiment, expressing negative sentiment, or expressing positive sentiment. PSP is defined as the number of positive sentences in the document divided by the number of non-neutral sentences.

Sindhwani and Melville (2008) build a transductive semi-supervised sentiment classifier that takes advantage of lexical prior knowledge from a corpus of positive and negative words. Documents are considered to be related if they share a similar set of terms from this corpus. This avoids a potential problem where document contents may overlap in ways that are not strongly related to document labels. The approach is trialled on three corpora: (1) a set of positive and negative blog posts about Barack Obama and Hilary Clinton; (2) a set of positive and negative blog posts about IBM’s Lotus line of Enterprise collaboration products; and (3) the movie review corpus discussed earlier.

These semi-supervised approaches are distinct from the other examples discussed in this section because of the way they are designed to work in scenarios where content-only performance is very low. This characteristic allows them to make effective use of mostly standard measures of similarity. Amongst the transductive approaches, the PSP metric stands out as the one measure of similarity that is derived from a complex process that produces inter-document links that are not based on the same feature model used by the content-only classifier. The Agarwal and Bhattacharyya paper stands out as the only fully supervised technique that uses a similarity-based approach, but this also lacks a distinction between the feature models used for content-only classifier and relationship representations, so it is somewhat surprising that performance is good. I deal with this question further in Section 5.6.

Experimental results have showed some improvement over a content-only approach for each of the different types of implicit explicit inter-document relationship discussed in this
section. Section 2.5 has an analysis of the relative performance of different approaches.

This section has surveyed the range of established collective document classification tasks, and in the process elucidated the difference between explicit and implicit inter-document relationships. New experiments with collective document classification using explicit links and implicit links will be addressed in Chapters 4 and 5 respectively. I now move to a discussion of the algorithms that are the basis for the classifiers used for both types of task.

2.4 Collective Classification Algorithms

In this section I describe algorithms for collective classification. I begin by describing iterative classifiers, which are the most conceptually simple approaches. I then introduce Markov Random fields and describe two families of techniques based on these models: (1) global techniques; and (2) dual classifier techniques. Finally, I describe a flow-based dual classifier technique.

My descriptions (and the remainder of this thesis) will assume the out-of-sample setting for two reasons: (1) this is the most broadly applicable setting because it does not assume that any labelled test data is available; and (2) this setting allows for simple comparison with content-only approaches, because there are no labelled test examples to account for. For each of the three groups of techniques there are simple algorithmic additions to support the in-sample setting.

2.4.1 Iterative Classifiers

Iterative classifiers adapt the standard automated document classification approach for use in collective document classification. They use the standard document vector model with basic machine learning algorithms like SVM.

Iterative classifiers use an additional set of feature vectors \( F \) where each \( \vec{f}_i \in F \) is a relational feature vector for document \( d_i \in D \). The local classifier \( M_{AR} \) accepts both the document feature vector and relational feature vector for a given instance, with the classification for a given instance specified as \( M_{AR}(\vec{x}_i, \vec{f}_i) \).
Figure 2.2: A toy binary collective document classification training problem with four connected documents. The documents have a three element vector representation and document labels are known. Alternative relational feature vectors for positional mapping and count aggregation are shown. Diagrammatic style is adapted from McDowell et al. (2009).

The key difficulty in making this approach work is finding a way to represent the network structure of a collective classification problem as a vector. Consider the toy document classification shown in Figure 2.2. The figure shows four connected documents with attribute vectors of length 3. Perhaps the most obvious way of representing the inter-document relationships in the relational vectors $F$ is via positional mapping. In this scheme, one feature is reserved to indicate connectivity to each instance in the collection. So for the instance $d_1$ the relational feature vector $\vec{f}_1 = (0, 1, 0, 0)$, indicating that the document is connected to $d_2$ and not connected to $d_3$ or $d_4$.

This approach has obvious problems. First, it does not generalise. A local classifier trained with this feature model could not be applied to any document collection of more or less than four documents, because the size of the document collection is embedded in the feature vector. Second, these features do not appear to embody any helpful knowledge. It is not useful to know the identities of linked documents if we do not also have some knowledge of the contents of those documents.

The alternative to a mapping scheme that embeds the whole network structure in relational feature vectors is an aggregation scheme that distills the network information into a fixed number of features. The simplest and most popular aggregation scheme for local clas-
sifiers is count aggregation, where the relational feature vector uses one element to store the number of neighbour nodes in each category. Figure 2.2 also illustrates this scheme. The relational feature vector has two variables: the first to count the number of neighbours in the positive class and the second to count the number of neighbours in the negative class. So for the instance \( d_1 \) the relational feature vector \( \vec{f}_1 = (0, 1) \), which indicates that the document is connected to 0 instances in the positive class and 1 instance in the negative class \( (d_2) \). I review the full range of aggregation schemes for iterative classifiers in Section 2.4.2.

At this point the answer to the question of how to execute the learning algorithm for a local classifier becomes evident. First, the aggregation scheme is selected. Then the relational feature vectors are computed and the attribute vectors and feature vectors are combined for each instance and passed to the SVM, logistic regression, naive Bayes or other learner, which then execute as normal. What is not clear is how the learned classifier is to be applied to unseen data. In particular, it is not clear how relational feature values are derived when instance labels are unknown.

The solution, not surprisingly given the name of the algorithm, is to use an iterative approach. Classifications are first bootstrapped using a classifier trained only on the attribute vectors, \( M_A \). These output classifications are used to calculate relational feature vectors, which are passed to the local classifier, \( M_{AR} \) to recompute the classifications. These outputs are then used to update the relational features, and the process repeats until the classifications stabilise or a fixed number of iterations is completed. A more formal definition is given in Algorithm 1 and a diagrammatic overview is shown in Figure 2.3. Figure 2.4 illustrates the process of iterative classification on the toy dataset used in the previous example.

### 2.4.2 Feature Models for Iterative Classifiers

Having established the basic theory of iterative classifiers, I now move on to analysis of the different feature models that have been used for iterative classifiers.

In an early and highly cited work in the field, Neville and Jensen (2000) develop an iterative classifier and apply it to classifying companies based on industry type. This work introduces a helpful distinction between different types of relational features. *Static relational attributes* do not change their values as the labels of related object change. *Dynamic
Algorithm 1 Iterative classification

\[ D = \text{documents}, \, X = \text{attribute vectors}, \, F = \text{relational feature vectors}, \, Y = \text{labels} \]

\[ M_A = \text{content only classifier}, \, M_{AR} = \text{local classifier} \]

\[ N_i = \text{set of neighbours for document } d_i \]

\begin{algorithm}
\begin{algorithmic}
\FOR {each node } \( d_i \in D \) \DO [bootstrapping]
\STATE \( Y_i \leftarrow M_A(\vec{x}_i) \)
\ENDFOR
\REPEAT [iterative classification]
\FOR {each node } \( d_i \in D \) \DO 
\STATE compute \( \vec{f}_i \) using current assignments to neighbours \( N_i \)
\ENDFOR
\FOR {each node } \( d_i \in D \) \DO 
\STATE \( Y_i \leftarrow M_{AR}(\vec{x}_i, \vec{f}_i) \)
\ENDFOR
\UNTIL {labels have stabilized or maximum iterations reached}
\end{algorithmic}
\end{algorithm}

relational attributes, the more common type, need to be updated as classifications change, and so depend on the iterative behaviour of the classifier.

Lu and Getoor (2003a) develop an iterative document classifier that defines three distinct types of features based on the labels of connected instances. Each type defines a set of features that correspond one-to-one with the label set, \( L \). The features have default value 0 for all instances except where their activation condition occurs:

- For mode features, set the value 1 for the feature that corresponds to the category that is most frequently represented in the set of neighbour label assignments.

- For count features, set the value to the count of the neighbours that have the corresponding label assignments.

- For binary features, set the value to 1 if there are one or more neighbours with the corresponding label assignment.

Testing on a corpus of hyperlinked web pages and two corpora of citation-linked aca-
demic papers, Lu and Getoor find that count-link and binary-link features perform equivalently and mode-link features are slightly inferior.

This work also introduces a distinction between features that represent in-links, where the neighbour document is the source of the link, out-links, where the neighbour document is the target of the links, and co-citation links, where both documents contain an out-link to a particular document. A related work by the same authors covers the same ground in the in-sample setting (Lu and Getoor 2003b).

The same tasks are tackled by McDowell et al. (2007a) who use binary features and have the distinction of being the first to perform iterative classification using a \( k \)-nearest neighbour base classifier. This work also makes uses of a threshold for links, where the relational features are only set in cases where \( n \) or more neighbours have the given label assignment.

Neville et al. (2003) introduce the concept of a multiset relational feature, which can
1. Initial state.

2. Apply content-only classifier (set $Y$).

3. Compute relational feature values (set $F$).

4. Apply local classifier (set $Y$ and reset $F$).

5. Re-compute relational features (set $F$).

(repeat steps 4 and 5 . . . )

Figure 2.4: A toy binary collective document classification problem with four connected documents. The documents have a three element vector representation and all document labels are unknown. Relational feature vectors are calculated via count aggregation. Observed (c.f. inferred) variables are indicated by dark backgrounds. Adapted from McDowell et al. (2007b).
represent the values of all of the neighbours of a given instance. For example, a multiset feature might take the value \{1, 1, 0, 0\} to represent the fact that the first two neighbours are assigned to the positive class and the second two to the negative class. During naive Bayes inference, each label in the multiset is separately used to update the conditional probability that an instance has a given label. This approach does not apply to other types of base classifier like SVM or \(k\)-nearest neighbour. [McDowell et al. (2010)] successfully apply multiset features in experiments on the standard academic papers and hypertext collective classification tasks.

[McDowell et al. (2009)] compare count features, multiset features and proportion features. Like mode, count and binary features, proportion features are based on a simple aggregation scheme that generates one feature for each label in the label set. The feature is set to a value between 0 and 1 that is equal to the proportion of neighbours that have the given label assignment.

[McDowell et al.] use cross-validation to choose the best feature representation for a range of real and synthetic collective classification tasks, and report the relative numbers of experiments for which each representation is chosen. They find that multiset features are easily the best performing relational feature type when using naive Bayes. For logistic regression, count features are chosen roughly twice as often as proportion features. For \(k\)-nearest neighbour, proportion features dominate.

Several other papers apply proportion features [McDowell et al. 2007b, McDowell and Aha 2012], count features (Sen et al. 2008), or both (Somasundaran et al. 2009a, Somasundaran et al. 2009b).

[Bilgic et al. (2007)] attempt the link-prediction problem, where the goal is to find real relationships between instances, rather than relying on known relationships to improve base classification. They use an iterative classifier with count features.

### 2.4.3 Markov Random Fields

An alternative to the iterative approach just described is to formulate a collective classifier based on a global objective function. The most common global approach to collective classification treats the entire document graph, including attributes, as a pairwise Markov
random field (Sen et al. 2008).

As before, let \( G = (D, E, X, Y, L) \) specify our document graph, where \( D \) is a set of documents, \( E \) is a set of edges, each \( \vec{x}_i \in X \) is an attribute vector for document \( d_i \in D \), each \( Y_i \in Y \) is a label variable for \( d_i \), and \( L \) is a set of labels. Our task is to infer the values of \( Y_i \) for the all documents.

Note that it is possible to formulate Markov random fields for scenarios where some portion of the documents have known labels, i.e. the in-sample setting. I omit this for the reasons described in the introduction to Section 2.4.

A pairwise Markov random field is given by the pair \( (G, \Psi) \), where \( \Psi \) denotes a set of clique potentials. There are two distinct types of functions in \( \Psi \):

1. For each \( d_i \in D \), \( \psi_i \in \Psi \) is a function \( \psi_i : L \rightarrow \mathbb{R}^+ \cup \{0\} \). In our context this function represents the “non-relational evidence” for \( d_i \) having class \( l \). It produces values that take into account only the content of the document in question, ignoring any relationships with other documents.

2. For each \( (d_i, d_j) \in E \), \( \psi_{ij} \in \Psi \) is a function \( \psi_{ij} : L \times L \rightarrow \mathbb{R}^+ \cup \{0\} \). This is a “compatibility function”, in our context representing the likelihood that the two documents are linked.

Figure 2.5 shows the collective document classification example from the previous section re-formulated as a pairwise Markov random field. Instead of showing observed document features, the nodes have \( \psi_i \) values indicating their non-relational preference to be classified in either the positive or negative class. In this case a total numeric weight of 10 is distributed between the two classes. If it were desirable to think of the \( \psi_i \) values probabilistically, the total numeric weight could be set to 1 instead. For example, the interpretation of the model is identical if \( \psi_1(1) \) and \( \psi_1(0) \) are 0.6 and 0.4 instead of 6 and 4.

The arcs now have tables of \( \psi_{ij} \) values indicating the preference of the nodes to receive classifications relative to each other, derived from some source of inter-document relationship information that is left unspecified for the purpose of this example. Note that in each table the value of \( \psi_{ij}(0, 1) \) is the same as the value for \( \psi_{ij}(1, 0) \), and the value of \( \psi_{ij}(0, 0) \) is the same as the value for \( \psi_{ij}(1, 1) \). This reflects the fact that inter-document relationship...
preferences normally only differentiate between “same label” and “different label”. For example, $\psi_{1,2}(0, 0) = \psi_{2,1}(1, 1) = 3$ and $\psi_{1,2}(0, 1) = \psi_{2,1}(1, 0) = 7$. This indicates that the preference is for nodes $d_1$ and $d_2$ to receive different labels. As with the $\psi_i$ values, it is the ratio of values for different labels, not their absolute values, that indicates preference. Another important special case is shown for $\psi_{2,4}$, which has the same value (5) for all label combinations, indicating the lack of any “same label” or “different label” preference.

The crucial difference between the Markov random field representation for a collective classification problem and the iterative classifier representation is that the former represents each inter-document link explicitly, while the latter relies on aggregation functions to summarise link information.
2.4.4 Classifying with Markov Random Fields

A pairwise Markov random field defines a probability distribution for a given set of label assignments, $Y$:

$$ P(Y) = \frac{1}{Z} \prod_{d \in D} \psi_i(Y_i) \prod_{(d_i,d_j) \in E} \psi_{ij}(Y_i,Y_j) \tag{2.1} $$

where

$$ Z = \sum_{Y'} \prod_{d \in D} \psi_i(Y'_i) \prod_{(d_i,d_j) \in E} \psi_{ij}(Y'_i,Y'_j) \tag{2.2} $$

and $Y'$ denotes the complete set of possible alternative labellings.

Given this distribution, it is conceptually simple to extract the best labelling by choosing the one that corresponds with the highest marginal probability. Unfortunately this approach is computationally impractical because of the need to sum over an exponentially large number of alternative labellings.

Nevertheless, there are computational optimisations that have allowed efficient inference over a range of special cases, with linear-chain conditional random fields (CRFs) being the most prominent (e.g. Lafferty et al. 2001, Zhao et al. 2010, Ghosh et al. 2011).

Unfortunately none of the optimisations is able to deal with so-called loopy models, which have cyclic connections between nodes. Figure 2.5 is an example of such a loopy model. While sequencing tasks are often a natural fit for linear-chain CRFs, many document domains are likely to contain loops because there are few rules governing link structure. Academic papers are an exception: citation graphs normally have no loops because only past work is included. Web documents (e.g. homepages) have no such limitation because of their “live” and continuously updated nature. Similarly, by-name references in a debate almost always create cycles because speakers can contribute to the same debate multiple times and tend to refer to each other reciprocally.

Fortunately, there are two well-known approximate solutions for pairwise Markov random fields that can deal with loopy structures. Loopy belief propagation was used in early collective classification work (Taskar et al. 2002) and has remained popular since (Sen et al. 2008, McDowell et al. 2009, Stoyanov and Eisner 2012).
Applied to a pairwise Markov random field, loopy belief propagation is a message passing algorithm that can be concisely expressed as the following pair of equations (McDowell et al. 2009):

\[ m_{i \rightarrow j}(l) = \alpha \sum_{l' \in L} \psi_i(l') \psi_{ij}(l', I) \prod_{k \in N_i \cap D \cup \{j\}} m_{k \rightarrow i}(l') \]  

(2.3)

\[ b_i(l) = \alpha \psi_i(l) \prod_{k \in N_i \cap D \cup \{j\}} m_{k \rightarrow i}(l) \]  

(2.4)

where \( m_{i \rightarrow j} \) is a message sent by \( d_i \) to \( d_j \) and \( \alpha \) is a normalization constant that ensures that each message and each set of marginal probabilities sum to 1. The message flow from \( d_i \) to \( d_j \) communicates the belief of \( d_i \) about the label of \( d_j \). The algorithm proceeds by making each node communicate with its neighbours until the messages stabilise. The marginal probability is then derived by calculating \( b_i(l) \).

Algorithm 2 is a pseudo-code description of loopy belief propagation intended to given an understanding of how the algorithm is implemented (Sen et al. 2008).

Mean-field is another approximate algorithm that can be applied to loopy pairwise Markov random fields. It is a message passing algorithm, like loopy belief propagation, that can be expressed as follows:

\[ b_i(l) = \alpha \psi_i(l) \prod_{j \in N_i \cap D} \prod_{l' \in L} \psi_{ij}^{h(l')}(l', I) \]  

(2.5)

The mean-field algorithm simply re-computes this equation for each document until the marginal probabilities stabilise. Pseudo-code is shown in Algorithm 3.

Loopy belief propagation and mean-field have both been justified as variational methods for Markov random fields (Jordan et al. 1999; Weiss 2001; Yedidia et al. 2005).

2.4.5 Learning with Markov Random Fields

The algorithms described in the last section rely on the potential functions \( \psi_i \) and \( \psi_{ij} \). These functions are not suitable for generalising from training data because they are document-specific: training a classifier requires the use of a feature model. To work
Algorithm 2 Loopy belief propagation

// $D =$ documents, $E =$ links, $N =$ neighbourhood function

// $L =$ label set, $\psi_i, \psi_{ij} =$ clique potentials

for each $(d_i, d_j) \in E$ do {initialisation}

for each $l \in L$ do

$m_{i \rightarrow j}(l) \leftarrow 1$

end for

end for

repeat {message passing}

for each $(d_i, d_j) \in E$ do

for each $l \in L$ do

$m_{i \rightarrow j}(l) \leftarrow \alpha \sum_{l' \in L} (\psi_i(l') \psi_{ij}(l', l) \prod_{k \in N_i \cap D \setminus \{j\}} m_{k \rightarrow i}(l'))$

end for

end for

until all $m_{i \rightarrow j}(l)$ stop changing

for each $d_i \in D$ do {belief computation}

for each $l \in L$ do

$b_i(l) \leftarrow \alpha \psi_i(l) \prod_{k \in N_i \cap D} m_{k \rightarrow i}(l)$

end for

end for
around this problem, the potential functions are expressed as log-linear combinations of generalisable features. As with non-collective document classifiers, this involves using gradient-based methods to learn a set of weights from a bag-of-words representation of the training data. A set of relational feature weights is also learned. Most commonly, these feature weights are of the form \( w(l, l') \), indicating the likelihood that a document with label \( l \) is linked to a document with label \( l' \) (Taskar et al. 2002; Taskar et al. 2003a; McDowell et al. 2007b; Takamura et al. 2007; Sen et al. 2008; McDowell et al. 2009; Tan et al. 2011).

A key exception to this approach to relational features is the work reported by Stoyanov and Eisner (2012). They use a separate bag-of-words to represent link-specific language. Where the former approach only takes into account the distribution of links between different classes in the training set, this approach learns weights for specific language features and so has a more nuanced relational model. For example, it may be that the majority of links in a particular document domain represent label agreement, except for those cases where a particular word appears in the context of the link. A single feature representation
cannot capture the distinction, but a more detailed feature model can. The implementation is identical with the only difference being the number of weights that are learned.

2.4.6 The Minimum-Cut Technique

The minimum-cut technique is a popular alternative to Markov random fields for collective classification. It has a computationally tractable exact solution, and involves formulating a binary collective classification task as a flow graph and finding solutions using standard methods for solving minimum-cut (maximum-flow) problems.

The method for applying this technique in an in-sample setting is described by Blum and Chawla (2001). The out of sample setting is as follows:

1. For each \( d_i \in D \), let \( w_i : L \rightarrow \mathbb{R}^+ \cup \{0\} \). This gives the content-only preference that document \( d_i \) be assigned the given label.

2. Let \( w' \) be a function \( w' : E \rightarrow \mathbb{R}^+ \cup \{0\} \). This gives the relational preference that the document pair \( (d_i, d_j) \) be assigned the same label.

3. Construct a weighted graph from the set of documents, \( D \), and two additional classification nodes \( d^+ \) and \( d^- \), representing the positive and negative classes respectively.

4. Connect each document \( d_i \in D \) to the \( d^+ \) node with weight \( w_i(l') \), representing the strength of their preference to be given the label \( l' \), based on content-only evidence.

5. Connect each document \( d_i \in D \) to the \( d^- \) node with weight \( w_i(l) \), representing the strength of their preference to be given the label \( l \), based on content-only evidence.

6. Connect each document pair \( (d_i, d_j) \in E \) with weight \( w'(d_i, d_j) \), representing the strength of their preference to receive the same label. Connections with weight 0 may be omitted.

7. Determine the minimum \( d^+, d^- \) cut for the graph, i.e. find the minimum total weight set of edges that can be removed to disconnect \( d^+ \) from \( d^- \). This can be found using a max-flow algorithm, with \( d^+ \) set as the source, \( d^- \) as the sink, and edge weights treated as capacities. Removing the edges results in a two partitions of the graph.
The documents connected to the $d^+$ node are assigned to the positive class and the documents connected to the $d^-$ node are assigned to the negative class.

This approach is equivalent to finding the optimal solution for the cost function for labellings:

$$\text{cost}(Y) = \sum_{d_i \in D} w_i(Y_i) + \sum_{(d_i, d_j) \in E: Y_i \neq Y_j} w^{r}(d_i, d_j)$$

(2.6)

There are three limitations to this technique: (1) it can only be applied to binary problems; (2) there is no way to represent “different label” relationship preferences; and (3) the output of the classifier does not include any measure of classifier confidence.

Figure 2.6 shows our toy collective classification problem formulated as a minimum-cut problem. A crucial difference between this and Figure 2.5 is that there is no link between documents $d_1$ and $d_2$. The clique potentials in the Markov random field indicate that this is a “different label” link, i.e. the preference is for these two documents to be assigned to different classes. There is no way of representing this preference with the minimum-cut approach, so the link is omitted.

Note that there is an approximate solution for multi-class minimum-cut collective classification problems based on linear-programming (Ganchev and Pereira 2007). I do not use this approach in this thesis because my focus is on binary tasks.

As with the Markov random field approaches, there is a need for a method of converting generalisable feature observations into graph weights. The most common way of doing this is to use two separate classifiers, one to produce the content-only preferences, and one to produce the inter-document preferences. I now describe this approach.

### 2.4.7 Dual Classifiers

So far in this section I have described frameworks for doing collective classification using two distinct approaches: (1) iterative classifiers; and (2) global objective functions using variational methods for Markov random fields. I now move on to a third and final family of approaches that is neither iterative nor based on a global formulation. These approaches have been established in the literature for some time, but have not previously
Figure 2.6: A toy binary collective document classification problem framed as a flow graph. The dotted line shows the minimum-cut.
been identified as belonging to a single category. I adopt the term \textit{dual classifier} to refer to classifiers that work by breaking collective classification down into three parts:

1. **Base classification.** Produce base classifications using two distinct classifiers: (1) a content-only classifier that assigns labels from the label set $L$ considering only document content; and (2) a relationship classifier that classifies inter-document relationships as either \texttt{SameLabel} or \texttt{DifferentLabel}. A \texttt{SameLabel} classification indicates that the two documents should receive the same label from the label set, while a \texttt{DifferentLabel} classification indicates that they should receive different labels.

2. **Normalisation.** Normalise the scores output by the two base classifiers so they can be input into a collective classification algorithm.

3. **Decoding.** Produce final classifications by optimally decoding the content-only and relationship level preferences using a collective classification algorithm.

Figure 2.7 shows this process graphically.

The key feature of this approach is the separation of duties between the content-only and relationship classifiers. Like the iterative classifier approach, the dual classifier can make use of standard document classifiers designed for problems that can be represented using simple feature vectors. Like the global formulations, the dual classifier uses collective classification algorithms to decode classification preferences in order to assign labels.

The labelling process with a dual classifier is the same as that shown in Figures 2.5 and 2.6: a collective classification algorithm like loopy belief propagation, mean-field, or minimum-cut is used to decode per-instance and per-relationship preferences into final classifications. The global methods discussed earlier derive these preference values from a single globally optimised feature representation; dual classifiers achieve the same result by use of two separate feature representations, two classifiers, and a normalisation step.

### 2.4.8 Dual Classifier Normalisation

A key advantage of dual classifiers is that they allow the use of standard (non-collective) base classifiers. The SVM classifier is the state-of-the-art approach for a range of tasks and
have been a popular choice for base classifiers. To use SVM, one needs to have a method of normalising the signed decision plane distance produced by the classifier. For loopy belief propagation, mean-field and minimum-cut, this means converting it to pair of non-negative real numbers.

[Platt (1999)] describes a technique for converting the output of an SVM classifier to a calibrated posterior probability. Platt finds that the posterior can be fit using a parametric form of a sigmoid:

$$P(Y_i = 1|s_i) = \frac{1}{1 + \exp(As_i + B)}$$  \hspace{1cm} (2.7)

where $s_i$ is the decision plane distance output by the SVM for document $s_i$ and $A$ and $B$ are the sigmoid parameters.

This is equivalent to assuming that the output of the SVM is proportional to the log
odds of a positive example. Analysis by Platt shows that this technique improves error rate compared to a plain linear SVM and that probabilities are of comparable quality to those produced using a regularised likelihood kernel method.

The sigmoid can be fit using maximum likelihood estimation over the training data set. This approach introduces a bias, most significantly for instances at the margin, which will be forced to have an absolute value of exactly 1. Platt notes that the bias is usually not severe for linear SVMs.

Agarwal and Bhattacharyya (2005) and Angelova and Weikum (2006) use approaches based on this method to convert SVM outputs to probabilities for use in dual classifiers (Wu et al. 2004).

A content-only probability calculated via probabilistic SVM normalisation is ready for direct input into a collective decoding algorithm. For the Markov random field methods let $\psi_i(1) = P(Y_i = 1|s_i)$ and $\psi_i(0) = 1 - \psi_i(1)$. For minimum-cut let $w_i(1) = P(Y_i = 1|s_i)$ and $w_i(0) = 1 - w_i(1)$.

A relationship probability calculated via probabilistic SVM normalisation is ready for direct input into a Markov random field method. In the binary case let $\psi_{ij}(1, 1) = \psi_{ij}(0, 0) = P(Y_i = Y_j|s_{ij})$ where $P(Y_i = Y_j)$ is the probability that documents $d_i$ and $d_j$ belong in the same class and $s_{ij}$ is the score output by the SVM relationship classifier. Let $\psi_{ij}(1, 0) = \psi_{ij}(0, 1) = 1 - \psi_{ij}(1, 1)$ as normal.

Relationship probabilities have to be adjusted for minimum-cut because the model can only use positive weights. One way of doing this is to throw away all relationships for which $P(Y_i = Y_j) = 0.5$. The remainder are converted into weights by the transformation $w'(d_i, d_j) = 2P(Y_i = Y_j)$. This approach converts a probability in the range $[0.5, 1]$ to a weight in the range $[0, 1]$, making it proportional to the weights used to represent content-only preferences.

### 2.4.9 Dual Classifier Applications

Most of the work in collective document classification has focussed on iterative classifier and global approaches. The key advantage of these approaches is their conceptual simplicity and support for multi-class problems. Nevertheless, there are some advantages
Chapter 2: Literature Review

to the dual classifier approach, which I will now explore by means of three examples: (1) Agarwal and Bhattacharyya (2005) do binary sentiment classification of movie reviews; (2) Thomas et al. (2006) do binary sentiment classification of Congressional floor-debate transcripts; and (3) Angelova and Weikum (2006) evaluate their framework on topic classification of academic papers, genre classification of movie websites and topic classification of Wikipedia pages.

Each of the three uses an SVM content-only classifier. This is clearly a major motivator for the choice of a dual classifier architecture. Both Agarwal and Bhattacharyya and Thomas et al. point out that they are motivated by earlier work which shows that SVMs are the best performer in their domain (Pang et al. 2002).

Agarwal and Bhattacharyya and Thomas et al. also share their choice of collective classification algorithm: minimum-cut. Angelova and Weikum uses relaxation labelling, which is another approximate approach for resolving Markov random fields (Pelkowitz 1990).

The most striking commonality between the three approaches is in the complexity of their relationship models. Where the majority of other work uses simple techniques that rely on explicit representations and do not differentiate between links, these three all use complex derivation strategies that produce links with variable strengths.

There are no explicit links in the movies corpus, so Agarwal and Bhattacharyya produce a fully-connected graph with link weights based on a term-overlap similarity measure. Introducing these “relationships” is shown to be sufficient for a significant performance boost, despite the fact that the terms used to determine similarity are the same as those used for base classification.

Angelova and Weikum use links based on co-authorship for the academic paper topic classification task links. For the movie webpage classification task the links are based on actors who appear in both films. For the Wikipedia topic classification tasks hyperlinks are used. A cosine similarity measure is used to augment links in two ways: (1) links are removed if the two documents do not have a threshold degree of similarity, calculated using their document vectors; and (2) links are strengthened or weakened depending on the similarity of the classes to which the two documents are currently assigned. The second approach is based on the intuition that neighbouring documents should receive similar class
labels. The similarity of the different labels is measured by taking a cosine similarity score for each label pair using super-documents composed of the concatenated training documents for each label.

Thomas et al. use a second SVM to classify bag-of-words representations of the context windows around by-name references (see Section 2.3.2 for a more detailed explanation of by-name references).

None of the tasks attempted by Agarwal and Bhattacharyya and by Angelova and Weikum has an obvious interpretation as a global approach because of their reliance on measures of similarity. It might have been possible to engineer some complex features to represent the same information as part of a Markov random field model, but this would seem to complicate rather than simplify things. In the case of Thomas et al. the major benefit of the dual classifier approach is the ability to use an SVM to do relationship classification. Perhaps the same outcome could have been obtained using some form of structured SVM (Sarawagi and Gupta 2008; Joachims et al. 2009). However, only a dual classifier can support a modular approach whereby the relationship classifier could be easily replaced with, say, a state-of-the-art sentence sentiment classifier (e.g. Carrillo de Albornoz et al. 2010).

2.5 Comparison of Collective Classification Techniques

There is a surprisingly limited amount of work focussed on comparing collective document classification techniques. While there are a few corpora that appear across different papers, there are usually differences in experimental configuration that make direct comparison impossible. In this section I give a detailed treatment of the research that is available and explain some of the open questions that the latter parts of this thesis will address. The section is broken into two parts. The first deals with classifier performance, i.e. how accurately each technique can be expected to perform on different types of task. My discussion on this topic will include a detailed summary of the experimental configuration and results obtained for the relevant experiments, so as to ensure all of the necessary context is given to understand performance ramifications. The second deals with classifier complexity, i.e.
the relative quantity of computational resources required when running each technique.

\section*{2.5.1 Classifier Performance}

All of the collective document classifier approaches discussed in this chapter are justified in terms of their ability to improve on non-collective alternatives. Some work has also gone into performing balanced comparisons of algorithms to try to show the relative performance characteristics of each.

There are four works that are the basis for the performance comparisons given in this section. Sen \textit{et al.} (2008) provide a detailed comparison of an iterative classifier, loopy belief propagation, and mean-field. They also consider Gibbs Sampling, a statistical extension to the iterative classifier that I do not discuss here (Gilks \textit{et al.} 1999). McDowell \textit{et al.} (2009) provide a detailed comparison of several types of iterative classifier and loopy belief propagation. Two works provide comparable experiments that allow a dual classifier with minimum-cut (Thomas \textit{et al.} 2006) and loopy belief propagation (Stoyanov and Eisner 2012) to be compared.

Sen \textit{et al.} do their evaluation using two corpora of academic papers, linked by citations. Stemming and stop word removal are performed, and words with document frequency less than 10 are removed. The Cora Corpus contains 2708 documents divided into seven topics representing different branches of machine learning research (McCallum \textit{et al.} 2000). There are 1433 words in the vocabulary and 5429 links. The CiteSeer Corpus has six class labels, 3312 documents, 3703 distinct words in the vocabulary and 4732 links.

Sen \textit{et al.} use two different sampling approaches to create a series of train-test splits. Both result in a non-trivial number of labelled examples being included in the test set (the in-sample setting), one more so than the other. Unfortunately the exact proportion that results for each is not specified.

Naive Bayes and logistic regression versions of the iterative classifier are trialled. Binary bag-of-words features are used and relational features are represented using count aggregation. The feature representation used for the Markov random fields is not specified, though it seems likely that binary bag-of-words is also used. Links are represented using a single feature weight for each label pair.
Tables 2.1 and 2.2 show experimental results for the Cora and CiteSeer corpora respectively. There are four main conclusions from the results:

1. All of the collective approaches provide statistically significant improvements over their non-collective baselines.

2. Logistic regression does better than naive Bayes in an iterative classifier, with a statistically significant improvement on the Cora Corpus.

3. Loopy belief propagation and mean-field appear to slightly outperform the iterative classifier, though the difference is not statistically significant.

4. Incorporating more known document labels leads to better performance in general, with no obvious advantage for one classifier.

Sen et al. also perform experiments with synthetic data to try to show the effect on performance of varying two parameters: (1) homophily, which they define as the average percentage of a document’s neighbours that have the same label; and (2) link density, defined generally as the degree of connectedness in the document graph. The simulation algorithm is described in detail, though some of the experimental parameters are omitted, presumably because of a desire to focus on the general trends revealed by the simulations. These are as follows:

**Collective classifier performance increases as homophily increases**

At very high homophily all of the classifiers give near perfect performance. The one
Table 2.2: Collective classification performance on the CiteSeer Corpus as reported in Sen et al. (2008). Results are shown for sampling strategies that lead to fewer known labels and more known labels. The actual proportion of known labels is not reported.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Fewer known labels</th>
<th>More known labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Naive Bayes</td>
<td>74.27</td>
<td>74.87</td>
</tr>
<tr>
<td>Baseline</td>
<td>Logistic regression</td>
<td>73.34</td>
<td>73.21</td>
</tr>
<tr>
<td>Iterative</td>
<td>Naive Bayes, count features</td>
<td>75.40</td>
<td>76.83</td>
</tr>
<tr>
<td>Iterative</td>
<td>Logistic Regression, count features.</td>
<td>76.29</td>
<td>77.32</td>
</tr>
<tr>
<td>Global</td>
<td>Mean-field</td>
<td>76.57</td>
<td>77.32</td>
</tr>
<tr>
<td>Global</td>
<td>Loopy belief propagation</td>
<td><strong>76.63</strong></td>
<td><strong>77.59</strong></td>
</tr>
</tbody>
</table>

exception is the naive Bayes iterative classifier, which flattens out at a homophily value of 0.5, or after showing about half of the total improvement given by the others. No explanation is offered for this, though the generally inferior performance of this algorithm in the real-world tests suggests it would not be a desirable choice in any case.

**Collective classifier performance increases as link density increases**

As should be expected, the classifiers are helped by increasing the amount of relationship information by adding more links. However, there is some contrary behaviour at very high link density. At the highest level measured, presumably close to a fully connected graph, loopy belief propagation experiences a sudden drop in performance, to the point where it scores significantly lower than the content-only classifiers. There is a similar but much less pronounced drop for the naive Bayes iterative classifier, while mean-field and the logistic regression iterative classifier are unaffected. As Sen et al. note, this problem has been observed by others (Neville and Jensen 2008). It occurs in a scenario where the algorithm suddenly converges on an extreme labelling where all instances are given the same label.

McDowell et al. do their real-world performance comparison using the Cora and CiteSeer corpora as well as the WebKB Corpus (Craven et al. 1998). They also use two other corpora, which I omit on the grounds that they are not document classification tasks.

The WebKB Corpus is a collection of 1541 university home pages to be classified into
McDowell et al. use a binary representation of the 100 most frequent features in each corpus as document features. For the iterative classifiers, cross validation is used to select the best performing relational feature model (count, proportion or multiset) at training time. For loopy belief propagation, links are represented using a single feature weight for each label pair.

The purpose of the McDowell et al. analysis is to show the benefits of “caution” in collective classification, a capacity that some algorithms have to manage estimation uncertainty when performing inference using relational features. Loopy belief propagation is shown to exhibit this behaviour, as does a variation to the standard iterative classifier referred to here as cautious iterative classification. The experiments on synthetic and real data demonstrate that the relative gain from cautious behaviour increases with increasing autocorrelation (homophily), decreasing content-only predictiveness and decreasing labelled proportion.

Each of the three corpora are tested in a configuration where 10% of the labels in the test set are known. For Cora, the iterative classifier and cautious iterative classifier both achieve accuracies of approximately 80%. Loopy belief propagation does worse with a score of approximately 73%.

For CiteSeer the trend is similar with approximately 76% for the iterative classifiers and 74% for loopy belief propagation.

For WebKB loopy belief propagation does the best with a score of approximately 65%. The iterative classifier is next with 60% followed by the cautious iterative classifier with 58%.

The cautious iterative classifier is shown to perform particularly well when content-only predictiveness is decreased by throwing away document attributes, with the most noticeable gain occurring when content-only accuracy is between 40% and 60%.

The experiments with synthetic data support the Sen et al. conclusions that: (1) collective classifier performance increases as homophily increases; (2) collective classifier performance increases as link density increases; and (3) the one crucial difference between the classifiers is that loopy belief propagation performance drops off dramatically at very
The final comparison of collective document classification algorithms that is significant to this thesis drawn from Thomas et al. and Stoyanov and Eisner. Their works use the ConVote Corpus of U.S. Congressional floor-debate transcripts. Thomas et al. provide the original benchmark for the task using a dual classifier with SVM base classifiers, a tailored normalisation strategy, and the minimum-cut algorithm. Stoyanov and Eisner demonstrate the relative benefits of minimum-risk training for loopy belief propagation compared to the standard approach of using maximum likelihood estimation. Minimum-risk training attempts to learn parameters to minimise the task-specific loss of whatever inference and decoding methods will be used at test time.

Both works perform collective classification on the ConVote Corpus using binary bag-of-words features in the out-of-sample setting. A detailed description of the ConVote Corpus is provided in Chapter 3 so I omit the details of numbers of instances, features and links from this section. I also omit the details of the normalisation scheme used by Thomas et al., as it is discussed in detail in Chapter 4.

Note that a unique type of link representation is used for the ConVote task. Where links between documents normally have a binary linked or not linked representation, links in ConVote are represented by a bag-of-words. The words are taken from the context of the document immediately surrounding a by-name reference to another speaker. Theoretically, this allows the classifier to learn that some features are indicative of a SAME_LABEL relationship and others a DIFFERENT_LABEL relationship. For example, the word agree might indicate SAME_LABEL because it tends to be used in statements like I agree with [name of other speaker]. Section 3.2.2 has a detailed explanation of this point.

Stoyanov and Eisner present results for several different variations of their minimum-risk approach as well as multiple variations of loopy belief propagation, all of which do better than the Thomas et al. approach. The results in Table 2.5.1 show that loopy belief propagation performs significantly better than minimum-cut, while the minimum-risk variant performs significantly better again. The lower performance for the minimum-cut approach may be explained by the fact that DIFFERENT_LABEL classifications output by the relationship classifier have to be thrown away because they are not supported by the minimum-cut algorithm. Another possibility is that the normalisation technique used for the dual classifier
Table 2.3: Collective classification performance on the ConVote Corpus as reported in \cite{Thomas2006} and \cite{Stoyanov2012}. Note that the two papers provide different numbers of significant figures. The figures quoted for loopy belief propagation are for the sum-product versions of the Stoyanov and Eisner classifiers.

is sub-optimal. Chapter 3 discusses these questions in detail.

I close this analysis with the following four general conclusions:

1. There is a limited amount of information from which to draw conclusions about the relative performance of collective classification algorithms.

2. There is no clear performance difference between the iterative classifier and Markov random field approaches, except for the fact that loopy belief propagation performs very badly when link density is very high.

3. The merits of the minimum-cut algorithm are not clear. It will be at its best in domains where there are no differentLabel links.

4. In general, adding more links or making links more predictive increases overall performance.

### 2.5.2 Classifier Complexity

For an iterative classifier, the cost of an iteration comes primarily from updating the values for relational features and running the SVM or other classifier to compute the label. For a linear classifier like SVM or logistic regression the cost of running the local classifier is $O(N)$, where $N$ is the number of documents being classified. Assuming each document is connected to a small number of neighbours, the cost of updating relational feature values is also $O(N)$ per iteration. In the extreme case of a fully connected document graph the number of updates required is $O(N(N-1))$. 

The number of iterations required for an iterative classifier is obviously variable. Prior work has found that in cases where the classifier does not terminate earlier, cutting off after 10 iterations seems to have no significant impact on accuracy (Sen et al. 2008, McDowell et al. 2009).

Loopy belief propagation and mean-field do not need to recompute relational feature values, but their main loops iterate over the set of neighbours for each document. Assuming this set is small, the cost is $O(N)$. Again, the extreme case of a fully connected document graph gives $O(N(N - 1))$ iterations.

In practice, the cost of classifying with loopy belief propagation and iterative classification is similar (McDowell et al. 2009).

The minimum-cut algorithm has a polynomial time solution which, like the other algorithms, tends to be linear in practice (Blum and Chawla 2001).

The different learning algorithms have considerably more variable complexity. For iterative classifier and dual classifier approaches there is no iteration required in training, so the cost is equivalent to training for a non-collective problem of the same size. In the case of SVM this can be linear cost for large training sets (see Section 2.2.3).

When performing gradient-based optimisation for global methods it is necessary to execute the classification algorithm many times. McDowell et al. find that this makes loopy belief propagation at least an order of magnitude slower.

2.6 Summary

In this chapter, I have presented the existing work that is the basis for the remainder of this thesis. The reader should now be prepared with a detailed understanding of the tasks and methods that I will be implementing and extending in the chapters that follow.

I began by describing document classification, which is the basic machine learning task that this work is based on. I then described a formal framework for collective document classification and introduced the concepts of explicit and implicit inter-document links. I fleshed out these concepts by reviewing prior applications in both of these categories.

I then introduced three distinct families of collective document classifier: (1) itera-
tive classifiers, which work by incorporating neighbour information into document-level features using aggregation functions; (2) global methods based on loopy belief propagation and mean-field, which treat the entire document graph as a single entity to be classified; and (3) dual classifier approaches, which combine a content-only classifier and a relationship classifier using a graphical model. I contrasted these approaches in terms of the previous applications, relative performance, and relative complexity.

In the next chapter I will introduce the test corpora that will be used for the experiments in this thesis.
Chapter 3

Corpora

3.1 Introduction

In this chapter I introduce the corpora that will be the basis for the experimental work in this thesis. I begin by defining a set of corpus selection criteria. I then introduce the ConVote and Bitterlemons corpora, discuss their origins and structure, and show that they meet the selection criteria. A description of the real-world document classification task that each corpus is intended to simulate is given, along with a brief analysis of the fitness of each corpus for that purpose.

3.1.1 Corpus Selection Criteria

Five main characteristics determine the suitability of a corpus for use in the collective document classification experiments in this thesis:

1. Availability. This appears to be a trivial requirement, but it deserves to be noted first. When research time and resources are limited, it is wise to select a readily available corpus rather than one that needs to be manually compiled or otherwise sought after.

2. Existing benchmarks for collective classification. Existing benchmarks make it possible to compare results. This serves the dual purpose of allowing comparison with other methods and providing a point of external reference for increased confidence about the reliability of experimental results.
3. **Unconstrained vocabulary.** Collective classification works when it can use features that are either absent or ambiguous in training data. The former category only exists in cases where the corpus domain has a large, unconstrained vocabulary. Domains like earnings reports and commodity prices (found, for example, in the popular Reuters-21578 Text Categorisation Corpus) have restricted vocabularies and are less likely to be suitable for collective classification.

4. **Dominance of task-relevant content.** Features that are absent or ambiguous in training data become useful when they allow the collective classifier to link pairs of test-set documents that are likely to have the same label. This is possible when the test set contains unseen features that are correlated with specific labels. For instance, feature $a$ may be strongly correlated with label $A$. If feature $a$ has not been seen in training it will not be useful for content-only classification. However, if instances $X$ and $Y$ both contain feature $a$, a collective classifier can set a “same label” preference between them and so increase the chances of both being classified correctly.

A significant part of the challenge of collective classification is detecting test-set relationships that actually relate to the dimension of classification. There are many scenarios in which a feature may be shared across documents in the test set for reasons entirely unrelated to the labels of those documents. For example, when performing sentiment classification of movie reviews it would be easy to find overlapping test set features relating to movies that do not have reviews in the training set. These might be references to the same actors, locations, or characters. This kind of similarity is not likely to help in determining whether the reviews are positive or negative.

To reduce the difficulty involved in differentiating relational features that are helpful for collective classification from those that are not, I select corpora that appear to be dominated by content that is relevant to the dimension of classification.

5. **Presence of explicit inter-document relationships.** My purpose in this thesis is to investigate the use of both explicit and implicit inter-document relationships for collective classification. To explore explicit relationships, I need at least one corpus with simple, easily identifiable features that serve to relate two or more instances.
Three examples are: (1) formal citations in academic literature; (2) hyperlinks between web pages; and (3) references to the name of another participant in a group conversation. Whatever type of relationship is used, it is essential that it can be detected using a programmatic method and resolved unambiguously into a pointer to another document or documents in the corpus.

### 3.2 The ConVote Corpus

In this section I describe the ConVote corpus, the first of two corpora used for the experiments in this thesis.

#### 3.2.1 Corpus Origins

ConVote is a collection to transcripts of floor-debates from the U.S. Congress \[\text{Thomas et al.}\ 2006\]. Floor debates are formal debates that take place in the House or Senate to consider a bill or resolution. Members take turns giving speeches describing their position and attempting to persuade other members to adopt their view. Occasionally members will also give follow-up speeches, ask or answer questions, or give rebuttals. Proceedings end with a vote in which each member publicly lodges either a “for” or “against” for the bill or resolution.

\[\text{Thomas et al.}\] created the ConVote Corpus using GovTrack\[\text{1}\] an independent website that collects publicly available data on the legislative and fund-raising activities of U.S. congresspeople. GovTrack’s stated objective is to “help the public research and track the activities in the U.S. Congress, promoting and innovating government transparency and civic education through novel uses of technology.”\[\text{2}\]

The HTML versions of GovTrack’s transcripts of floor-debates for 2005 were downloaded, cleaned, tokenised, and broken into debates and constituent speeches for use in the corpus. They were then cross-matched with GovTrack’s voting records, allowing each speech to be labelled with an identifier for its speaker and their recorded vote for the corre-

---

1 http://govtrack.us
2 http://www.govtrack.us/about.xpd
Chapter 3: Corpora

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Train</th>
<th>Test</th>
<th>Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>1.2M</td>
<td>800K</td>
<td>300K</td>
<td>60K</td>
</tr>
<tr>
<td>Speeches</td>
<td>3857</td>
<td>2740</td>
<td>860</td>
<td>257</td>
</tr>
<tr>
<td>Debates</td>
<td>53</td>
<td>38</td>
<td>10</td>
<td>5</td>
</tr>
<tr>
<td>Average number of speeches per debate</td>
<td>73</td>
<td>72</td>
<td>86</td>
<td>51</td>
</tr>
<tr>
<td>Average number of speakers per debate</td>
<td>32</td>
<td>31</td>
<td>41</td>
<td>23</td>
</tr>
<tr>
<td>Average number of tokens per speech</td>
<td>324</td>
<td>316</td>
<td>369</td>
<td>252</td>
</tr>
<tr>
<td>By-name references</td>
<td>2292</td>
<td>1528</td>
<td>618</td>
<td>146</td>
</tr>
<tr>
<td>Average number of by-name references per speech</td>
<td>0.6</td>
<td>0.6</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>Percentage of speeches with vote “for”</td>
<td>54</td>
<td>52</td>
<td>58</td>
<td>54.</td>
</tr>
<tr>
<td>Percentage of speeches with no by-name references</td>
<td>70</td>
<td>71</td>
<td>66</td>
<td>68</td>
</tr>
</tbody>
</table>

Table 3.1: ConVote statistics.

sponding debate. Debates in which the losing side obtained less than 20% of the vote were omitted from the corpus on the grounds that they have less interesting discourse structure because they are not “controversial”.

Short, formulaic speeches that are clearly off topic (e.g. Madame Speaker, I am pleased to yield 5 minutes to the gentleman from Massachusetts?) were removed automatically. Speeches containing the word amendment were also removed because they tended to contain opinions on amendments rather than the bill being discussed.

The corpus was tokenised by splitting the text at whitespace boundaries. Punctuation characters were treated as separate tokens with a string of consecutive punctuation characters regarded as one token. There were two exceptions to this rule: (1) the apostrophe s was regarded a single token; and (2) periods used to signify abbreviations were taken as part of the word they abbreviated.

Next, case-normalisation was applied and debates were randomly assigned to training, development and test sets for use in collective classification experiments.

The final component of the corpus is a set of explicit inter-document relationships, which are described in the next section.
Reference to something said by a previous speaker:

... my good friend mr. dreier suggested that there will not be a divisive debate this morning. ...

... mr. speaker, perhaps the gentleman from new york (mr. rangel) is not aware that the senate has included the alternative minimum tax in their reconciliation tax package. ...

Reference to the author of the bill or resolution:

... and that is why the gentleman from new york’s (mr. rangel) is the only honest way if you want to protect the middle class. ...

... if we know, with the bill of the gentleman from wisconsin (mr. sensenbrenner), who is there legally, it is much easier to tell who is there illegally. ...

Expression of gratitude:

... i appreciate mr. miller who has worked very diligently with me and understands my concerns. ...

... mr. speaker, i want to thank the gentleman from new york (mr. rangel) for offering this motion to recommit today. ...

Figure 3.1: Tokenised sample text from the ConVote Corpus showing examples of common ways speakers refer to each other by name or title. The by-name references that are annotated in the corpus are shown underlined. Note that the names given in parentheses are a part of the transcriptions provided by GovTrack.

3.2.2 Corpus Structure

Speakers in floor-debates frequently refer to each other by name or title in their speeches. Figure 3.1 gives examples of this in three common cases: (1) when a speaker refers to something said by another speaker; (2) when a speaker refers to the author of the bill or resolution being debated; and (3) when a speaker expresses gratitude to another speaker.

Thomas et al. augmented their corpus with a set of annotations to link these by-name references to the identity of the speaker being referred to.

The statistics in Table 3.1 give a picture of the final size and structure of the ConVote Corpus with the by-name reference annotations incorporated. The corpus is roughly balanced, with 54% of all speeches having a “for” vote. On average there are 0.6 by-name references for each speech. These are distributed across a minority of the corpus, with 70% of the speeches being isolated, i.e. not connected to any other speeches via by-name
3.2.3 Modifications to the ConVote Corpus

For the purposes of this work, two modifications are made to the ConVote Corpus as presented by Thomas et al.:  

1. Concatenate speeches by the same speaker. For the majority of their experiments, Thomas et al. elect to process speeches individually, regardless of the fact that multiple contributions by a speaker to a single debate must, by definition, all have the same vote. They prefer an experiment with more instances and inter-document references to better emphasise the usefulness of their method of collective classification. I concatenate these multiple contributions into a single, unified, speech for each speaker within a debate. This gives a more natural view of the task for two reasons:

   (a) Use of the by-name references in the ConVote Corpus for collective classification requires knowledge of speaker-labels for all speeches. Concatenating the contributions from each speaker simplifies computation without introducing any new information.

   (b) Speakers often contribute a number of smaller speeches in addition to the main speech in which they outline their position. These may form part of a question and answer (e.g. Yes, I am, Mr. Speaker, in its current form.) or may be clarifications or additional arguments. These kind of short, highly contextualised utterances present a very different classification challenge to that posed by longer, structured debate contributions. Subsuming them into other speeches allows me to focus on techniques that are applicable to longer documents.

There are two apparent drawbacks to this approach:

(a) Having a single label for a speaker rules out the scenario where a speaker changes opinion in the course of the debate. This is in fact very rare, as the debates are highly polarised and dominated by the members who have already
established their positions. Thomas et al. adopt the same simplification in their experiments focusing on speech-level classification.

(b) The distinction between speakers and speeches is lost. This information is potentially useful in scenarios involving more nuanced classification, which is beyond the scope of this thesis.

2. Drop train, test, development set assignments. Common machine learning practice dictates that, where possible, corpus-based evaluations are performed using k-fold cross-validation rather than a holdout estimate (Witten and Frank 2005). This approach makes more efficient use of the available data by allowing for larger training sets and ensuring that each instance is assigned to the test set an equal number of times (Blum et al. 1999). I drop the holdout configuration used by Thomas et al. and instead use k-fold cross-validation. k is set to 53, the number of debates in the corpus, to allow each debate to be tested once using all of the other debates as training data.

These changes mean that my numeric performance figures will not be directly comparable with those reported by Thomas et al. This is unfortunate but justifiable given the benefits of cross-validation. I will give performance figures for my own implementations of the Thomas et al. techniques so that comparison is still possible.
3.2.4 Network Structure of the ConVote Corpus

Figure 3.2 shows the structure of a small section of the modified ConVote Corpus. It includes sections of example text that should help the reader gain a better grasp of the corpus contents.

Table 3.2 gives statistics for the modified ConVote Corpus. The identical figures for the average number of speeches and speakers per debate reflect the fact that each speaker now contributes only one unified speech. The ratio of by-name references to speeches has increased from 0.6 to 0.9, with only 31% of speeches left isolated. The average number of tokens per speech has increased from 324 to 735.

The corpus retains its roughly balanced class distribution, with 49% of speeches having the “for” label.

Figure 3.3 shows the distribution of by-name references among the 1171 instances that are connected, i.e. the source or target of one or more references. Half of the connected instances have two references or more and a significant proportion have five or more. Beyond this, there is a tail of speeches that have much larger numbers of references, with a handful having more than 30.

3.2.5 Fitness for Purpose

The ConVote Corpus is intended for use in experiments that simulate the process of judging the likely vote of a member of congress based on their debate contribution. This is something of a niche task and probably more interesting from an experimental perspective than a practical one. However, as Thomas et al. note, analysis of political text is an area of particular interest given the increasing role of the web in the political process.

From a statistical point of view, ConVote limits sample bias by using content from a defined period of time (the 2005 calendar year). The decision to omit “non-controversial” debates means that classifiers evaluated with ConVote will be untested on this type of debate, so statements about their probable utility in real-world use have to be appropriately qualified.

ConVote is an excellent fit for the corpus selection criteria defined at the beginning of this chapter. In particular:
Blackburn, Marsha (R) [for]

at this time, I would like to recognize the gentleman from Texas (Mr. Hensarling) who has worked tirelessly not only on budgeting and not only on looking at how we budget, but looking at what happens with tax policy and the ramifications that that has throughout our economy, both for our large businesses, our small businesses and for our families. We appreciate the leadership that he has brought to the budgeting issue, looking at both sides of that ledger, your inflows and your outflows. ... 

Hensarling, Jeb (R) [for]

Mr. Speaker, unless we enact H.R. 4297 and defeat the Democratic substitute, Americans will receive a most unwelcome Christmas gift from the Democrats, a huge automatic tax increase. This will cost families billions of dollars and jeopardize millions of their jobs. Mr. Speaker, let me tell you just about a few of those jobs that could be lost in my East Texas district if the Democrats have their way in raising taxes. ... 

McDermott, James ‘Jim’ (D) [against]

...the repeal that you tried to put through here under Clinton was an attempt to let the top off taxes at all. You simply wanted to give them an internal tax holiday if they could figure out how to manipulate the tax structure. The average janitor does not have a way to manipulate the system, and that is why the gentleman from New York’s (Mr. Rangel) is the only honest way if you want to protect the middle class. I urge your vote for the alternative minimum tax proposal. 

Rengel, Charles B. (D) [against]

Mr. Speaker, the gentleman picked a heck of a time to lose his voice here now. Mr. Speaker, I would like to ask the gentleman just one question on my time. These very important tax cuts or extension of tax cuts you are talking about, could you share with me as simply as possible as to when they expire, what year? Mr. Speaker, I had an old law professor, and he once told me, if you don’t have the facts going for you, raise your voice. I never understood it, but I do now. 

Thomas, William ‘Bill’ M. (R) [for]

Mr. Speaker, I yield myself 30 seconds. Mr. Speaker, perhaps the gentleman from New York (Mr. Rangel) is not aware that the Senate has included the alternative minimum tax in their reconciliation tax package. They have already voted on it. So there is no need to provide any assurance from the House side, because the Senate has already included it. But, again, that is reality. No, it is a rhetorical question. Whose time is it, Mr. Speaker?

Figure 3.2: Tokenised sample text from a ConVote debate taken from a session of Congress “Providing for consideration of HR 4297 Tax Relief Extension Reconciliation Act of 2005”. The underlined sections have annotations giving the identifier of the speaker being referred to. Separate speech segments are indicated by paragraph breaks; in the actual modified corpus segments are separated by newlines. Ellipses indicate where speech segments that have been shortened for the purposes of this illustration. Votes “for” and “against” are labelled along with party labels “D” and “R”. The latter are not used in any experiments in this thesis.
Figure 3.3: Cumulative distribution of by-name references for ConVote speeches.

1. **Availability.** Thomas et al. have made ConVote freely available for download.\(^3\)

2. **Existing benchmarks for collective classification.** Thomas et al. created ConVote for the purpose of performing collective classification experiments. The paper includes a detailed description of their method and results.

3. **Unconstrained vocabulary.** The rhetorical language used by members of congress is drawn from a rich vocabulary. It ranges from highly informal banter, through semi-formal argumentation and quotations from legislation. The range of topics dealt with in the corpus is very broad and includes stem cell research, terrorism and euthanasia.

4. **Dominance of task-relevant content.** Unlike movie reviewers, congressional debaters rarely attempt to give objective descriptions of the bills and resolutions they are discussing. They assume that their colleagues are already familiar with the details, and focus only on those aspects of the subject that are material to their argu-

\(^3\)http://www.cs.cornell.edu/home/llee/data/convote.html
ment; objectivity is set aside in favour of rhetorical attempts to persuade. I posit that in this case the language that is specific to a particular debate can be used to find relationships useful inter-document relationships. I offer two points in favour of this view, gleaned from my personal study of congressional transcripts and other forms of political debate:

(a) Congressional debaters often seem to borrow content from others on their side, perhaps because they use common sets of speaking notes.

(b) Political speech is repetitive. Different contributors on the same side of a debate seem to have a set of specialised words and phrases that they use to forward their argument.

5. **Presence of explicit inter-document relationships within topics.** As already discussed, ConVote has explicit inter-document relationships within topics in the form of by-name references to other speakers in a debate. [Thomas et al.](#) have already shown that these are useful for collective classification. We will see more of this in Chapter 4.

There is one further point in favour of ConVote. I posit that the chance of detecting useful inter-document relationships is higher in a corpus that is broken into multiple topics, subjects, or sub-domains. I use these terms in the general sense to refer to documents that are somehow more closely related to each other than they are to other documents in the corpus. When a topic is “unseen”, i.e. not present in the training data, there are two factors that combine to suggest the possibility of finding useful inter-document relationships:

1. By definition, documents within a topic are inter-related, so there is increased potential for detecting relationships between documents within a topic.

2. Topics that have not been seen in training will contain topic-specific language that the classifier will not be able to utilise effectively. This presents an opportunity because: (1) performance of content-only classification will be lower, so there is scope for improvement; and (2) inter-document relationships derived from features related to the topic will be available to bridge the gap.
3.3 The Bitterlemons Corpus

3.3.1 Corpus Origins

Bitterlemons is a collection of articles on the Israeli-Arab conflict harvested from the Bitterlemons website (Lin et al. 2006). The website describes itself as “reflecting a joint Palestinian-Israeli effort to promote a civilized exchange of views about the Israel-Arab conflict and additional Middle East issues among a broad spectrum of participants.” In each weekly issue, the editors, Ghassan Khathhib (a Palestinian) and Yossi Alpher (an Israeli), contribute an article giving their perspectives on some aspect of the conflict. In addition, the editors invite two guest authors to contribute articles to the issue, one giving an Israeli perspective and the other a Palestinian perspective. Sometimes these guest contributions take the form of an interview.

The HTML versions of the Bitterlemons articles published between 2001 and 2005 were downloaded, cleaned, and tokenised for use in the corpus. The corpus creators removed all references to edition numbers, publication data, topic labels, author names and bibliographic information so that the corpus could be used for experiments that focused strictly on interpretation of the content of the articles.

The corpus is tokenised by splitting the text at whitespace boundaries. Punctuation characters are treated as separate tokens with a string of consecutive punctuation characters regarded as one token. Case-normalisation is not applied.

3.3.2 Corpus Structure

The statistics in Table [3.3] give a picture of the size and structure of Bitterlemons. The corpus is somewhat smaller than ConVote, with 501,000 tokens in 594 articles. Bitterlemons topics are much smaller than ConVote debates, with only four articles in each topic. The corpus has an even split between Israeli and Palestinian articles. Unlike the ConVote Corpus, there are no explicit inter-document references. Figure [3.4] shows example sections of the four articles that comprise the topic “Sharon’s Herzliya speech”.

4 http://www.bitterlemons.org/
5 http://www.bitterlemons.net/about.php
There are a number of ways to look at Israeli Prime Minister Ariel Sharon’s Herzliya speech of December 18, 2003. They appear to offer us a little good news, but a lot more of the bad news we have become used to in the past three years. The following "takes" are not mutually exclusive.

First, the good news. Sharon, along with many on the political right, appears finally to have understood the demographic problem and the dangers of direct ongoing occupation. He wants to disengage. The architect of the settlement movement in the West Bank and Gaza over the past 25 years is indicating that some settlements will have to be "redeployed." While he clearly still does not comprehend the minimal territorial needs of a viable Palestinian state, and he studiously avoids relying publicly on demographic arguments...

Zeev Hever has served in recent years as head of the "Amana" movement, the settlement arm of the Yesha Council (Council of the Communities in Judea, Samaria and Gaza). Hever, nicknamed Zambish, is a close personal friend of the prime minister of Israel. What do these two have in common, with their quarter-century age differential? Simply this: Ariel Sharon, despite his frequent involvement in financial improprieties, in his heart of hearts likes and respects idealists who keep their distance from financial issues and devote themselves entirely to the Jewish people and to Zionist ideals. In his eyes Zambish symbolizes these ideals.

On December 16, two days before Sharon’s Herzliya speech, Hever addressed the press for the first time in years. It happened at Migron, the outpost Sharon intends to dismantle. Hever, who is a kind of alter ego to Sharon when it comes to Zionism...

Speaking broadly, Palestinians saw the speech of Israeli Prime Minister Ariel Sharon given at the Herzliya conference as one of the most dangerous Israeli plans to imperil the peace process and the Israeli and Palestinian peoples. This plan proposes to use the wall that Israel has built along the outlines of a 35-year-old settlement expansion project in order to determine by force the final arrangements for the Palestinian territories. In his speech, Sharon talked explicitly about consolidating illegal settlements, rather than removing them, with no regard for Israel’s obligation to stop building settlements according to the roadmap.

While Sharon gave a verbal nod to the roadmap, he also reaffirmed his speech in Aqaba earlier this year, which includes the multiple Israeli reservations to the...

A: Sharon was giving the Palestinians an ultimatum: "either you accept my roadmap with the 14 alterations injected by my government (and this is the maximum that you will get out of me) or we are going to unilaterally give you less than that. The choice is yours -- if you opt for negotiations, I can give you more, but you should know (and this is a tacit understanding) that the most you will get even then is the most that is offered in 'my roadmap', which includes only the land inside the wall."

A: First, I don’t think that Sharon is changing his ideology, but practically, there are many things that he wants to take into account. He wants to please the American administration, but he also wants to use this time during which the American administration is entering an election campaign to push for his own interests. He also wants to send...

Figure 3.4: Tokenised Bitterlemons Corpus sample text from an edition on the topic “Sharon’s Herzliya speech”. Text has been truncated to fit. Perspectives Israeli and Palestinian are labelled along with whether the contributor is an editor or guest. Unlike the ConVote Corpus, there are no explicit inter-document relationships. Paragraph breaks are included for legibility.
### Table 3.3: Corpus statistics for the Bitterlemons Corpus. Fields relating to by-name references are shown for ease of comparison with ConVote.

<table>
<thead>
<tr>
<th>Total</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokens</td>
<td>501K</td>
</tr>
<tr>
<td>Articles</td>
<td>594</td>
</tr>
<tr>
<td>Topics</td>
<td>149</td>
</tr>
<tr>
<td>Average number of articles per topic</td>
<td>4</td>
</tr>
<tr>
<td>Average number of tokens per article</td>
<td>843</td>
</tr>
<tr>
<td>By-name references</td>
<td>0</td>
</tr>
<tr>
<td>Average number of by-name references per article</td>
<td>0</td>
</tr>
<tr>
<td>Percentage of speeches with label Israeli</td>
<td>50</td>
</tr>
<tr>
<td>Percentage of speeches with no by-name references</td>
<td>100</td>
</tr>
</tbody>
</table>

#### 3.3.3 Modification to the Bitterlemons Corpus

A significant number of guest articles appear in the form of interviews, with the editors posing a series of questions that are answered individually by the guest. The question text is problematic, because, unlike the rest of the text, it does not originate with the nominated author (the guest) and is therefore not indicative of the author’s opinion. For the purposes of this work, I have chosen to omit all of the question blocks from articles that have an interview form. This removes a problem where potentially misleading input is provided to the document classifier. In my judgement the benefit of improving the quality of data input in this way outweighs the cost of divergence from the approach taken by the corpus compilers.

#### 3.3.4 Fitness for Purpose

The Bitterlemons Corpus is intended for use in experiments that simulate the process of detecting the perspective of an author contributing opinion to a polarised debate. As with ConVote, this task is particularly relevant given the increasing role of the web in the political process.

Like ConVote, Bitterlemons limits sample bias by using all of the articles from the Bitterlemons website from a defined period of time.

Bitterlemons is a relatively small corpus and very narrow in scope; half of the articles
are written by only two authors. I chose Bitterlemons not because it is representative of a realistic task, but because it has a convenient combination of the properties I need to explore collective classification using implicit inter-document references. In particular:

1. **Availability.** Lin *et al.* have made the Bitterlemons Corpus freely available for download.⁶

2. **Existing benchmarks for collective classification.** The Bitterlemons Corpus has no existing benchmarks for collective classification. This is to be expected, because the Bitterlemons Corpus does not have an obvious network structure. I will be attempting to show that the corpus has implicit inter-document relationships which can be detected and used in collective classification.

3. **Unconstrained vocabulary.** As with ConVote, the rhetorical language in Bitterlemons is drawn from a rich vocabulary. Each edition deals with some new aspect of the conflict, with contributions in both essay and interview form.

4. **Relationship between topic-specific content and classification task.** For the same reasons as with ConVote, I anticipate that the rhetorical language in the Bitterlemons will be suitable for detecting useful inter-document relationships. In any kind of rhetorical discussion, participants on the same side seem to have a set of specialised words and phrases that gives a special shared character to their arguments.

5. **Presence of explicit inter-document relationships within topics.** There are no explicit inter-document relationships within the Bitterlemons topics. See point.⁵

Unlike ConVote, the topical structure of Bitterlemons is not likely to be helpful for collective classification. There are two reasons for this. First, there are only four articles in each edition. This limits the number of potential relationships. Second, the standard division of the corpus into training and test sets requires that each edition be separated, with the articles contributed by the editors in one set and those contributed by the guests in the other. This division of the corpus was chosen by the compilers as a way of increasing

---

⁶https://sites.google.com/site/weihaiolnatcmu/data
the difficulty of the task. Moreover, using the editor contributions in the training set allows
the task to be framed as “perspective” classification, rather than author attribution, i.e. we
are focused on the content of the contributions rather than stylistic or biographical features
that may identify one editor or the other.

3.4 Summary

The ConVote and Bitterlemons corpora are both collections of political text collected
from the Web. ConVote is a set of transcripts of floor-debates from the United States
Congress. Bitterlemons is a collection of topical essays on the Arab-Israeli conflict. Con-
Vote speeches contain explicit inter-document relationships in the form of by-name refer-
ences to other speakers which have already been shown to be useful for collective classi-
fication. Bitterlemons contains no such references, but has a rhetorical style and topical
structure that may allow derivation of useful relationships based on textual similarity.

It will be the task of the next chapter to find the best way to do collective classification
on ConVote using by-name references.
Chapter 4

Collective Document Classification with Explicit Inter-document Relationships

4.1 Introduction

In this chapter experiments are conducted to show the utility of a range of state-of-the-art techniques for collective classification using the explicit inter-document relationships in the ConVote Corpus. The chapter begins by describing and critiquing the Thomas et al. (2006) approach to this task, which uses a dual classifier model and the minimum-cut algorithm. I then introduce two alternative approaches: (1) a dual classifier using either the mean-field or loopy belief propagation algorithm; and (2) an iterative classifier. I demonstrate a range of ways in which these alternative approaches are superior for the task. I then provide a comparison with the state-of-the-art approach for collective classification on the ConVote corpus (Stoyanov and Eisner 2012) and show that my dual classifier approaches give superior performance. Finally, a theoretical analysis of the strengths and weaknesses of the various approaches is provided, together with suggested criteria for selecting an algorithm for other collective classification tasks.
4.2 Problem Formulation

Recall from Section 2.3.1 that collective document classification assumes we are given a graph \( G = (D, E, X, Y, L) \) where \( D \) is a set of document vertices, \( E \) is a set of edges, each \( \vec{x}_i \in X \) is an attribute vector for document \( d_i \in D \), each \( Y_i \in Y \) is a label variable for \( d_i \), and \( L \) is a set of labels. The task is to infer the values for \( Y \).

For convenience of notation, assume that \( N \) is a neighbourhood function giving the link structure of the graph, where \( N_i \subseteq D \setminus \{d_i\} \) (i.e. nodes cannot link to themselves).

To perform collective classification of ConVote debates, let \( D \) refer to the aggregate contributions of individual speakers in a debate and let \( E \) represent the set of by-name references between speakers. Represent the “for” and “against” votes using the label set \( L = \{\text{For}, \text{Against}\} \).

4.3 Thomas et al. Dual Classifier Approach

Thomas et al. use a dual classifier approach to collective classification. Recall from Section 2.4.7 that the dual classifier approach requires three steps:

1. **Base classification.** Produce base classifications using two distinct classifiers: (1) a content-only classifier that classifies documents considering only their content; and (2) a relationship classifier that classifies inter-document relationships. In this case, the content-only classifier will give the preference that each speech be classified with one of the two labels For and Against. The relationship classifier will indicate the preference that each by-name reference connects pairs of speakers with SAMELABEL or DIFFERENTLABEL, i.e. both classified as For or Against or one For and one Against.

2. **Normalisation.** Normalise the scores output by the two base classifiers so they can be input into a collective classification algorithm.

3. **Decoding.** Produce final classifications by optimally decoding the content-only and relationship level preferences using a collective classification algorithm.

Figure 4.1 gives a diagrammatic overview of the Thomas et al. approach.
4.3.1 Base Classification

Thomas et al. use linear kernel SVMs as their base classifiers. They use the SVM^light implementation by Joachims (1998).[1]

The content-only classifier is trained to predict For or Against labels using a bag-of-words feature model with binary, unigram features. Thomas et al. select this feature model because it had been shown to perform well on binary sentiment classification tasks (Pang et al. 2002).

The relationship classifier is trained to predict SAMELABEL or DIFFERENTLABEL based on the binary, unigram features found in the text immediately before and after by-name references, which is referred to as the “context window”. This approach follows from the

---

assumption that speakers refer to each other using words that indicate whether they are taking an agreeing or disagreeing position on the debate. The wisdom of this approach is made clear if we look at short context windows around the example text given in the last chapter (Figure 3.2).

The first example by-name reference was from one For speaker to another:

at this time, i would like to recognize the gentleman from texas (mr. hensarling) who has worked tirelessly not only on budgeting . . .

Immediately after the by-name reference, the speaker compliments the speaker being referred to, saying he has *worked tirelessly*. The positive adverb gives a clue about classification of the relationship. The second example is similar, with an Against speaker complimenting the speaker he refers to for his alternative to the bill being debated, which he calls *honest*:

. . . and that is why the gentleman from new york’s (mr. rangel) is the only honest way if you want to protect the middle class . . .

In the third and final example, the use of the mildly pejorative *is not aware* below correctly suggests to a human reader that these two speakers will vote differently:

. . . mr. speaker, perhaps the gentleman from new york (mr. rangel) is not aware that the senate has included the alternative minimum tax in their reconciliation tax package . . .

*Thomas et al.* find good performance by deriving the binary unigram features for their relationship classifier from the 30 tokens before the by-name reference, 20 tokens after, and the name itself.

### 4.3.2 Normalisation

The decision plane distance computed by the content-only SVM is normalised to a positive real content-only classification weight, with outliers flattened:

\[
\begin{align*}
    w_i(1) &= \begin{cases} 
        1 & d_i > 2\sigma^c; \\
        (1 + \frac{d_i}{2\sigma^c})/2 & |d_i| \leq 2\sigma^c; \\
        0 & d_i < -2\sigma^c.
    \end{cases}
\end{align*}
\]  

(4.1)
where $\sigma^c$ is the standard deviation of the decision plane distance, $d_i$, over all of the instances in the debate and $w_i(0) = 1 - w_i(1)$.

The relationship classifier output is processed similarly:

$$w^r(d_i, d_j) = \begin{cases} 
0 & d_{ij} < \theta; \\
\alpha \cdot d_{ij} / 4\sigma^r & \theta \leq d_{ij} \leq 4\sigma^r; \\
\alpha & d_{ij} > 4\sigma^r 
\end{cases}$$

(4.2)

where $\sigma^r$ is the standard deviation of the decision plane distance, $d_{ij}$ over all of the relationships in the debate. The $\alpha$ and $\theta$ variables are free parameters. The former allows the relative importance of content-only and relationship preferences to be altered. If relationship preferences are to be treated as more important than content-only preferences, $\alpha$ will be set to a value greater than 1. Otherwise $\alpha$ will have a value less than 1. The $\theta$ parameter represents a threshold for the SVM decision plane distance, below which the preference will not be counted. The use of this parameter is explained in the next section.

Thomas et al. do not provide much explanation for their normalisation strategy, which I will hereafter refer to as variance-based normalisation. It is clear, however, that the approach works to produce the positive real numbers required for decoding and that it also serves to flatten any outliers down to the given maximum scores of 1 and $\alpha$ respectively.

### 4.3.3 Decoding

To decode the classification preferences, Thomas et al. use the minimum-cut approach described in Section 2.4.6. Recall that, under this approach, a given class assignment $Y$ is assigned a cost that is the sum of per-instance and per-pair class costs derived from the content-only and relationship classifiers respectively:

$$\text{cost}(Y) = \sum_{d_i \in D} w_i(Y_i) + \sum_{(d_i, d_j) \in E: Y_i \neq Y_j} w^r(d_i, d_j)$$

(4.3)

The cost function is modelled in a flow graph where extra source and sink nodes represent the For (1) and Against (0) labels. Each node in $D$ is connected to the source and sink with capacities $w_i(0)$ and $w_i(1)$ respectively. Pairs classified in the SameLabel class are linked with capacities $w^r(d_i, d_j)$. 
An exact optimum and corresponding overall classification is efficiently computed by finding the minimum-cut of the flow graph. There are a variety of algorithms that can do this. I use the Push-Relabel algorithm as implemented by Cherkassky and Goldberg (1995).

A note about corpus structure: Recall from Section 3.2.3 that I have elected to classify a single instance for each speaker where Thomas et al. focused on using a single instance for each speech. To handle this difference, Thomas et al. added a component to their optimisation function to assign infinite cost to classifications that placed speeches by the same speakers in different classes. I omit this here because, as discussed in Section 3.2.3, it is more helpful to focus on this simpler construction of the task.

4.3.4 Tuning

Thomas et al. use a separate set of corpus data to tune the α and θ parameters. This is done by repeatedly evaluating the tuning set performance of the classifier using different parameter values to find those that yield best performance. These parameter values are then used for the final classification. The search space for the α parameter is between 0, where relationship preferences will be disregarded entirely, and some large positive number, a_max, where all relationship preferences will be observed in the final classification. The θ parameter is allowed to take one of two values. In the normal case it is set to 0 to exclude all DIFFERENTLABEL relationship classifications. This is necessary because the minimum-cut method of decoding only supports SAMELABEL preferences. Alternately, θ is set to the mean SVM score for the given debate. This is intended to be a simple way of retaining only “high-precision” relationships, i.e. those that the classifier has most confidently labelled as SAMELABEL.

4.3.5 Heuristics for handling DIFFERENTLABEL relationships

Bansal et al. (2008) develop additions to the Thomas et al. minimum-cut approach to incorporate DIFFERENTLABEL relationship classifications. They use post hoc adjustments of

\[\text{Available for download at http://www.avglab.com/andrew/soft.html}\]
graph capacities based on simple heuristics. Two of the three approaches they trial appear to offer performance improvements:

**The SetTo heuristic:** This heuristic works through the set of graph edges, $E$, in order and tries to force $Y_i$ and $Y_j$ into different classes for every different label $(d_{ij} < 0)$ relationship classifier output where $i < j$. It does this by altering the four relevant content-only weights, $w_i(1)$, $w_i(0)$, $w_j(1)$, and $w_j(0)$. Assume without loss of generality that the largest of these values is $w_i(1)$. If this preference is respected, it follows that $Y_j$ should be assigned class Against. Bansal et al. instantiate this chain of reasoning by setting:

- $w_i(1) := \max(\beta, w_i(1))$
- $w_j(1) := \max(\beta, w_j(1))$
- $w_i(0) := 1 - w_i(1)$
- $w_j(0) := 1 - w_j(1)$

where $\beta$ is the target value $\in (.5, 1]$.

**The IncBy heuristic:** This heuristic is a more conservative version of the SetTo heuristic. Instead of replacing the content-only preferences with fixed constants, it increments and decrements the previous values so they are partially preserved:

- $w_i(1) := \min(1, w_i(1) + \beta)$
- $w_j(0) := \min(1, w_j(0) + \beta)$
- $w_i(0) := 1 - w_i(1)$
- $w_j(0) := 1 - w_j(1)$

where $\beta$ is the increment value $\in (0, 1]$.

### 4.4 Critical Analysis of the Thomas et al. Approach

Thomas et al. demonstrate that their approach gives superior floor-debate classification performance than the equivalent content-only method, without attempting to show that it
performs better than any other collective approaches. There are a number of theoretical limitations to their approach, which I will now discuss. First, I consider limitations that have to do with the minimum-cut decoding technique; second, I consider limitations that have to do with other aspects of the Thomas et al. dual classifier. Finally, I consider the Bansal et al. heuristics.

4.4.1 Limitations of the Minimum-Cut Technique

The limitations of the minimum-cut technique for decoding floor-debate classification problems are three-fold, and correspond with the general limitations of the minimum-cut technique discussed in Section 2.4.6.

**Lack of support for DifferentLabel outputs**

The minimum-cut technique cannot be used to decode DifferentLabel relationship preferences, because there are no efficient methods for solving minimum-cut problems with negative flows. This means that the relationship links that are assigned to the DifferentLabel class have to be discarded. This is a major drawback to the approach. The primary utility in collective approaches lies in their ability to fill in gaps in information not picked up by content-only classification. All available relationship information should be applied to this end, so we need models capable of accepting both SameLabel and DifferentLabel relationships.

Fortunately, the impact of this shortcoming is partially mitigated when classifying the ConVote Corpus. Inspection of the corpus shows that approximately 80% of by-name references indicate agreement. In Section 4.8 I will show experimental results that indicate relatively poor performance on the 20% of DifferentLabel links in the corpus, which further mitigates the loss associated with using minimum-cut.

**Lack of classifier confidence information**

The minimum-cut technique does not produce a measure of classifier confidence. As noted in Section 2.4.6 this kind of information is often very useful. Without it, the Thomas et al. classifier cannot be used to give a human user a guide as to which speeches are more or less likely to be classified correctly. Similarly, the classifier is
less useful as a component of a meta-classifier because it can only provide a binary feature.

**Lack of support for multi-class problems**

Floor-debate classification, as it is posed in this thesis and by [Thomas et al.], is a binary classification task. The lack of multi-class support in the minimum-cut technique is not therefore a direct problem. However, it would be quite possible to extend the task to include additional tasks. For example, it might be desirable to have an Abstain class, to try to detect speakers who abstain from voting. Alternately, it could be helpful to add a CHANGEDMIND class, to detect speakers who change their minds in the course of discussion and who therefore can’t be classified as For or AGAINST on the basis of their total contributions to the debate. In these cases it would be necessary to choose a new decoding approach.

### 4.4.2 Limitations of the Normalisation Method

I note three theoretical difficulties with the normalisation approach adopted by [Thomas et al.]:

**Arbitrary outlier grouping**

The formulae for the normalisation techniques incorporate thresholds that set all SVM scores beyond a certain level of confidence to a common maximum: content-only scores with absolute values that are greater than twice the standard deviation of scores in the debate are set to 1 and 0 for the positive and negative classes respectively; relationship classifier scores greater than 4 times the standard deviation of positive scores in the debate are all set to $\alpha$. These values appear to be arbitrary.

**Unjustified dependence on variance**

Both the content-only and relationship classifier normalisation techniques set normalised values that depend on the relationship between the raw score and the variance of the scores within the debate. The is questionable because, intuitively, the strength of a classification should be independent of variance. As a case in point, consider a set of instances in a debate all classified as similarly very weak positives
by the SVM. The Thomas et al. normalisation method would assign these the maximum score because of their low variance, in effect treating weak links as very strong ones.

**Necessity for tuning**

The $\alpha$ and $\theta$ parameters have to be tuned for the relationship between content-only and relationship preferences to be handled correctly. This is computationally expensive. Moreover, parameter tuning is likely to become a source of inaccuracy in cases where the tuning and test debates have dissimilar link structures. For example, if the tuning debates tend to have fewer, more accurate links, the $\alpha$ parameter will be higher. This will not produce good results if the test debates have more frequent, less accurate links.

### 4.4.3 Limitations of the Bansal Heuristics

There are four obvious theoretical limitations with the heuristics proposed by Bansal et al. for handling DIFFERENTLABEL relationship classifications using minimum-cut:

**Arbitrary graph structure**

The most obvious problem with the Bansal et al. heuristics is the arbitrary nature of the manipulations, which produce a flow graph that has an indistinct relationship to the outputs of the two classifiers.

**Tuning difficulty**

Bansal et al. trial a range of target $SetTo$ and $IncBy$ values, with varying impacts on performance. No attempt is made to demonstrate a method for choosing a good target value and it is not clear that the same tuning approach used to set $\alpha$ and $\theta$ would be successful. In any case, having a third parameter to tune would make the process more time-consuming and increase the risks of incorrect tuning.

**Inability to handle instances with multiple DIFFERENTLABEL relationships**

As Bansal et al. point out, proceeding through the set of graph edges, $E$, in order
means that earlier changes may be undone for speakers who have multiple DIFFERENTLABEL relationships. The correct behaviour in this context would be to choose a new value that takes into account both links.

**No embodiment of relationship classifier confidence**

The confidence of the relationship classifier is not embodied in the graph structure. The most marginal DIFFERENTLABEL relationship, classified just on the negative side of the decision plane, is treated identically to the most confident one furthest from the decision plane.

### 4.5 Alternative Dual Classifier Approaches

In this section I describe new dual classifier approaches that use the decoding methods based on Markov random fields introduced in Section 2.4.3: loopy belief propagation and mean-field. These two methods offer similar capabilities to the minimum-cut method, but without the same shortcomings. They both support DIFFERENTLABEL preferences, classifier confidence, and multi-class problems.

Unlike the minimum-cut technique, which uses a “flow” concept, Markov random field techniques are inherently probabilistic. This means that it makes sense to have classification preferences expressed as probabilities. A simple solution to obtaining these would be to switch to a classification model that outputs probabilities, like linear regression. However, I elect to retain the same SVM base classifiers used by Thomas et al. for two reasons: (1) using the same base classifiers provides for better contrast between collective classification methods, which will be the sole cause of any performance variations; and (2) SVMs seem to be a more popular and possibly better-performing choice for sentiment classification tasks.³ To achieve this, I adopt the method for probabilistic normalisation of SVM scores described in Section 2.4.8. Recall that this works by fitting the decision plane distances to a sigmoid using maximum likelihood estimation over the training data.

By applying this technique to the base classifiers, we can produce the following clique

³Some preliminary experiments I conducted with linear regression using ConVote supported this conclusion, with SVMs giving 1-2% better accuracy scores.
potentials:

\[ \psi_i(\text{For}) = P(Y_i = 1|d_i) \]  
\[ \psi_i(\text{Against}) = 1 - P(Y_i = 1|d_i) \]  
\[ \psi_{ij}(\text{For, For}) = 1 - P(Y_i = Y_j|d_{ij}) \]  
\[ \psi_{ij}(\text{Against, Against}) = \psi_{ij}(\text{For, For}) \]  
\[ \psi_{ij}(\text{For, Against}) = 1 - \psi_{ij}(\text{For, For}) \]  
\[ \psi_{ij}(\text{Against, For}) = \psi_{ij}(\text{For, Against}) \]  

where \( d_i \) is the decision plane distance output by the content-only classifier, \( d_{ij} \) is the decision plane distance output by the relationship classifier, and \( n_p \) is a function converting a decision plane distance to a probability.

This approach addresses each of the shortcomings of the Thomas et al. normalisation method described in the last section. In summary: (1) outliers are now handled naturally by the sigmoid; (2) scores within a debate are allocated independently to each other; (3) no tuning is necessary because the normalised values for the two different types of classification preference are already in mutual proportion. This is a convenient side-effect of the use of probabilities as opposed to arbitrary flow quantities.

Figure 4.2 gives a diagrammatic overview of the alternative dual classifier.

4.5.1 Optional Tuning Step

Probabilistic SVM normalisation makes a tuning step theoretically unnecessary. There are, nevertheless, two reasons to implement a tuning step:

1. Tuning may help to correct a bias in one of the classifiers. It may be the case that one of the base classifiers tends to be more confident about its predictions relative to its actual performance. If this is the case, the probabilities that are supplied to the decoding step will be mismatched and performance will be diminished. A tuning step might enable this to be corrected by “dampening” the confidence of the content-only classifier or relationship classifier.
2. Given that the Thomas et al. method uses a tuning step, it seems reasonable to try to compare the approaches in an equivalent setting.

Let $\psi'_i$ and $\psi'_{ij}$ now refer to the dampened versions of the content-only and relationship preference functions respectively, such that:

\[
\psi'_i(\text{For}) = \psi_i(\text{For}) + \frac{\min(0, \gamma)(\psi_i(\text{For}) - \psi_i(\text{Against}))}{2} \tag{4.10}
\]

\[
\psi'_{ij}(\text{For, For}) = \psi_{ij}(\text{For, For}) - \frac{\max(0, \gamma)(\psi_{ij}(\text{For, For}) - \psi_{ij}(\text{For, Against}))}{2} \tag{4.11}
\]

where $\gamma$ is the dampening parameter $\in [-1, 1]$, $\psi'_i(\text{Against}) = 1 - \psi'_i(\text{For})$, $\psi'_{ij}(\text{Against, Against}) = \psi'_{ij}(\text{For, For})$, and $\psi'_{ij}(\text{Against, For}) = \psi'_{ij}(\text{Against, For}) = 1 - \psi'_{ij}(\text{For, For})$. 

Figure 4.2: The Markov random field dual classifier with probabilistic SVM normalisation.
This approach works by reducing the difference between the preferences for the two classes (For and Against or SameLabel and DifferentLabel) by an amount that is proportional to the absolute value of the dampening parameter. If the dampening parameter is less than 0 only the content-only preferences will be dampened, which in effect gives more relative weight to relationship preferences. If the dampening parameter is greater than 0 only the relationship preferences will be dampened, which gives more relative weight to the content-only preferences.

### 4.6 Global Approach based on Markov random fields

Another way to address the limitations of the dual classifier is to move to a global approach based on Markov random fields (see Section 2.4.3). This approach has similar advantages to the dual classifier approach based on Markov random fields:

1. DifferentLabel outputs are inherently supported, because Markov random fields are probabilistic.
2. Classifier confidence is supported, again, because Markov random fields are probabilistic.
3. Multi-class support is provided.
4. Unlike the dual classifier approach, normalisation is not required because the model does not need to integrate the results of component classifiers.

The key disadvantage of the global approach compared to dual classifier approaches is that learning takes at least an order of magnitude longer (see Section 2.5.2).

Stoyanov and Eisner (2012) achieve very good results for the ConVot classification using loopy belief propagation and a learning algorithm called minimum-risk training. Where normal learning algorithms assume exact inference, minimum risk training is designed to compensate for the error that is inherent in approximate inference methods. Stoyanov and
Eisner show that this technique gives a significant improvement over the standard log-likelihood parameter learning approach.

The results reported by Stoyanov and Eisner are for the same holdout configuration used by Thomas et al. and the same bag-of-words and context window bag-of-words feature model. As discussed in Section 3.2.3, I believe the holdout configuration to be sub-optimal, and will use 53-fold cross-validation instead. In the experiments section of this chapter (Section 4.8) I will show the performance of the Stoyanov and Eisner approach in this alternative configuration. I use an implementation of minimum-risk made available online by Stoyanov and Eisner.

4.7 Alternative Iterative Classifier Approaches

In this section I describe alternatives to the Thomas et al. approach that use iterative classifiers.

Recall from Section 2.4.1 that the iterative classifier approach has three major components:

1. Base classification. Produce base classifications using a content-only classifier. As with the dual classifier approach, the content-only classifier will give the preference that each speech be classified with one of the two labels For or Against. There is no relationship classifier.

2. Addition of relational features. Produce local vectors by adding relational features to the vectors previously used for content-only classification.

3. Iterative re-classification. Use a local classifier to classify the new feature vectors. Update the relational features after each iteration to reflect new class assignments. Repeat until class assignments stabilise or a threshold number of iterations is met.

Figure 4.3 gives a diagrammatic overview of the iterative classifier approach to floor-debate classification.

---

https://sites.google.com/site/ermasoftware/
4.7.1 Base Classification

As with the alternative dual classifier approach, I use a content-only base classifier that is identical to the one used by [Thomas et al.], a linear kernel SVM with binary unigram features. This allows any changes in performance to be directly attributed to the collective classification method.

4.7.2 Relational Features

Recall from Section 2.4.1 that the local classifier, $M_{AR}$, accepts both the document feature vector, $\vec{x}_i$, and relational feature vector, $\vec{f}_i$.

In any iterative classifier, the key design feature is the choice of relational features. In this case the challenge is to represent the by-name reference information in a way that will be most helpful to the classifier. I will use three models: (1) count features; (2) binary
context window features; and (3) term frequency context window features.

**Count features:** Count features have already been described in Section 2.4.2. The relational feature vector for count features, $\vec{f}_c^i$, contains two elements, representing the counts of neighbours that have the For and Against labels respectively. The value of these features is denoted by the function $f^c(i, l)$ which counts the number of documents in the neighbourhood $N_i$ that have label $l$. For example, if the document $d_i$ has one neighbour with label For, the value of $f^c(i, \text{For})$ will be 1.

Assuming that the majority of by-name references occur between speakers who vote the same way, a high value for either of the two features will be strongly correlated with the corresponding class.

**Context window features:** I introduce context window relational features as the relational feature equivalent of features used in the Thomas et al. relationship classifier.

Context window features come from the product space $L \times W$ where $W$ is the set of unigrams used in by-name reference context windows. Feature values are determined by the function $f^{cw}(i, w, l)$, which counts the number of documents in the neighbourhood $N_i$ that have label $l$ and are linked to $d_i$ via a context window that includes the unigram $w$. For example, if the document $d_i$ has one neighbour with label For and the context window relating $d_i$ to that neighbour makes use of the word agree once, the value of $f^{cw}(i, \text{agree}, \text{For})$ will be 1.

As with the Thomas et al. relationship classifier, this model implements the intuition that a speaker’s voting intention can be deduced from the words used in by-name references to or from a speaker with a known vote.

I will experiment with two kinds of context window features: (1) binary features in which the feature value is always 1 if $f^{cw}(i, w, l) > 0$; and (2) frequency count features in which the feature value is equal to $f^{cw}(i, w, l)$.

Figure 4.4 shows count and context window iterative classifier features for three interlinked sample speeches. The speeches shown are real examples from the ConVote Corpus (see Figure 3.2) with two simplifications: (1) by-name references to speakers other than the three shown have been excised; and (2) the context windows (shown next to the arrows representing by-name references) have been shorted for readability. Observe the following:
As stated in the definition above, there are only two count features, \( f^c(i, \text{For}) \) and \( f^c(i, \text{Against}) \).

The \( d_2 \) and \( d_3 \) nodes both have the \text{Against} count feature only, because their only relationship is with the \( d_1 \) node, which has the \text{Against} label.

The \( d_1 \) node has both the \text{Against} and \text{For} count features, because it is connected to a node from each class.

Some of the context window features have a value of 2 and some have the value 1. This indicates that they are frequency count context window features. If they were binary context window features all would have the value 1.

The \( d_2 \) node has a set of context window features derived from the relationship with the \( d_1 \) node. Each has the \text{Against} label, corresponding with the label of the \( d_1 \) node. There is one feature for each unique token type in the context window. The context window features for the \( d_3 \) node follow the same pattern.

The \( d_1 \) node has context window features with the \text{Against} and \text{For} labels due to its relationships with the \( d_2 \) and \( d_3 \) nodes respectively.

The classes shown are from the corpus labels, so this is a training scenario. During classification, the classes used to derive the features are taken from the previous iteration, not from corpus labels.

### 4.8 Experiments

In this section I compare the performance of the [Thomas et al.] and [Stoyanov and Eisner] approaches with the alternative dual classifier and iterative classifier approaches.

In the tables and analysis that follow accuracies are reported as percentages of instances correctly classified. All experiments are conducted using a 53-fold cross-validation configuration, where each of the 53 debates is classified in turn using the other 52 debates as training data. Micro-averaging is used, which means that the figures quoted refer to the
Chapter 4: Collective Document Classification with Explicit Relationships

Figure 4.4: Count and context window iterative classifier features for three inter-linked sample speeches. Context windows are shortened for readability.
overall proportion of instances that are correctly classified, rather than an average proportion across debates.

When it is quoted, statistical significance has been determined using approximate randomisation with $p < .05$. Approximate randomisation is a computer intensive model for hypothesis testing that is designed to work in cases where the sampling distribution of the test statistic is not known. This is in contrast to analytical significance testing approaches like the Student’s $t$-test, which assume a distribution (Nooreen 1989). Clearly, comparative evaluation of performance of text classifiers using accuracy scores is an example of a situation where the sampling distribution is not known, so approximate randomisation is a good fit. I use stratified approximate randomisation with 10000 iterations.

Two baseline scores are shown in the tables for collective classification results. “Majority” gives the performance of the simplest possible classifier, which classifies every instance with the label that is most frequent in training data. “Content-only” gives the performance of the bag-of-words linear SVM used to perform base classification.

Results for collective classification are broken down into three categories:

1. **Connected.** Connected instances are those that are the source or target of at least one by-name reference. Accuracies for collective instances are shown separately because it is only these instances that can be affected by collective classification techniques.

2. **Isolated.** Performance over isolated instances will always be the same, regardless of what collective classification technique is applied.

3. **All.** This category gives the performance on the overall task. Use this figure to understand the impact of the technique on the task as a whole, as it might be perceived by an end-user.

### 4.8.1 Relationship Classifier Performance

Table 4.1 shows the performance of the relationship classifier. Recall that this classifier is trained to use the context-windows around by-name references between speakers to determine whether two speakers are SAMELABEL or DIFFERENTLABEL. It is used by the dual classifiers.
Chapter 4: Collective Document Classification with Explicit Relationships

<table>
<thead>
<tr>
<th></th>
<th>No. of relationships</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>All relationships</td>
<td>1549</td>
<td>81.28</td>
</tr>
<tr>
<td><strong>DifferentLabel</strong> relationships</td>
<td>89</td>
<td>69.66</td>
</tr>
<tr>
<td><strong>SameLabel</strong> relationships</td>
<td>1460</td>
<td>81.99</td>
</tr>
</tbody>
</table>

Table 4.1: Relationship classifier performance for all relationships and for relationships classified as **DifferentLabel** and **SameLabel**.

Overall performance is good, with 81.28% of the 1549 by-name references in the corpus correctly classified. It is important to note that there are only 89 by-name references that are classified as **DifferentLabel**. Of these, just 69.66% are correct. This result suggests that there may be minimal benefit to be obtained by moving to a model that supports **DifferentLabel** relationships. Section 4.8.3 and Section 4.8.4 will give the experimental results that address this question.

4.8.2 Local Classifier Performance

Table 4.2 shows the performance of the local classifiers built with three different types of relational features: count features, binary context window features, and frequency count context window features. This test measures the ability of the classifiers to predict the labels of speeches when they are given perfect information about the labels of their neighbours. These results represent a rough upper bound for the performance of the iterative classifiers, which obtain neighbour labels iteratively.

Frequency count context window features perform best, with 90.01% accuracy over connected instances, a 14% absolute improvement on the content-only baseline. Binary context window features are almost as good, with 89.94% accuracy over connected instances. Count features are the lowest performing variety of relational feature, with 87.28% accuracy.

4.8.3 Collective Classifier Performance without Tuning

Table 4.3 contrasts the performance of the Thomas et al. and Stoyanov and Eisner approaches with the alternative dual classifier and iterative classifier formulations described
Table 4.2: Local classifier performance for count features, binary context window features and frequency count context window features. Neighbour labels are given by oracle.

<table>
<thead>
<tr>
<th></th>
<th>Connected</th>
<th>Isolated</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority</td>
<td>52.86</td>
<td>48.30</td>
<td>51.44</td>
</tr>
<tr>
<td>Content-only</td>
<td>76.15</td>
<td>79.17</td>
<td>76.40</td>
</tr>
<tr>
<td>Count</td>
<td>87.28</td>
<td>79.17</td>
<td>84.76</td>
</tr>
<tr>
<td>Binary context window</td>
<td>89.84</td>
<td>79.17</td>
<td>86.52</td>
</tr>
<tr>
<td>Frequency count context window</td>
<td>90.01</td>
<td>79.17</td>
<td>86.64</td>
</tr>
</tbody>
</table>

earlier. To create a level playing field, none of the classifiers uses any tuned parameters. I will discuss the motivation for this comparison further in Section 4.9.

To obtain un-tuned results using variance-based normalisation, I set $\alpha = 1$ and $\theta = 0$. These settings mean, respectively, that relationship preferences are considered to have equal weight with content-only preferences, and that no relationship information is discarded.

To get un-tuned results from the SVM classifiers I simply use SVM\textsuperscript{light} with default parameters. For the global formulations I follow Stoyanov and Eisner by setting the regularisation parameter to the default value $0.1$.

The table shows that each of the major categories of approach has at least one variant that gives statistically significant improvements over the content-only baseline. The Thomas et al. approach (dual classifier with variance-based normalisation and minimum-cut decoding), gives a 1.65% gain over the baseline, yielding a score of 78.05% for the overall task. The Stoyanov and Eisner global approach gives a 3.35% gain over the baseline, leading to a score of 79.75%.

Of the methods using the Bansal et al. (2008) heuristics for incorporating Different-Label relationships, only the IncBy(0.05) and IncBy(0.15) variants provide statistically significant improvements over the Thomas et al. approach. IncBy(0.15) is the best, giving a 1.03% gain over connected instances.

The three dual classifier methods using probabilistic normalisation far outperform the variance-based normalisation alternatives. Minimum-cut, loopy belief propagation, and mean-field give 6.83%, 7.00%, and 6.23% absolute gains for connected instances over the Thomas et al. approach, respectively. These three also give statistically significant
improvements over the Stoyanov and Eisner approach, with absolute gains over connected instances of 2.99%, 5.29% and 5.72% and respectively.

The best overall performer for the task is the loopy belief propagation variant, with a score of 82.87%.

The iterative classifier methods have variable performance, with all variants being inferior to two best dual classifiers by a significant margin. Count features give a 1.03% advantage over the Thomas \textit{et al.} approach for connected instances, which is not a statistically significant improvement. Binary context window features do slightly better with a 1.28% improvement, which is statistically significant. Both of these variants do worse than the Stoyanov and Eisner approach.

Frequency count context window features give the best iterative classifier performance, besting all of the variance-based normalisation dual classifiers with an overall score of 80.93%. This represents statistically significant absolute gains of 2.88% and 1.18% over the Thomas \textit{et al.} and Stoyanov and Eisner approaches respectively.

The relative scores for the different iterative classifiers accord with the local classifier results in the previous section.

### 4.8.4 Collective Classifier Performance with Tuning

In this section I assess the performance of each of the methods with tuned parameters. The methods using variance-based normalisation have their $\alpha$ and $\theta$ tuned by use of cross-validation over the training fold. Specifically, a grid search is used to find the parameters that give the best micro-average performance for a 52-fold cross-validation experiment over the debates in the training fold. The methods using probabilistic normalisation have their $\gamma$ parameter tuned in the same way.

For the Stoyanov and Eisner approach I use the same technique to select the best value for the regularisation parameter.

For simplicity I continue to use SVM$^{\text{light}}$ with default parameters. Since there is no other tuning option for the iterative classifier approaches, scores will be identical to those given in the last section.

Results are shown in Table 4.4.
Table 4.3: Collective classification performance with no tuned parameters. Results in the “All” category marked with $\star$ are statistically significant improvements over the Thomas et al. approach (dual classifier with variance-based normalisation and minimum-cut decoding). Results in the “All” category marked with $\dagger$ are statistically significant improvements over the Thomas et al. approach and the Stoyanov and Eisner approach (global classifier with minimum-risk learning and loopy belief). All of the global, dual classifier and iterative classifier approaches give statistically significant improvements over the content-only baseline. Statistical significance is determined using approximate randomisation with $p < 0.05$.

The Thomas et al. approach (dual classifier with variance-based normalisation and minimum-cut decoding), gains an extra 2.12% accuracy by the addition of tuning, yielding an overall score of 81.17%.

Surprisingly, the Stoyanov and Eisner approach (global classifier with minimum-risk learning and loopy belief) loses 0.64% accuracy by the addition of tuning, and is no longer superior to the Thomas et al. approach.

None of the methods using the Bansal et al. (2008) heuristics for incorporating different label relationships provide any improvement over the Thomas et al. approach. SetTo(1.0) obtains the same score as the Thomas et al. approach.

The advantage of the three dual classifier methods using probabilistic normalisation is
now much less pronounced. The minimum-cut variant now performs identically to its tuned variance-based normalisation equivalent. Loopy belief propagation and mean-field give 2.30% and 2.73% gains over connected instances respectively. The best overall performer for the task, including tuned and un-tuned approaches, is the tuned variant of loopy belief propagation, with a score of 83.05%.

Both the tuned and un-tuned variants of the loopy belief propagation and mean-field approaches achieve statistically significant improvements over the tuned versions of the Thomas et al. and Stoyanov and Eisner approaches.

It is important to note that adding tuning to the probabilistic normalised versions of minimum-cut and loopy belief propagation actually decreases performance, by 2.30% and 0.17% respectively. This is evidence that the probabilities output by the content-only and relationship classifiers were better matched before tuning than after. Tuning is an inexact process that depends on the training data as an approximation for the test data, so this minor variation in performance is not surprising.

The iterative classifier methods, which do not support tuning, are now weaker than all of the dual classifier approaches, except the Thomas et al. approach, by statistically significant margins.

4.9 Conclusions

In this section I analyse the results of my experiments and present final conclusions.

The most fundamental conclusion of this chapter is that the new dual classifier techniques achieve significant improvements over the previous state-of-the-art of ConVote floor-debate classification. The score of 83.05% obtained by the dual classifier with tuned probabilistic SVM normalisation and loopy belief propagation decoding is the best recorded result for collective classification of ConVote using bag-of-words features and 53-fold cross-validation. This is the new state-of-the-art for the task.

Beyond this task-specific conclusion, the experimental results provide evidential support for five additional conclusions in relation to collective classification in general:

Probabilistic SVM normalisation is superior to variance-based SVM normalisation
Table 4.4: Collective classification performance with tuned parameters. Results in the “All” category marked with † are statistically significant improvements over the Thomas et al. approach and the Stoyanov and Eisner approach (global classifier with minimum-risk learning and loopy belief). All of the global, dual classifier and iterative classifier approaches give statistically significant improvements over the content-only baseline. Statistical significance is determined using approximate randomisation with $p < 0.05$.

In Section 4.4.2 I described three problems with the variance-based approach to SVM normalisation: (1) outliers are grouped arbitrarily; (2) the dependence on variance is unjustified; and (3) computationally expensive tuning is necessary to obtain best results.

In Section 2.4.8 I introduced probabilistic SVM normalisation as an alternative approach that appeared to address the theoretical shortcomings of variance-based normalisation. This approach appeared to be a simple alternative with consistent handling of outliers, no dependence on variance, and no requirement for tuning.

The experimental results presented in the previous section provide evidential support for this analysis. In the test for un-tuned performance, the variants that used probabilistic normalisation all did far better than those that used variance-based normali-
sation, with the minimum performance delta over connected instances being 4.18%. With parameter tuning enabled, the best variance-based normalisation variant was still 2.73% short of the best probabilistic normalisation variant over connected instances.

The tuned and un-tuned scores for probabilistic normalisation were roughly equivalent. This is an important result, because it suggests that the extra computation required for parameter tuning can be safely dispensed with. Parameter tuning by cross-validation over the training fold consumed by far the majority of total computation time for any of the methods trialled. For ConVote, parameter tuning for each one of the 53 evaluations requires a 52-fold cross-validation evaluation over the training fold. Parts of these 52 evaluations have to be repeated for each tested parameter value. On my test hardware, un-tuned experiments took a matter of minutes, compared to many hours for their tuned equivalents, so the saving is significant.

**Loopy belief, mean-field and minimum-cut are all legitimate choices for decoding**

In Section 4.4.1 I described three theoretical problems with the use of minimum-cut as a decoding technique: (1) there is no support for the DIFFERENTLABEL relationship classifier output; (2) no measure of classifier confidence is available; and (3) there is no support for multi-class problems. It is only the first of these limitations that has any bearing on ConVote floor-debate classification, and the results show that minimum-cut is competitive with loopy belief propagation and mean-field despite this. In the un-tuned configuration, minimum-cut is very slightly superior (0.41%) to mean-field and very slightly inferior (0.12%) to loopy belief propagation. With tuning, minimum-cut is 1.58% behind loopy belief propagation and 1.88% behind mean-field.

This similarity in performance is probably attributable to the fact that DIFFERENTLABEL relationship classifications are neither very frequent nor very reliable in this experiment. As noted in Section 4.8.1 there are only 89 by-name references that are classified as DIFFERENTLABEL, of which just 69.66% are correct.

Loopy belief propagation and mean-field should be first choices for multi-class prob-
lems or problems where good DIFFERENTLABEL relationship information is available. Where this is not the case, all three algorithms are legitimate choices and can be easily substituted for each other as required.

**There is no reason to use the Bansal et al. heuristics**

My experimental results show that the IncBy and SetTo heuristics do not provide significant and reliable performance gains over the Thomas et al. approach. In an un-tuned configuration the IncBy algorithm provides small but statistically significant improvements, but these are markedly inferior to the probabilistic normalisation approaches and also inferior to the frequency count context window iterative classifier. In a tuned configuration both heuristics serve only to degrade the results obtained using the Thomas et al. approach.

In Section 4.4.2 I described four problems with the heuristics: (1) they produce an arbitrary graph structure; (2) they are difficult to tune; (3) they fail to handle multiple DIFFERENTLABEL relationships correctly; and (4) they fail to take account of relationship classifier confidence. My results bear out this analysis. Bansal et al. presented experimental results that supported their conclusion that both heuristics may provide improvements. I contend that their choice of a single, fixed data split has been misleading and that the more statistically robust experiments presented here carry more weight.

**Dual classifiers are superior to iterative classifiers**

The experimental results for both tuned and un-tuned configurations showed that dual classifiers performed better than iterative classifiers. In the un-tuned configuration, the best dual classifier (probabilistic normalisation, loopy belief propagation) performed 2.81% better than the best iterative classifier (frequency count context window) over connected instances.

The best tuned dual classifier (probabilistic normalisation, mean-field) did 3.07% better.

This is not to say that iterative classifiers are without promise. Their conceptual simplicity is appealing. The most compelling point in their favor is their ability to unify
content-only and relationship features in a single classifier. Conceptually speaking, such an approach should allow the two types of features to inter-relate in more nuanced ways. A case in point comes from the present use of a fixed size context window to build a relationship classifier. Future approaches may be able to do away with this arbitrary separation of features by training a local classifier to consider all words in terms of their impact on content-only classification and their relations to neighbours.

**Frequency counts are best for relational representation of context windows**

Frequency count context window features were the best of the three types of relational features trialled. These features provide the most nuanced information about the words used in the context of by-name references, and as such, their superior performance is not a surprise. Binary context window features are second best, and count features are the least helpful.

The relative performance of the three feature types is roughly as would be expected. Context window features do better than count features because they consider the words used for by-name references; count features can only rely on the fact that the majority of by-name references are between speakers who vote the same way.

It is interesting to note that frequency count context window features are better than their binary equivalent. This is counter to conventional wisdom in sentiment analysis, which suggests that binary features are better, and is best explained by the fact that context windows are much shorter than normal documents.

### 4.9.1 Differences over Earlier Work

As noted in the front-matter of this thesis, the experiments and analysis in this chapter are directly derived from my previously published work ([Burfoot et al. 2011](#)). Readers of that work may notice some differences in my experimental configuration and conclusions. In this short section I will describe the experimental differences and explain the divergence in conclusions.
In the Burfoot et al. paper I used $10 \times 10$-fold cross validation, i.e. 10 runs of 10-fold cross validation using different randomly assigned data splits. I realised a problem with that approach when preparing for this chapter. As noted in Section 3.2.2, the ConVote corpus is comprised of 1699 aggregated speeches evenly split between the For and Against classes. In a normal text classification experiment on ConVote it would simple to produce a stratified 10-fold cross validation experiment by splitting these instances evenly across the 10 folds. This becomes problematic in a collective experiment, where the 1699 speeches are in fact immutably assigned to only 53 debates. In the correct, theoretical view, it is the 53 debates, not the 1699 speeches, that need to be assigned to folds. Ignoring this would mean throwing away some portion of the inter-document relationships that occur within debates.

The problem with my earlier approach arises because of the need to generate folds in variable sizes so that debates are not split up. The experiments that validate $10 \times 10$-fold cross validation as having higher power and replicability than alternatives do not apply to this scenario (Bouckaert 2003).

I have chosen the present configuration to acknowledge that the normal properties of $10 \times 10$-fold cross validation can not apply to my experiments. The leave-one-debate-out 53-fold cross-validation configuration is a simple alternative that minimises bias by making the maximum amount of data available for training and reduces variance by averaging over many repetitions.

The second difference concerns tuning. In the paper I did not attempt any tuning of potentials normalised with probabilistic SVM normalisation, so the discussion in this chapter about tuned and un-tuned configurations is new. The more crucial difference between the paper and this chapter is my choice of tuning data. In the paper I used a randomly allocated 10% of the training data for tuning of variance-based normalised potentials. In this chapter I use cross-validation over the training fold. The resultant difference in performance is marked. In the paper, the score for the Thomas et al. method was 3.53% worse than the score for the un-tuned probabilistic normalisation approach with loopy belief propagation. In this chapter the difference is 1.7%.

My conclusion in the paper was that “[t]he superior performance of the dual classifier approach with loopy belief propagation and mean-field suggests that either algorithm could
be considered as a first choice for collective document classification”. This now seems too strong. This chapter has shown that minimum-cut with variance-based normalisation has closer to comparable performance with the alternatives when sufficient tuning data is available. It has also shown that minimum-cut can do as well as the alternatives when using un-tuned probabilistic normalisation.

4.10 Future Work

The purpose of the work in this chapter has been to show the relative benefits of alternative approaches to collective classification using the ConVote Corpus. Like Thomas et al., I have used very simple bag-of-words base classifiers. This decision reflects the view that collective classification is essentially orthogonal to base classification. It would have been possible to use much more nuanced feature models for the base classifiers without making any changes to the collective classification approach, but this would have been a distraction from the goal of the work. Nevertheless, in future it will be interesting to see how accurately ConVote can be classified by combining state-of-the-art base classifiers and collective classification approaches.

There are several works which have established benchmarks on ConVote using more advanced non-collective approaches (Greene 2007; Sokolova and Lapalme 2008; Balahur et al. 2009; Balahur et al. 2009; Martineau and Finin 2009; Martineau et al. 2009; Yessenalina et al. 2010b). Similarly, there are a great many state-of-the-art approaches to sentiment classification that have not yet been applied to ConVote (e.g. Taboada et al. 2011, AR et al. 2011, Yessenalina et al. 2010a). These could likewise be combined with collective classification approaches.

Probabilistic SVM normalisation offers a convenient, principled way of incorporating the outputs of an SVM classifier into a dual classifier. Another opportunity for future work is to consider normalisation approaches for other classifiers. For example, confidence-weighted linear classifiers have been shown to give superior performance to SVMs on a range of tasks and may therefore be a better choice for collective document classification (Dredze et al. 2008).
There is much potential gain from improvements to the basic relationship classifier approach used here. Techniques for sentence-level sentiment classification may offer immediate improvements (e.g. Yessenalina and Cardie 2011, Maas et al. 2011, Täckström and McDonald 2011). There is also research on sentiment polarity classification of citations in academic literature that may be directly applicable to improving performance with ConVote (Athar 2011).

The experiments in this chapter have also provided further confirmation of the conclusion of Thomas et al. (2006) about the usefulness of collective approaches for predicting the votes of contributors to congressional debates. Significant performance benefits were obtained through the use of relatively simple measures of inter-speaker relationship. There is clearly significant scope for expanding the analysis of congressional debate content to include much more nuanced measures of inter-speaker relationship.

Finally, there is much to be gained from broadening the approach in this chapter to include other test domains that feature explicit inter-document relationships. Web encyclopedias and scholarly publications are two examples of document domains where network structures have already been used to assist classification and where the techniques in this chapter could be applied (Cao and Gao 2005, Gantner and Schmidt-Thieme 2009).
Chapter 5

Collective Document Classification with Implicit Inter-document Relationships

5.1 Introduction

The previous chapter explored the use of explicit inter-document relationships for collective classification. It showed that the by-name references in the ConVote Corpus can be used to significantly improve the performance of a document classifier. It also established a body of knowledge about techniques for conducting collective classification. In this chapter, I aim to apply that knowledge to a more challenging task: using collective classification to improve document classifier performance without the aid of explicit inter-document relationships like by-name references. I will demonstrate a technique by which inter-document relationships can be derived implicitly. I will show also that these relationships can be used to improve document classifier performance.

The experiments in this chapter use the ConVote and Bitterlemons corpora, introduced in Chapter 3. Both corpora are collections of political language: ConVote is a set of transcriptions of U.S. congressional debates; Bitterlemons is a series of essays and interviews giving perspectives on the Israel-Palestine conflict.

This chapter, like the thesis as a whole, begins with the observation that some types of documents have features which are either absent or ambiguous in training data, but which have the special characteristic of indicating relationships between the labels of documents.
Most often, an inter-document relationship indicates that two documents have the same label, but it may also indicate that they have different labels. In either case, classifiers gain an advantage if they can consider these features as well as the normal type of feature that associates a document with a particular label.

The major contribution of this chapter is in showing that useful inter-document relationships can be derived by analysing the similarities between documents within each corpus. In Chapter 3 I postulated that the ConVote and Bitterlemons corpora would be susceptible to this technique because of their unconstrained vocabularies and abundance of task-relevant content. The experimental results presented in this chapter are consistent with that hypothesis.

In this first part of this chapter I present experiments and analysis to demonstrate the presence of inter-document features that are correlated with document labels. I show that these features can be derived from simple measures of similarity based on $n$-gram overlap. I will demonstrate that while these features are not individually very predictive of inter-document relationships, they can be used in combination to give performance that is better than a naive baseline.

In the second part of the chapter I will show that $n$-gram overlap features can be used to improve the performance of content-only classifiers for both ConVote and Bitterlemons. As discussed in Chapter 3, Bitterlemons has no explicit inter-document relationships and so is an ideal candidate for these experiments. For the experiments on ConVote I will use a version of the corpus that does not have the annotations for explicit inter-document relationships used in the previous chapter. I will demonstrate that for both corpora, contrary to expectations created in the previous chapter, the best performers are iterative classifiers and dual classifiers using minimum-cut decoding.

In the final part of this chapter I will present an analysis of the experiments that suggests criteria for applying these approaches to other tasks. I will also identify potential paths for building on the techniques in this chapter to develop more nuanced and accurate techniques for collective classification using implicit inter-document relationships.
5.2 Similarity-based Inter-Document Relationships

In this section I analyse inter-document relationships derived using measures of similarity between documents.

In Chapter 4 I was able to use the context-windows around by-name references to detect both SAME_LABEL and DIFFERENT_LABEL relationships. In this chapter I will limit myself to considering only SAME_LABEL relationships. This approach is based on two assumptions:

1. The more similar two documents are to each other, the more likely they are to have the same label. This is clearly the case with identical documents which in most cases would be guaranteed to have the same label. Certainly in the case of the Con-Vote and Bitterlemons corpora, if it were possible to have documents with identical content they would indicate the same opinion and so would carry the same label. Depending on the measure of similarity being used, it seems reasonable to assume that the documents in a corpus that are more similar will be more likely to carry the same label.

2. The reverse does not apply. In the domains we are concerned with, it is difficult to think of a measure of dissimilarity that would correlate reliably with label assignments. For example, two documents that use substantially different vocabularies do not necessarily have different labels. Two people could argue for the same position for entirely different reasons.

In the remainder of this section I will report three preliminary experiments to explore the usefulness of similarity-based inter-document relationships. In the first experiment (Section 5.2.1), I will address the question: are two documents more likely to have the same label if they use the same rare word?

The second experiment (Section 5.2.3) will ask further: can a measure of rare term overlap be used to reliably find pairs of documents that have the same label?

The third experiment (Section 5.2.4) will build on the answers to the first two questions to ask: are similarity-based measures sufficiently powerful to contribute to document classification performance?
None of these experiments involve collective classification on the ConVote or Bitterlemons corpora. However, they provide a basis for experiments to follow.

5.2.1 Preliminary Experiment 1: Finding Label Agreement with Shared Words

I now proceed to an experiment to measure the extent to which pairs of documents with sameLabel relationships can be detected in our target corpora using the most basic measure of similarity: shared words.

The premise of this experiment is that pairs of documents will be more likely to have the same label if they both use a statistically unlikely word. To quantify “unlikely” I calculate the inverse document frequency (idf) of each word in the corpus as:

\[
idf(t, D) = \log \frac{|D|}{|[d_i \in D : t \in T_i]|}
\]

(5.1)

where \( t \) is a word for which the score is being calculated and \( T_i \) is the set of unique words in document \( d_i \). Put in words, idf is the logarithm of the ratio of the number of documents in the corpus to the number of documents in which the word appears.

A pseudo-code description of the experiment is shown in Algorithm 4. In the algorithm, let \( W \) denote the set of unique 1-grams that appear in no more than half of all documents in the corpus. For ConVote, let \( P \) be the set of all intra-debate pairs \((d_i, d_j)\). For Bitterlemons, let \( P \) be the set of all intra-training-set and intra-test-set pairs \((d_i, d_j)\).

The omission of terms that appear in more than half of the documents in the corpus is designed to save computation by obviating the need to process very frequent terms which do not indicate any kind of meaningful similarity when they overlap. I will use this restriction when selecting terms for the remainder of the chapter. Note that the accuracy calculation may count a single pair of documents multiple times for each of the high-ranked terms it contains. This is so that the experiment can reflect the proportion of high-ranked terms that correspond with label agreement, rather than the proportion of agreeing document pairs that can be identified using high-ranked terms.

It is important to note that there is a slight difference in the logic used to find pairs across the two corpora. For ConVote I include a restriction that requires pairs to appear
Algorithm 4 Relationship Detection with Shared Words

\begin{algorithm}
\begin{algorithmic}
\FORALL{unique 1-grams $w \in W$}
\STATE compute $idf(w, D)$
\ENDFOR
\STATE $i \leftarrow 0$
\WHILE{$i \leq |W|$}
\STATE find pairs $(d_i, d_j) \in P$ that both contain one or more of the top $i$ weighted 1-grams
\STATE calculate the proportion of found pairs that are SAME_LABEL
\STATE $i \leftarrow n + 200$
\ENDWHILE
\end{algorithmic}
\end{algorithm}

within the same debate; shared words are not counted across debates. The effect of this restriction is to significantly reduce the overall number of pairs found. The idea is to take advantage of the topical structure of the corpus by linking high-ranking terms that appear in the same context. This context represents an additional layer of relatedness that may serve to increase relationship detection performance. As discussed in Section 3.3.4, Bitterlemons does not have a topical structure that can be used in this way.

Figures 5.1 and 5.2 show the results of the experiment on the ConVote and Bitterlemons corpora respectively. In both cases a correlation between SAME_LABEL relationships and term rank is evident. When only the 200 highest ranking terms are used, 68% of ConVote pairs and 63% of Bitterlemons pairs have the same label. When the top 8000 terms are used, performance drops to 58% for ConVote and 56% for Bitterlemons, close to the respective random baselines of 52% and 50%.

In opposition to the general trend, there is a peak on the Bitterlemons graph at 1000 terms, indicating that the lower ranked terms in this set perform better than the higher ranked terms. This result is slightly misleading, because the first 800 terms all have the same similarity scores.

The different x-axes on the two graphs reflect the fact that ConVote has more terms that satisfy the criteria, despite the fact that it is limited to shared words within debates. This is presumably attributable to the larger corpus size and the fact that congressional debates tend to include significant amounts of debate-specific language that does not appear elsewhere.
in the corpus.

![Proportion of result set that is SameLabel when ConVote instances are paired based on shared high-idf terms](image)

Figure 5.1: Proportion of result set that is SameLabel when ConVote instances are paired based on shared high-idf terms. The flat line shows baseline performance for random selection of pairs within debates.

### 5.2.2 Analysis of Shared Words

In this short subsection I attempt to explain the usefulness of shared words by giving some specific examples from the corpora. Figures 5.3 and 5.4 show examples of four pairs of speeches that are matched via high-ranked words for ConVote and Bitterlemons respectively. In each case, the terms used have the highest possible rank because they are not used anywhere else in the corresponding corpus.

The diagrams show the matched words underlined, with arrows indicating the relationships between the pairs. I have chosen four examples of SameLabel relationships being correctly detected for each corpus. These examples give a good picture of why the technique works more often than not but they are not a representative sample; there are many examples of false positives and true positives that have no intuitive justification that are not interesting to show here.
In the first example in Figure 5.3, taken from a debate on energy policy, two Democrat members refer to a statement from the Whitehouse about using a magic *wand* to lower gas prices. In both cases the reference is pejorative, and intended to indicate the weakness of Republican energy policy, and hence the bill in question.

The second example is taken from a debate on criminal justice. The two users of the term *misdemeanors* are making a similar point: that the tough sentencing requirements in the bill, intended to deal with gang violence, will unfairly target other, minor offenders.

In the third example the two Republicans cite the same legal precedent featuring the name *lefrois* as part of their support for the bill in question.

In the final ConVote example, a Democrat and a Republican state their support for *delisting* as a goal of the Endangered Species Act. Their agreement on this point appears to be indicative of their agreement in favour of the bill being debated.

The most interesting point about these examples is how debate-specific they seem to be. None of the terms has an obvious positive or negative sentiment or Republican or Democrat slant, but in the context of shared use in a debate it is clear that they are indicators of
agreement.

The Bitterlemons examples in Figure 5.4 are somewhat different. In the first example, both Palestinian authors use the word *succumb* to describe the action of the Palestinian people giving in to Israeli aggression. It seems reasonable that the word *succumb* would be correlated with the Palestinian perspective given the military and organisational dominance of Israel in the conflict.

The second example is similar to the first, with the two Palestinian writers using the term *dignified* as an attribute desired by the Palestinians. Again, it is easy to argue that this word has a Palestinian bias in the context of the Israel-Palestine issue, where the Palestinians are stateless and subject to military occupation.

In the third example, two articles by the Israeli editor, Yossi Alpher, are paired because they contain the rare term *quasi-existential*. It seems harder to argue that this term would be particularly likely to come from an Israeli rather than Palestinian contributor. Nevertheless, its rarity serves as an indicator that both articles belong to the same author, and therefore have the same perspective.

The final example has a clear association with Israel, with the term *survivors* used by the Israeli authors to refer to survivors of the World War II Jewish holocaust.

To summarise, the analysis so far has served to show that rare term overlap can indicate label agreement between pairs of instances. Performance is best in ConVote, where there are notable examples of rare terms indicating agreement in the context of particular debates. Bitterlemons performance is slightly weaker. The example pairs in Bitterlemons show agreement being detected between contributors who use terms that, while rare and therefore unlikely to be used well by a content-only classifier, do still carry an obvious polarity. This difference is as expected given that Bitterlemons does not have a topical structure (see discussion in Chapter 3).

5.2.3 Preliminary Experiment 2:

**Similarity-based Inter-document Relationship Detection**

So far in this section I have established that there is a correlation between document labels and shared rare terms, and explored some examples of how this correlation works. I
Crowley, Joe (D) [against]

the president 's top counselor dan bartlett said this week that there is no magic wand to reduce gas prices.

Emanuel, Rahm (D) [against]

mr. chairman , yesterday the president said , '' i wish i could simply wave a magic wand and lower gas prices tomorrow. ' '

Schakowsky, Jan 'Jim' (D) [against]

this bill would unnecessarily federalize a host of crimes currently and competently handled by the states ; penalize even non-violent crimes and misdemeanors as crimes of violence , including garden variety state offenses like resisting arrest

Scott, Robert (D) [against]

it is the lesser offenders , the children who get in fist fights , committing misdemeanors , who will be subject to the 10-year , mandatory minimum numbers in this bill.

Boehner, John (R) [for]

the ability of the commission to waive a deadline on a case-by-case basis when circumstances warrant it have been drawn into increased legal uncertainty by the recent decision of the u.s. circuit court of appeals for the second circuit in chao v. lefrois builder , incorporated , and

Norwood, Charles (R) [for]

although the second circuit agreed with our view in chao v. russell p. lefrois builder , inc. , 291 f.3d 219 ( 2d cir . 2002 ) , the commission has repeatedly rejected it

Cardoza, Dennis (D) [for]

i fully support the goal of species protection and conservation and believe that recovery and ultimately delisting of species should be the u.s. fish and wildlife service 's top priority under esa.

Hefley, Joel (R) [for]

and so the delisting process started . hopefully , we 'll see it completed sometime in the near future though there is some evidence that fish and wildlife is taking its time in doing so.

Figure 5.3: Tokenised sample text from ConVote showing examples of high-idf word overlap predicting SAME LABEL relationships between pairs of speakers. Segments have been shortened for the purposes of this illustration. Speaker names, votes “for” and “against”, and party affiliations are labelled.
Ahmad Harb (guest) [Palestinian]

Even if we wanted to succumb to Israeli pressure, it is impossible to make a Palestinian teach his child that Jaffa or Haifa or Palestine before 1948 was not his land.

Haidar Abdel Shafi (guest) [Palestinian]

This is being neglected and Sharon is having his way in brutalizing the Palestinian people in the hope that they will succumb and abandon their rights.

Samah Jabr (guest) [Palestinian]

The vision of two states does not meet any minimal ambition of peace, freedom and a dignified future for Palestinians.

Eyad El Sarraj (guest) [Palestinian]

For the Palestinians, this will be their opportunity to feel dignified by responding with the honorable, "Yes, we accept your apology and we accept you."

Yossi Alpher (editor) [Israeli]

the Palestinian suicide bombing campaign that was perceived by Israelis as a quasi-existential threat, and which the Palestinian Authority and the Palestine Liberation Organization did nothing to stop.

Yossi Alpher (editor) [Israeli]

The idea began with the suicide bombings, a quasi-existential threat to Israelis. The fence works. Israelis have every right to defend themselves.

Yisrael Harel (guest) [Israeli]

defensive war in which the Jewish people, scarcely three years after the Holocaust, lost one percent of its sons and daughters, including Holocaust survivors, the invaders were expelled.

Arie Lova Eliav (guest) [Israeli]

There is no parallel here, say, to WWII, where the Nazis destroyed a helpless Jewish people who were not at war with them, after which Germany agreed to compensate the survivors.

Figure 5.4: Tokenised sample text from Bitterlemons showing examples of high-idf word overlap predicting SameLabel relationships between pairs of contributors. Segments have been shortened for the purposes of this illustration. Author names and perspectives are labelled.

will now evaluate the ability of the term-overlap measure to perform SameLabel relationship detection.

Given a set of documents $D$, SameLabel relationship detection is the task of finding pairs of documents $(d_i, d_j)$ such that $Y_i = Y_j$, i.e. the members of each pair have the same
label. Success at this task is measured in terms of two dimensions: (1) recall, the proportion of all pairs that have \( Y_i = Y_j \) that are detected; and (2) precision, the proportion of all detected pairs for which \( Y_i = Y_j \). From the results in Section 5.2.1 above, we can expect that precision may be good at low recall and that it will drop off gradually as recall increases. The focus of this experiment is therefore to try to show good precision at some low level of recall, without worrying about the fact that only a very small portion of the possible \texttt{SAME} \texttt{LABEL} pairs is being returned.

In this experiment I will broaden the approach to include longer terms. I will still look for common words (unigrams), but I will also use 2-grams (bigrams), 3-grams (trigrams), 4-grams and 5-grams. Recall from Section 3.2.5 the two points that I offered in support of the view that task-relevant content in the two corpora can be used to find inter-document relationships:

1. Congressional debaters often seem to borrow content from others on their side, perhaps because they use common sets of speaking notes.

2. Political speech is repetitive. Different contributors on the same side of a debate seem to have a set of specialised words \textit{and phrases} that they use to forward their argument.

If my intuition about matching phrases is correct, longer \( n \)-grams may provide superior precision (because they are more powerful indicators of relationship) but even lower recall (because they overlap less frequently).

It is important to note that the word overlap test did not measure performance over the whole set of pairs. It focused on the performance of overlapping words without considering which pairs they were in. In this experiment overlap of multiple terms in the same pair will contribute to a score for that pair instead of being counted individually.

To combine all shared terms in a given pair into a single similarity score I use the standard cosine similarity metric:

\[
  s(i, j) = \frac{\vec{u}_i \cdot \vec{u}_j}{\|\vec{u}_i\|\|\vec{u}_j\|}
\]  

(5.2)
where $\vec{u}_i$ is the tf.idf weighted vector (Manning et al. 2008) used to represent document $d_i$. I use Boolean term frequencies on the assumption that, since Boolean term frequencies work best for content-only document classification, they will also be best for inter-document relationship detection.

A pseudo-code description of the experiment is shown in Algorithm 5. In the algorithm, let $U$ be the set of all tf.idf weighted vectors in the dataset.

Algorithm 5  Same-class link prediction evaluation
for all $n \in \{1, 2, 3, 4, 5\}$ do
    compute tf.idf weighted vectors, $U$ using term length $n$
    for all $(d_i, d_j) \in P$ do
        compute $s(i, j)$
    end for
    sort $P$ in descending order of similarity
    calculate precision and recall by selecting pairs from $P$
end for

Figures 5.5 and 5.6 show relationship detection performance for ConVote and Bitterlemons respectively. The trend is similar to the common words test: the document pairs with highest similarity tend to have matching labels, but precision tails off as recall increases.

For ConVote, unigrams are the most predictive at very low recall, with a peak of 79.70% precision at a recall of 0.38%. This performance tails off quickly, with all of the other lengths doing better after 2.5% recall. Trigrams appear to give the best sustained precision, closely followed by 4-grams.

For Bitterlemons there is a surprising trough for all term types except 5-grams between 0 and 5% recall, suggesting that there is a small core of very similar documents that do not have any label correlation. Otherwise the performance is similar to ConVote. 5-grams have a peak of 92.55% precision at just 0.20% recall and tail off immediately to be the worst performers. Bigrams appear to give the best sustained performance, closely followed by trigrams.

This brief experiment has demonstrated that, for our two test corpora, simple similarity
measures can be used to detect a good number of SAMELABEL relationships with accuracy that easily surpasses a random baseline. The next step is to try to measure how these links will perform in aggregate when used for document classification.

![Figure 5.5: Precision versus recall for ConVote relationship detection using cosine similarity.](image)

### 5.2.4 Preliminary Experiment 3: Similarity-based Local Classification

The first experiment in this section found an affirmative answer to the question: are two documents more likely to have the same label if they use the same rare word?

The second experiment in this section asked further: can this fact be used to reliably find pairs of documents that have the same label? The answer was positive again.

We now consider a third question: is similarity-based relationship detection sufficiently powerful to contribute to document classification performance?

The ultimate test of this question is in the application of similarity-based relational features to ConVote and Bitterlemons document classification using collective classifiers.
There is, however, a simpler way of determining whether or not similarity-based features can be expected to be useful. In this third experiment I will construct a local classifier that tests the performance of relational features without considering any content-only features.

A local classifier is a classifier that represents a collective classification problem from the point of view of a single instance. The labels of the other instances in the problem are taken as given and the challenge is to build a classifier that can take advantage of information about the target instance’s relationships to other instances.

Obviously, if neighbour labels are known and link prediction is accurate, classification becomes easy. If, however, link predictions are of varying quality, the classifier can only succeed if it maximises the impact of the helpful information with respect to the unhelpful. This experiment will determine whether or not the high precision relationship predictions on the left of the graphs in the previous experiment can be usefully extracted from, or combined with, the lower precision predictions towards the right.

I define the single feature type for the local classifier, which I term an “average similarity score”:

---

Figure 5.6: Precision versus recall for Bitterlemons relationship detection using cosine similarity.
Chapter 5: Collective Document Classification with Implicit Relationships

\[ f_s(i, l) = \frac{\sum_{d_j \in D \setminus \{d_i\}} s(i, j) \delta_{Y_j, l}}{\sum_{d_j \in D \setminus \{d_i\}} \delta_{Y_j, l}} \]

(5.3)

where \( \delta \) is the Kronecker delta. Put in words, the feature \( f_s \) is the average of the similarity scores for the pairings of the given instance with each of the instances that have the label \( l \). As noted above, a local classifier normally incorporates both content-based features and relational features. In this case I am attempting to measure the performance of the relational information only, so content-based features are not used.

To assign labels using this relational feature, the classifier simply finds the label that maximises the value of \( f_s(i, l) \):

\[ Y_i = \arg \max_l f_s(i, l) \]

(5.4)

If there is no single maximum (i.e. both labels give the same average similarity score) the classifier does not record a classification.

Results are shown in Tables 5.1 and 5.2 for ConVote and Bitterlemons respectively. For ConVote, accuracy is between 71.69\% for 1-grams and 75.77\% for 3-grams. The 1-gram and 2-gram classifiers produce results for all instances. For 3-grams, 4-grams and 5-grams there are, respectively, 3, 10, and 41 instances that cannot be classified. This decrease in recall performance with increasing lengths of \( n \)-grams is to be expected. The longer the \( n \)-gram that is used, the more likely it is that some instances will not have any features in common with other instances, meaning that the values for \( f_s(i, \text{For}) \) and \( f_s(i, \text{Against}) \) will both be zero.

Performance is somewhat better for Bitterlemons. The 3-gram classifier does the best with a score of 95.29\%, followed by 4-grams, 2-grams, 1-grams and 5-grams. There is a clear similarity in behaviour between the two corpora. In both cases 3-grams give the best performance and 1-grams and 5-grams are the two worst performers. The difference in absolute performance scores can be attributed to the fact that the Bitterlemons training instances consist of contributions by only two authors (the editors). The relationships between these instances are likely to be easier to predict because of features that are specific to each author. Recall from Chapter 3 that this is the main reason why the guest contributions are reserved for the test set.
Table 5.1: Local classifier performance for average neighbour similarity features on the ConVote Corpus.

<table>
<thead>
<tr>
<th>No. instances</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-grams</td>
<td>71.69</td>
</tr>
<tr>
<td>2-grams</td>
<td>75.40</td>
</tr>
<tr>
<td>3-grams</td>
<td>75.77</td>
</tr>
<tr>
<td>4-grams</td>
<td>74.54</td>
</tr>
<tr>
<td>5-grams</td>
<td>72.80</td>
</tr>
</tbody>
</table>

Table 5.2: Local classifier performance for average neighbour similarity features on the Bitterlemons Corpus.

<table>
<thead>
<tr>
<th>No. instances</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-grams</td>
<td>90.91</td>
</tr>
<tr>
<td>2-grams</td>
<td>92.93</td>
</tr>
<tr>
<td>3-grams</td>
<td>95.29</td>
</tr>
<tr>
<td>4-grams</td>
<td>93.60</td>
</tr>
<tr>
<td>5-grams</td>
<td>89.73</td>
</tr>
</tbody>
</table>

Unlike ConVote, Bitterlemons does not show diminishing recall as $n$-gram length is increased. This indicates there are all instance pairs have at least one 5-gram in common.

These results suggest two main conclusions:

1. Similarity-based relationship detection should be sufficiently powerful to contribute to document classification performance. Even given that fact that the document classifier will not have perfect information about neighbour labels available here, the high accuracies obtained for local classification should translate into useful relational features for use in collective classification.

2. Medium length $n$-grams (2, 3, or 4) are likely to give best performance. They appear to offer a helpful balance between being too short and therefore not predictive enough, and too long and therefore not common enough. That said, there is only a few percentage points separating the performance on all five of the $n$-gram lengths trialled.
Chapter 5: Collective Document Classification with Implicit Relationships

5.3 Dual Classifier Approach

In the last section I reported three preliminary experiments that show the usefulness of similarity-based features for predicting relationships between documents in the ConVote and Bitterlemons corpora. I now proceed to the objective that is the crucial part of this chapter: demonstrating that similarity-based relationships can be used in collective classifiers to improve on content-only classification. I begin by describing a dual classifier approach.

Recall from Section 2.4.7 that the dual classifier approach requires three steps:

1. **Base classification.** Produce base classifications using content-only and relationship classifiers. In this case, the content-only classifier will give the preference that each document be classified with one of the two labels For and Against (for ConVote), or Israeli or Palestinian (for Bitterlemons). The relationship classifier will indicate the preference that each document pair be SameLabel or DifferentLabel.

2. **Normalisation.** Normalise the scores, producing values for the classification preference functions, $\psi_i$, which can be input into a collective classification algorithm.

3. **Decoding.** Produce final classifications by optimally decoding the content-only and relationship level preferences using a collective classification algorithm.

5.3.1 Base classification

For content-only classification I use the same bag-of-words SVM with binary, unigram features as [Thomas et al.] introduced in Chapter 4. This classifier has been shown to be the best bag-of-words model for the Bitterlemons Corpus ([Beigman Klebanov et al. 2010]). The relationship classifier is the cosine similarity score defined above in Equation 5.2. There are two crucial differences between this classifier and the by-name reference classifier used in Chapter 4. First, this classifier produces classifications for all pairs, not just those pairs that have a by-name reference connecting them. Second, this classifier produces a similarity score (between 0 and 1) that does not indicate explicitly whether the two instances have a SameLabel or DifferentLabel relationship.
5.3.2 Normalisation

I use probabilistic SVM normalisation to convert the signed decision-place distance output by the content-only classifier into the probability that the instance is in the positive class (see Section 2.4.8).

The technique used to convert the cosine similarity score into a classification preference needs to fit complex criteria. Section 5.2.3 demonstrated that while the very highest similarity scores are good indicators of \textsc{sameLabel} relationships, classifier precision drops quickly as recall is increased. To avoid polluting the classification graph with large numbers of low quality links, the normalisation method should incorporate a threshold that discards a significant proportion of the test set pairs. The technique can also afford to be somewhat approximate because of the inexact nature of similarity-based relationship classification and the expectation that overall classifications will tend to be influenced more by high numbers of low confidence links working in concert than by single relationships.

I adopt the following binning technique to convert the cosine similarity score into a probability that the two instances are \textsc{sameLabel}:

$$
\psi_{ij}(l, l) = \begin{cases} 
0.9 & s(i, j) \geq b_1; \\
0.8 & b_2 \leq s(i, j) < b_1; \\
0.7 & b_3 \leq s(i, j) < b_2; \\
0.6 & b_4 \leq s(i, j) < b_3; \\
0.5 & s(i, j) < b_4; 
\end{cases} 
$$

(5.5)

where $\psi_{ij}(l, l)$ represents the \textsc{sameLabel} preference and the values for $b_1$, $b_2$, $b_3$, and $b_4$ are derived by sorting the relationships in the training data by similarity score and separating them into intervals holding a proportion of \textsc{sameLabel} pairs equivalent to the nominated probability. This approach is similar to unsupervised discretisation (Kotsiantis and Kanellopoulos 2006), except the intervals are arranged so that the output categories have a probabilistic interpretation.

Inspection of the graphs in Section 5.2.3 will show that the decision to limit the normalisation to 5 distinct probabilities means that the majority of relationships will be placed in the last interval and have their probability set to 0.5 (i.e. no information). This satisfies the requirement for a threshold that will remove the large body of unhelpful relationships. The
choice of a small set of probabilities also satisfies the intuition that using a larger number could involve reading too much in to what is essentially a heuristic measure of relationship.

5.3.3 Decoding

Decoding will be trialled using each of the three techniques used in Chapter 4:

1. minimum-cut (described in Section 2.4.6)
2. mean-field (described in Section 2.4.3)
3. loopy belief propagation (described in Section 2.4.3)

5.3.4 Tuning

The relative weights given to the content-only and relational classifiers can be tuned using the dampening method described in Section 4.5.1. This method introduces a dampening parameter in the range $[-1, 1]$ that serves to uniformly decrease the strength of either the content-only preferences (if the parameter is less than 0) or the relationship preferences (if the parameter is greater than 0).

For ConVote, the training fold is adapted for tuning by use of a 52-fold cross-validation approach, where each of the 52 debates in the training fold is classified using all of the other debates as training data. Bitterlemons does not have an internal structure within the training set, so it cannot be adapted in this way. Instead, I use leave-one-out, which trains one classifier for each instance in the training set. Unfortunately this approach is more computationally demanding. It also carries the risk of producing base classifications that are unrealistically accurate, because the training set is composed of articles by only two authors.

Figure 5.7 gives a diagrammatic overview of the dual classifier approach.
5.4 Iterative Classifier Approach

I now describe a second approach for collective classification using similarity-based information, based on iterative classification. Recall from Section 2.4.1 that the iterative classifier approach has three major components:

1. **Base classification.** Produce base classifications using a content-only classifier. As with the dual classifier approach, the content-only classifier will give the preference that each instance be classified with one of the two labels For or Against (for Con-Vote) or Israeli or Palestinian (for Bitterlemons). There is no relationship classifier.
2. **Addition of relational features.** Produce local vectors by adding relational features to the vectors previously used for content-only classification.

3. **Iterative re-classification.** Use a local classifier to classify the new feature vectors. Update the relational features after each iteration to reflect new class assignments. Repeat until class assignments stabilise or a threshold number of iterations is met.

### 5.4.1 Base Classification

Once again, content-only classification for the iterative classifier is performed using a bag-of-words SVM with binary, unigram features.

### 5.4.2 Relational Features

I derive relational features for the iterative classifier from the average similarity score, $f^a$, defined in equation 5.3.

$$f^{as}(i, l) = \begin{cases} 1 & f^a(i, l) > f^a(i, l'); \\ 0 & \text{otherwise;} \end{cases}$$  \hspace{1cm} (5.6)

Put in words, the feature $f^{as}(i, l)$ is set to 1 if and only if the average similarity of document $d_i$ to instances with label $l$ is greater than its average similarity to instances with label $l'$. In training, document labels are used when counting negative and positive instances to determine the values for $f^{as}$. In evaluation, the classes assigned in the previous iteration are used.

I have selected this approach because it appears to be the simplest way to express similarity information using relational features. As noted in Section 4.7.2, it is common for relational features to count the number of neighbours in each class. In this case a count is not helpful, because each instance is likely to be a neighbour to every other instance being classified.

The addition of relational features produces a relational feature tuple for ConVote in the form $(f^{as}(i, \text{For}), f^{as}(i, \text{Against}))$ and for Bitterlemons in the form $(f^{as}(i, \text{Israeli}), f^{as}(i, \text{Palestinian}))$.

Figure 5.8 gives a diagrammatic overview of the iterative classifier approach.
5.5 Experiments

In this section I assess the ability of the dual classifier and iterative classifier approaches I have described to perform collective document classification on the ConVote and Bitterlemons corpora.

In the tables and analysis that follow accuracies are reported as percentages of instances correctly classified. Experiments on ConVote are conducted using a 53-fold cross-validation configuration, where each of the 53 debates is classified in turn using the other 52 debates as training data. Micro-averaging is used, which means that the figures quoted refer to the overall proportion of instances that are correctly classified, rather than an average proportion across debates.

A fixed training/test split is used for evaluations on Bitterlemons.

When it is quoted, statistical significance has been determined using approximate randomisation with \( p < .05 \) (Nooreen 1989). See Section 4.8 for a description and justification.
for this approach.

Two baseline scores are shown in the tables for collective classification results. “Majority” gives the performance of the simplest possible classifier, which classifies every instance with the label that is most frequent in training data. “Content-only” gives the performance of the bag-of-words linear SVM used to perform base classification.

Unlike in Chapter 4, results are not broken down based on whether or not instances are “connected” or “isolated”. Chapter 4 was about maximising collective classification performance using an explicitly specified set of by-name references. This chapter assumes that all instances are implicitly connected to others and deals with the task of detecting as many relationships as necessary to maximise performance. The concepts of “connected” and “isolated” are irrelevant here; the only concern is overall performance.

5.5.1 Local Classifier Performance

Tables 5.3 and 5.4 show the performance of the similarity-based local classifier on ConV ote and Bitterlemons respectively. This test measures the ability of the classifier to predict the labels of speeches when it is given perfect information about the labels of their neighbours. These results represent a rough upper bound for the performance of the iterative classifier, which obtains neighbour labels iteratively.

For ConV ote, performance is best for 3-grams, which yield a score of 79.75%, a 3.35% improvement on the content-only baseline. The worst performance comes from 1-grams and 5-grams, with 2.71% and 1.70% gains respectively. As would be expected, this result mirrors the trend found when similarity-based features were evaluated in isolation in Section 5.2.4.

For Bitterlemons, 3-grams obtain the best accuracy with a score of 91.58%, a 5.05% improvement on the content-only baseline. 1-grams and 5-grams do worst with 3.36% and 3.03% gains respectively. Once again, this result mirrors the trend from Section 5.2.4.

Results in the next section will show whether or not these results translate into similar gains when neighbour labels are determined iteratively.
Table 5.3: Local classifier performance on the ConVote Corpus using similarity-based features. Neighbour labels are given by oracle.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Term length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>Content-only</td>
<td>76.40 76.40 76.40 76.40 76.40</td>
</tr>
<tr>
<td>Similarity-based local classifier</td>
<td>79.11 79.64 79.75 79.58 78.10</td>
</tr>
</tbody>
</table>

Table 5.4: Local classifier performance on the Bitterlemons Corpus using similarity-based features. Neighbour labels are given by oracle.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Term length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Majority</td>
<td>1 2 3 4 5</td>
</tr>
<tr>
<td>Content-only</td>
<td>86.53 86.53 86.53 86.53 86.53</td>
</tr>
<tr>
<td>Similarity-based local classifier</td>
<td>89.89 89.90 91.58 90.57 89.56</td>
</tr>
</tbody>
</table>

5.5.2 Collective Classifier Performance

Table 5.5 shows overall collective classifier performance on the ConVote Corpus. The best performer is the iterative classifier with 4-grams, with an accuracy of 79.05%. This is a statistically significant 2.65% gain on the content-only baseline. The iterative classifier is the best performer in general, obtaining the next four best results with statistically significant gains of 2.41%, 1.76%, 1.70% and 1.59% for 3-grams, 5-grams, 2-grams and 1-grams respectively.

The dual classifier with minimum-cut is the next best performer, with a best score of 77.45% for 5-grams, a statistically significant gain of 1.06%. 4-grams and 2-grams also provide statistically significant gains, but 3-grams and 1-grams do not.

For loopy-belief and mean-field the story is less positive. None of the variations gives a statistically significant improvement on the content-only baseline. The best performer is mean-field with 5-grams, with a score of 76.63, a 0.23% improvement on the baseline.

Table 5.6 shows overall collective classifier performance on the Bitterlemons Corpus. As with ConVote, the best performer is the iterative classifier. 4-grams and 3-grams are the top-performing variants, obtaining a score of 90.91%, a statistically significant 4.38% gain.
on the content-only baseline. 2-grams and 5-grams are the next best, with a statistically significant 3.37% gain on the content-only baseline. 1-grams are the only iterative classifier variant that does not yield a statistically significant improvement on the content-only baseline.

For both ConVote and Bitterlemons, the iterative classifier results are quite similar to the local classification scores discussed in the previous section. This indicates that the requirement to use a best guess for neighbour labels has not greatly reduced the performance of the similarity-based relational features.

The dual classifier results for Bitterlemons need special comment. As already mentioned (Section 5.3.4), leave-one-out tuning with the Bitterlemons training corpus is compromised. The aim of cross-validation on the training set is to gain a picture of likely performance on the test set. Unfortunately, the Bitterlemons Corpus is not homogeneous. Articles in each class in the training set are contributed by just one author; articles in the test set are contributed by different authors. Tuning on the Bitterlemons Corpus failed, because leave-one-out on the training set produced perfect, 100% performance, presumably because there are features specific to the two authors that make classification easy. This meant that the ideal dampening parameter was found to be exactly 1, i.e. collective classification was unnecessary, because the expected performance on the test set was 100%.

Strictly speaking, the dual classifier scores for Bitterlemons are all 86.53%, the content-only baseline. I have chosen instead to show the score for the optimal dampening parameter, found by inspection of the results. This is not a true representation of performance in the current content, but it serves to show the rough level of performance that could be expected if appropriate tuning data was available. The minimum-cut variant does best in this setting, with a top score of 90.57% for 5-grams, a statistically significant 4.40% improvement on the baseline. 4-grams, 3-grams and 2-grams also provide statistically significant improvements.

As with ConVote, none of the loopy belief or mean-field variants provide statistically significant improvements over the content-only baseline. The best performers are mean-field and loopy belief with 5-grams, with a score of 88.55%, a 2.02% improvement on the baseline.
Table 5.5: Collective classification performance on the ConVote Corpus. The dual classifier results are for dampeners tuned via cross-validation on the training fold. Results marked with ★ are statistically significant improvements over the content-only baseline. Statistical significance is determined using approximate randomisation with $p < 0.05$.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Majority</td>
<td>51.44</td>
<td>51.44</td>
<td>51.44</td>
<td>51.44</td>
<td>51.44</td>
</tr>
<tr>
<td>Baseline</td>
<td>Content-only</td>
<td>76.40</td>
<td>76.40</td>
<td>76.40</td>
<td>76.40</td>
<td>76.40</td>
</tr>
<tr>
<td>Dual</td>
<td>Cosine similarity, min-cut</td>
<td>75.22</td>
<td>77.22★</td>
<td>76.52</td>
<td>77.28★</td>
<td>77.46★</td>
</tr>
<tr>
<td>Dual</td>
<td>Cosine similarity, loopy belief</td>
<td>75.10</td>
<td>74.99</td>
<td>75.10</td>
<td>75.46</td>
<td>76.16</td>
</tr>
<tr>
<td>Dual</td>
<td>Cosine similarity, mean-field</td>
<td>75.10</td>
<td>74.99</td>
<td>75.10</td>
<td>75.46</td>
<td>76.63</td>
</tr>
<tr>
<td>Iterative</td>
<td>Average similarity score</td>
<td>77.99★</td>
<td>78.10★</td>
<td>78.81★</td>
<td>79.05★</td>
<td>78.16★</td>
</tr>
</tbody>
</table>

Table 5.6: Collective classification performance on the Bitterlemons Corpus. Because tuning is not possible on the Bitterlemons Corpus, the dual classifiers use an optimal tuning value found via inspection of the results. It is important to note that these values are not true representations of real-world performance. Results marked with ★ are statistically significant improvements over the content-only baseline. Statistical significance is determined using approximate randomisation with $p < 0.05$.

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Majority</td>
<td>49.83</td>
<td>49.83</td>
<td>49.83</td>
<td>49.83</td>
<td>49.83</td>
</tr>
<tr>
<td>Baseline</td>
<td>Content-only</td>
<td>86.53</td>
<td>86.53</td>
<td>86.53</td>
<td>86.53</td>
<td>86.53</td>
</tr>
<tr>
<td>Dual</td>
<td>Cosine similarity, min-cut</td>
<td>87.88</td>
<td>88.55★</td>
<td>88.89★</td>
<td>89.90★</td>
<td>90.57★</td>
</tr>
<tr>
<td>Dual</td>
<td>Cosine similarity, loopy belief</td>
<td>87.54</td>
<td>86.87</td>
<td>87.88</td>
<td>87.88</td>
<td>88.55</td>
</tr>
<tr>
<td>Dual</td>
<td>Cosine similarity, mean-field</td>
<td>87.54</td>
<td>86.87</td>
<td>87.88</td>
<td>87.88</td>
<td>88.55</td>
</tr>
<tr>
<td>Iterative</td>
<td>Average similarity score</td>
<td>87.54</td>
<td>89.90★</td>
<td>90.91★</td>
<td>90.91★</td>
<td>89.90★</td>
</tr>
</tbody>
</table>
5.5.3 Dual Classifier Dampening Response

I now proceed to an examination of the dampening response of the dual classifier methods. In this section I present six graphs showing the performance of the three different decoding algorithms on the two test corpora. This analysis is not crucial to evaluating the performance of the dual classifier approach, but it will help to establish a picture of its limitations in comparison with the iterative classifier approach.

Each of the graphs in this section shows the effect of a varying dampening factor on classification accuracy. In each graph only a small portion of the $[-1, 1]$ range supported by the dampening parameter is shown. The reason for this is visible on many of the graphs: performance is fixed at or near 50% until the dampening parameter is close to 1. This indicates that the probabilities of the content-only classifier and relationship classifier are badly mismatched; performance only becomes reasonable after the relationship preferences have been massively reduced in strength relative to the content-only preferences.

Figures 5.9, 5.10 and 5.11 show performance on Convote for minimum-cut, loopy belief, and mean-field respectively. The trend is the same in each, though not entirely visible because of the need to focus on the right hand side of the graph where the action is: performance is flat until a sudden jump up leading to steady improvement up to a peak shortly before the maximum dampening value of 1. At 1 the relationship preferences are entirely dampened and performance is the same as the content-only baseline.

For minimum-cut, 1-grams provide the highest peak accuracy with close to 78% at dampening factor 0.93. Each of the other $n$-grams jumps above the 76.40% baseline at close to this point, with 5-grams providing the most sustained period of high performance from dampening factor 0.85 through to almost 1.

Performance is worse for loopy belief and mean-field. Only 5-grams do better than the baseline, between approximately 0.92 and 0.95 dampening factor for both algorithms.

Figures 5.12, 5.13 and 5.14 show performance on Bitterlemons for minimum-cut, loopy belief, and mean-field respectively. The trend is the same: after a period of flat performance, scores steadily improve as the dampening factor is increased, reaching a peak shortly before the maximum dampening value of 1.

For minimum-cut, 5-grams give the best performance with a peak of 90.57% accuracy
at dampening factor 0.95. 4-grams do the next best followed by 3-grams, 2-grams and 1-grams. Each algorithm rises to a sudden peak and then trails off as it approaches maximum dampening. Loopy belief and mean-field give almost identical performance. Both show the same peak-and-trail-off shape as with minimum-cut but the performance gain is smaller, with 5-grams obtaining a best score of 88.55%.

![Figure 5.9: Dual classifier performance against dampening factor for the ConVote Corpus with similarity-based links and the minimum-cut algorithm. Scores are shown for the n-gram lengths 1, 2, 3, 4 and 5. The baseline is for content-only classification.](image)

### 5.6 Relationship to Prior Work

Section 2.3.3 describes the previous work that deals with the question of collective document classification using implicit inter-document relationships. Two types of relationship are described: (1) proximity-based links, which make use of a spatial or temporal dimension in the document corpus to derive useful relationships; and (2) similarity-based links, which make use of measures of similarity.

Most of the prior work in similarity-based links has been in the field of transductive semi-supervised classification. That task begins with the premise that only a small amount of labelled training data is available, so content-only classification is likely to be inaccurate.
By contrast, the supervised techniques in this chapter deal with large amounts of labelled training data and relatively high content-only performance – 76% for ConVote and 87% for Bitterlemons. It is reasonable to assume that the types of similarity-based relationships derived for transductive semi-supervised classification would be ineffective in a supervised context.

This conclusion is supported by an experiment that shows that the vocabularies of document pairs tend to overlap to similar degrees regardless of document class (Pang and Lee 2005).

The only true prior example of supervised document classification using similarity-based inter-document relationships is Agarwal and Bhattacharyya (2005). In this work, the task is to perform binary sentiment classification on a set of movie reviews. The similarity measure used to derive inter-document relationships is a simple measure of term-overlap formulated as follows:

\[ s^a(i, j) = \frac{\vec{x}_i \cdot \vec{x}_j - a_{min}}{a_{max} - a_{min}} \]  

(5.7)

where \( \vec{x} \) is a binary feature representation, \( a_{max} \) is the largest vector dot product for any

![Figure 5.10: Dual classifier performance against dampening factor for the ConVote Corpus with similarity-based links and the loopy belief algorithm. Scores are shown for the n-gram lengths 1, 2, 3, 4 and 5. The baseline is for content-only classification.](image-url)
Chapter 5: Collective Document Classification with Implicit Relationships

Figure 5.11: Dual classifier performance against dampening factor for the ConVote Corpus with similarity-based links and the mean-field algorithm. Scores are shown for the \( n \)-gram lengths 1, 2, 3, 4 and 5. The baseline is for content-only classification.

Figure 5.12: Dual classifier performance against dampening factor for the Bitterlemons Corpus with similarity-based links and the minimum-cut algorithm. Scores are shown for the \( n \)-gram lengths 1, 2, 3, 4 and 5. The baseline is for content-only classification.
Figure 5.13: Dual classifier performance against dampening factor for the Bitterlemons Corpus with similarity-based links and the loopy belief algorithm. Scores are shown for the $n$-gram lengths 1, 2, 3, 4 and 5. The baseline is for content-only classification.

Figure 5.14: Dual classifier performance against dampening factor for the Bitterlemons Corpus with similarity-based links and the mean-field algorithm. Scores are shown for the $n$-gram lengths 1, 2, 3, 4 and 5. The baseline is for content-only classification.
document pair and $a_{\text{min}}$ is the smallest.

Content-only classification is performed using an SVM and a bag-of-words feature model. The feature weights for adjectives are set using information from the Wordnet (Fellbaum 1998) synonymy graph. All other feature weights are binary.

Agarwal and Bhattacharyya convert the outputs of the content-only classifier to probabilities using probabilistic SVM normalisation and decode using minimum-cut. The graph is fully connected, with each pair of documents connected with the weight $s^a(i, j)$.

Results are outstanding. The content-only score of 75.8% is increased by 19.8% yielding a score of 95.6%.

Considered at surface value, this is an example of prior work that is both simpler and better performing than the approaches presented in this chapter. In an attempt to verify this experimentally I reproduce the Agarwal and Bhattacharyya experiment. I use the same corpus, 5-fold cross-validation, binary bag-of-words feature model, SVM content-only classifier, probabilistic SVM normalisation, and term-overlap based inter-document relationships. I omit the Wordnet-based feature weighting for adjective and use binary feature instead.

Results are shown in Table 5.7. The first noticeable difference is with the content-only score, which at 86.30% is far better than Agarwal and Bhattacharyya equivalent of 75.8%. My higher score is more consistent with the score of score of 87.15% obtained by Pang and Lee (2004) using a 10-fold cross-validation configuration.

One explanation for this disparity could be that the scheme used for weighting adjective features has had a strong negative impact on performance. This seems unlikely, given that the authors would surely have tried their experiment in the simpler configuration and not bothered to report their results if performance was not improved by their approach.

A more likely explanation is that Agarwal and Bhattacharyya arranged their folds such that the positive and negative classes were unevenly represented, i.e. without stratification. The lower performance of the Agarwal and Bhattacharyya method is consistent with this scenario (Wu and Chang 2005).

classifier that has been trained and tested on datasets with differing class distributions. Unfortunately the paper does not give any insight into the method used for assigning reviews to folds.
The second and much more important disparity between my reproduction and the original is collective classifier performance. Instead of increasing performance by 19.8%, my implementation reduces performance by 8.20%, to 78.10%. Clearly one of the two results is misleading. The simplest explanation is that the higher performance of the original experiment is again due to an unbalanced class distribution and that the method does not have the potential to provide reliable performance gains.

Several additional observations can be made in support of this view. First, it seems highly unlikely that useful relationships could be derived from any measure of similarity derived from exactly the same feature model as that used for content-only classification. The SVM content-only classifier will tend to assign similar documents to the same class without any help from inter-document relationships. Relationships can only become useful when they introduce some new information about similarity that was not already apparent in the content-only feature data. This is the case with the n-gram model used in this chapter because n-grams longer than 1 provide information that is not available in a bag-of-words model. Moreover, the n-gram makes use of information not available to the content-only classifier by finding overlaps based on features that may not have been seen in training data. The Agarwal and Bhattacharyya method has no such property.

The second observation in support of the view that the performance gains shown by Agarwal and Bhattacharyya were the result of an unbalanced class distribution comes from earlier in this chapter. The results in Section 5.5.3 show that with insufficient dampening, each of the dual classifier methods exhibits a strong bias in favour of one class. The graphs show performance at its worst when all of the documents are placed in the one class (performance of 50%). As the dampening factor is increased, performance increases as the label imbalance is corrected. As I note in Section 2.5.1 this phenomenon has also been observed by others, and tends to occur with highly connected link graphs.

The Agarwal and Bhattacharyya approach uses a fully connected link graph. The gains they obtain for their collective technique are consistent with the effect of this phenomenon taking place in a dataset with a high class imbalance.

The final point in favour of this view comes from history. Of all of the relevant works that cite the Agarwal and Bhattacharyya paper, none have attempted to reproduce the experiment or apply it to other tasks (Thomas et al. 2006; Pang and Lee 2008; Bansal et al. 2008).
<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Majority</td>
<td>50.00</td>
</tr>
<tr>
<td>Baseline</td>
<td>Content-only</td>
<td>86.30</td>
</tr>
<tr>
<td>Dual</td>
<td>Term-overlap relationships, min-cut</td>
<td>78.10</td>
</tr>
</tbody>
</table>

Table 5.7: Collective classification performance on the Movies Corpus using the Agarwal and Bhattacharyya method.

Su and Markert 2009; Tan et al. 2011). This is despite the unprecedented performance levels reported.

To sum up, my work in this chapter is unique amongst its forebears in the field of similarity-based collective classification because of its applicability to supervised document classification and its use of features not observed in training data. The closest prior work is that of Agarwal and Bhattacharyya (2005). Contrary to the outstanding results presented in that paper, I have shown experimental, theoretical and historical evidence that the techniques in this chapter are both more powerful and more generally applicable.

5.7 Conclusions

The key conclusion of this chapter is that measures of similarity can be used to produce collective classifiers that give better performance than a bag of words baseline. Both iterative classifier and dual classifier models can be effective, depending on the characteristics of the classification task. This is a new result and should be considered relevant and significant to any document classification task with potential to be tackled collectively.

My hypothesis in Chapter 3 was that the ConVote and Bitterlemons corpora would be amenable to relationship detection because of their “dominance of task-relevant content”. In other words, a large percentage of the words and phrases in both corpora appear to be specific in some way or other to the class of the document. This is in contrast with corpora in which only a small portion of the document may be applicable to the document class, while the remainder is orthogonal to the dimension of classification. To prove this hypothesis I would have needed to consider a much broader array of corpora, including negative examples, and develop a formal definition and measure for “task-relevant content”.
Nonetheless, the evidence collected in this chapter is consistent with the hypothesis. Beyond these two general conclusions, I offer the following practical findings, directly derived from the experiments presented in this chapter.

**Iterative classifiers are superior to dual classifiers**

The experimental results are clear: iterative classifiers do better than dual classifiers at collective classification using similarity-based relationships. Their superiority goes beyond measures of performance. Iterative classifiers are simpler to implement, with no need for a complex conversion from similarity score to probability, and more efficient, with no parameters to tune. The key advantage of the iterative classifier seems to lie in its ability to sum up relationship information in a single average similarity score. This allows the classifier to neatly learn the correct weights for just two relational features. In contrast the dual classifier approaches need to go through the complexity of deriving a dampening parameter and expressing each similarity score as an arc in the graph. Section 5.7.1 has a discussion of how this latter issue may cause problems. Crucially, because the performance of the dual classifier methods is far worse than the baseline when tuning is not applied, iterative classifiers are the only choice for tasks like Bitterlemons that do not have training data that is suitable for tuning by cross-validation.

**Minimum-cut is the first choice dual classifier decoding algorithm**

Dual classifiers using minimum-cut were the only ones that provided statistically significant improvements over the content-only baseline. This is a somewhat surprising result, given the similar performance of minimum-cut, loopy belief, and mean-field in Chapter 4. The poor performance of loopy belief propagation in this setting is consistent with earlier observations (see Section 2.5.1), but it is not clear why minimum-cut does a little better and mean-field does not. The variation in performance may be attributable to the different graphical models used by the three methods. Loopy belief and mean-field are Markov random field approximations and minimum-cut is based on a flow model. It may be that the flow model performs better when faced with the highly connected graphs and very low link probabilities specific to similarity-based collective classification. Further research may answer this question.
3-grams and 4-grams are the most effective overlap features

Results throughout the chapter have shown that, of the five variants trialled, 3-grams and 4-grams most often give the best performance. This includes results for relationship detection (Section 5.2.3), similarity-based local classification (Section 5.2.4) and overall collective classification (Section 5.5.2). 3-grams and 4-grams appear to provide the ideal balance between being too short, and therefore not predictive enough, and too long, and therefore not overlapping frequently enough.

5.7.1 A Theoretical Difficulty with Graph-based Methods

Counter to expectations created in the previous chapter, work in this chapter has shown that iterative classifiers are significantly better than dual classifiers for similarity-based collective classification. I believe that this is due to shortcomings in the graph-based methods used by the dual classifiers to resolve content-only and relationship preferences. In particular, it appears that graph-based methods are not well-suited to handling the very frequent, low quality links produced by similarity-based methods. This is in contrast to the less frequent, more reliable links produced by the by-name references used in the last chapter.

I offer the following scenario as an example of this problem:

1. Let us define a collective classification problem in which there are 10 positive instances and 4 negative instances.

2. Assume that all 10 of the positive instances are correctly and confidently identified by the content-only classifier.

3. Assume further that 3 of the 4 negative instances are correctly and confidently identified by the content-only classifier. The fourth is given a marginal classification so it is not clear whether it is a positive or negative. Relationship preferences will be the determining factor.

4. Assume that a single similarity-based feature is used to determine that the marginally classified instance is equally related to 4 of the 10 positive instances and all 3 of the 3 negative instances.
Intuitively, the marginal instance should be classified as negative because it is related to all of the negative examples and only a minority of the positive examples. Graph-based decoders will classify the marginal instance as positive, because the cumulative effect of the 4 links to confidently classified positive instances is greater than the 3 links to confidently classified negative instances. In contrast, the average similarity score used for the iterative classifiers in this chapter will correctly interpret the relational information as indicating a preference for the negative class. Figure 5.15 gives a diagrammatic view of this scenario.

I posit that for the experiments in this chapter, differences in class distribution for content-only preferences have confounded the graph-based models. This conclusion is supported by the experiments in Section 5.5.3 which show extremely poor performance when near-maximum dampening is not applied. Inspection of the outputs shows that these low scores occur when all instances are assigned the same class. The graphical structure created by the frequent, semi-reliable links output by the similarity-based relationship classifier appears to have an imbalanced effect, where all instances are pulled into the class that has the highest aggregate content-only preferences.

5.8 Future Work

The work in this chapter represents a beginning on what could be a fruitful thread of research. Possibly the most crucial question suggested by this work is: what is the range of tasks on which the techniques could be expected to be effective? In the previous section I described how my hypothesis about task-relevant content could be further tested by running the experiments in this chapter on a range of other tasks. In Section 3.1.1 I flagged movie review sentiment classification as an existing task that my hypothesis suggests would not be amenable to similarity-based collective classification. This corpus could be a first option, and would require some data manipulation to produce a suitable topic-based structure. There is a very large range of other tasks that could be tried. One relatively new task that has some interesting properties is Twitter user geolocation (e.g. Eisenstein et al. 2010; McGee et al. 2011; Wing and Baldridge 2011).

The second most important question raised by this work is the extent to which the
Figure 5.15: An example of incorrect handling of similarity-based links by graph-based models. The marginally classified instance in the middle is connected to four of 10 confidently predicted positives and three of three confidently predicted negatives via a single feature. Intuitively it should be classified as a negative, because it is related to all of the negative examples and a minority of the positive examples. It will be classified as a positive because of the greater number of links to positive instances. Note that connections between confidently predicted instances have been omitted because they have no impact on final classifications.

techniques can be expected to apply usefully in combination with state-of-the-art feature models. Improving on bag-of-words is one thing, but these techniques will be much more useful if they are able to provide improvements that go beyond the state-of-the-art. In theory, it is reasonable to suggest that combining similarity-based collective classification with other non-collective techniques would work, because the methods would be orthogonal. An obvious place to start testing this would be on the Bitterlemons Corpus, where a range of techniques have been applied ([Lin et al. 2006], [Greene 2007], [Greene and Resnik 2009], [Paul and Girju 2010]). There are also several works which have established benchmarks on ConVote using more advanced non-collective approaches ([Greene 2007], [Sokolova and Lapalme 2008], [Balahur et al. 2009], [Balahur et al. 2009], [Martineau and Finin 2009], [Martineau...])
et al. 2009; Yessenalina et al. 2010b).

It would also be possible to apply state-of-the-art approaches to sentiment classification (e.g. Taboada et al. 2011; AR et al. 2011; Yessenalina et al. 2010a) to both ConVote and Bitterlemons and combine these with the techniques from this chapter.

There is room for a good deal of analysis and experimentation to fully answer the question of why iterative classifiers perform better than dual classifiers. There is also scope for work to more fully explore why the minimum-cut decoding algorithm does better than mean-field and loopy belief. This work could follow the trend set in prior research by adopting a method for generating synthetic collective classification problems with different structures (see Section 2.5.1). There is also a need for more detailed theoretical analysis to explain differences in performance.

Finally, just as bag-of-words models are only a basic starting point for content-only classification, it seems likely that n-gram features could be enhanced or superseded by more nuanced feature models for similarity-based relationship detection. For example, paraphrase detection (e.g. Malakasiotis 2009; Das and Smith 2009) is an existing area of research that could be used to provide similarity-based relationship detection using whole sentences. There is significant scope for future research in this area.
Chapter 6

Conclusions and Future Work

This thesis has been about document classification, which is the process of automatically assigning documents to categories based on their content. Document classification is a foundational task within the science of natural language processing because of its simplicity, inherent usefulness and applicability as a component of more complex systems.

Normal document classifiers are content-only, i.e. they do their classification by considering the contents of each document in isolation. Collective document classifiers, the type dealt with in this thesis, are designed to take into account the relationships between documents in a collection in order to achieve better results. One of the main tasks dealt with in this thesis was predicting the vote of speakers in U.S. congress based on their contribution to a floor-debate. The ConVote corpus consists of the transcribed verbal contributions of members of congress to floor-debates on different bills. These transcriptions are matched with voting records, so that each contribution of a speaker to a debate is labelled as For or Against.

A content-only approach trained using bag-of-words features does somewhat well at this task. A significant improvement is obtained by adopting a collective approach that takes advantage of the fact that speakers give clues about their relative votes in the language they use when referring to each other by name.

The second task dealt with in this thesis was detecting the perspective of contributors to a web publication dealing with the Israel-Palestine conflict. Each issue of the publication consists of two articles written from the Palestinian perspective and two from the Israeli
perspective. The job of the classifier is to assign an Israeli or Palestinian label to articles based on their content. There are no explicit inter-document relationships in this corpus, so collective classification needs to include a process for deriving useful implicit inter-document relationships from the document contents.

This emphasis on collective approaches is important not just because of the possibility of increased performance, but because of the current prevalence of linked document collections. In particular, the web is a growing source of documents that are linked to each other via hyperlinks and social networking relationships. It should be possible to make good use of these relationships using collective classifiers.

This thesis has provided a novel and significant contribution to the body of knowledge about collective document classification. I aimed to distinguish approaches that applied explicit inter-document relationships, like the by-name references in congressional debate transcripts, from approaches that applied implicit relationships. I defined implicit relationships as those that are not directly embodied by any specific document content, but which can be inferred indirectly.

In Section 1.1 I described seven different high-level goals for this thesis and how they were to be met. I will now discuss the conclusions of this thesis in terms of those seven goals.

**Provide a review of collective document classification techniques and applications**

In Chapter 2 I provided a detailed overview of existing collective document classification techniques and applications. I surveyed the existing collective classifiers based on explicit links and implicit links. Four major types of explicit links were discussed: (1) links in hypertext documents; (2) citations in academic papers; (3) links in social publishing sites; and (4) by-name references. There were also two major types of implicit links: (1) links based on measure of proximity, where some physical or temporal relationship is informative; and (2) links based on measures of similarity.

In Chapter 2 I described the three major approaches to designing collective classifiers that can make use of these inter-document relationships. Iterative classifiers work by adding relational features to standard classifiers and iteratively updating the values
of these features. Global formulations use approximations of Markov random fields to produce a probability distribution over labels using a graphical model. Dual classifiers use separate content-only classifiers and relationship classifiers and combine these preferences using a decoding algorithm.

The main goal of Chapter 2 was to introduce the models and related work necessary for a reader to understand the remainder of the thesis in context. It also offered two novel concepts that are useful for understanding the literature: (1) the distinction between explicit and implicit inter-document relationships provides a useful way of thinking about the different types of links used in collective document classification; and (2) the dual classifier concept provides a convenient description of family of collective classifiers found in the collective document classification literature.

**Provide a new experimental comparison of collective techniques**

Chapter 4 provided a comparison of collective techniques on the ConVote corpus. It found the best performance was achieved with a dual classifier using probabilistic SVM normalisation, which is a technique for converting the output of an SVM classifier to a probability (see Section 2.4.8). There was little to separate the variants of this classifier that use mean field, loopy belief propagation and minimum-cut algorithms for decoding. The best overall performer was the tuned version of mean-field (see Section 4.8).

The next best performers were the dual classifiers with variance-based normalisation and minimum-cut decoding. Variance-based normalisation is an alternative method of normalising the output of an SVM so it can be input into a decoding method (see Section 4.3.2). Close behind these come the global formulation based on loopy belief propagation and iterative classifiers.

The simple conclusions from this comparison are that dual classifiers are the best choice for the task and that probabilistic SVM normalisation is an effective way of converting the outputs of an SVM for use in a dual classifier.

There are two special cases where mean field and loopy belief propagation would be considered before minimum-cut as dual classifier decoding algorithms:
1. There are a significant number of DifferentLabel inter-document relationships. The standard minimum-cut algorithm has no computationally tractable solution for collective classification with these relationships.

2. A measure of classifier confidence is required. The standard minimum-cut algorithms outputs a binary classification only, without any measure of classifier confidence.

**Show better than state-of-the-art performance**

In Chapter 4, my dual classifier methods based on SVMs and probabilistic SVM normalisation do better than the previous benchmarks set by Thomas et al. (2006) and Stoyanov and Eisner (2012). This result carries the caveat that a 53-fold cross-validation configuration is used instead of a fixed train/test split. As discussed in Section 3.2.3, I believe this setting provides a more reliable estimate of likely real-world performance.

In Chapter 5, the similarity-based method used for collective classification is the first of its kind, and as such represents the state-of-the-art. This is particularly important for Bitterlemons, which has no explicit inter-document relationships and has never before been collectively classified.

**Provide a balanced evaluation of feature models for representing explicit links**

I trialled three feature models for representing explicit links in iterative classifiers: count features, binary context window features and frequency count context window features.

A count feature is a count of the number of neighbours that have a given label. A context window feature corresponds to the use of a given term in the context of a by-name reference to or from a neighbour of a given class. The frequency count variant is set to the count of the number of uses of the token in context windows relating the instance to neighbours with the given label. The binary variant is set to 1 in any case where the frequency count equivalent feature would be non-zero.

The experimental results showed that frequency count context window features were the best choice, followed by binary context window features then count features.
Develop a new similarity-based method for implicit link construction

In Chapter 5, I developed the concept of similarity-based inter-document relationship detection. This approach used a similarity measure between tf.idf weighted feature representations to find pairs of documents that were more likely to carry the same label. I used $n$-gram features instead of bag-of-words features so that the similarity measure would be based on features that had not been seen by the content-only classifier.

I developed two methods for incorporating this measure into a collective classifier. The dual classifier approach was based on a customised normalisation technique that converted the similarity score into a probability to allow input into a standard decoding algorithm. The iterative classifier used an average similarity score aggregation feature. This feature was calculated as the average similarity between the target document and all other documents with the given label.

Experimental results showed that both of these approaches performed better than random baselines and yielded collective performance gains.

I conclude that similarity-based inter-document relationship detection is an effective technique for implicit link construction.

Conduct an experimental comparison of collective techniques that use implicit links

Chapter 5 provided a comparison of collective techniques on the ConVote and Bitterlemons corpora using only implicit inter-document relationships. For Bitterlemons, this was the first use of the corpus for collective classification. The results showed iterative classifiers to be better performers than dual classifiers, a reversal of the trend shown for explicit links. This result is in keeping with results from earlier works where graphical methods showed sharply diminished performance with high numbers of inter-document relationships (see Section 2.5.1). It also fits with a theoretical justification for the superiority of iterative classifiers in scenarios where inter-document relationships are derived from measures of similarity (see Section 5.7.1).

I conclude that iterative classifiers are likely to be the superior choice for collective classification scenarios where link density is high, such as when deriving inter-
Chapter 6: Conclusions and Future Work

document relationships using measures of similarity.

Amongst the dual classifier approaches, only those that used minimum-cut decoding provided any performance gains. I conclude that if it is necessary to use a dual classifier instead of an iterative classifier, minimum-cut should be the first choice decoding algorithm.

I now offer two general conclusions that do not fit into any of the goal-specific categories discussed so far:

**Collective document classification is a powerful and broadly applicable approach**

This thesis has provided evidence towards the conclusion that collective document classification works well and can be applied in a variety of situations. On both the ConVote and Bitterlemons tasks, performance was significantly better than the content-only baseline. In the case of ConVote the absolute performance gain over connected instances using by-name references was about 8%. Using only implicit relationships, the gain was roughly 3%. Using derived relationships with Bitterlemons yielded a gain of roughly 4%. It is crucial to note that these gains are achieved in the context of relatively high content-only scores. For Bitterlemons in particular, a 4% improvement on an 89.53% baseline represents a large relative reduction in error-rate. Collective techniques are designed to be applied in combination with state-of-the-art content-only techniques and provide numerically small yet significant performance improvements.

The good results on the Bitterlemons task, combined with previously established results on ConVote and other types of corpus (hypertext documents, academic papers, newsgroup posts, tweets, etc.) suggest that collective techniques are broadly applicable. With an effective and general technique for deriving similarity-based links, the range of possible applications becomes very much larger. My hypothesis that good performance with this approach depends on an abundance of task-relevant content (see Section 3.1.1) is supported by the experimental results.

**Good collective performance depends on choosing the right approach**

The relative performance of the two main collective classification approaches trialled
in this thesis varied with the task being attempted. Dual classifiers were much better than iterative classifiers at taking advantage of the explicit inter-document relationships in the ConVote corpus. Relative performance was reversed when using implicit inter-document relationships, with iterative classifiers doing much better and some of the dual classifier techniques actually decreasing performance. Clearly, the performance of a collective classification technique depends on its fitness for the task at hand.

Future Work

Future work should consider the combination of the methods investigated in this thesis with more advanced content-only approaches. For both Chapters 4 and 5, this involves looking at sentiment classification techniques that have been shown to be effective on the ConVote corpus and on similar tasks. For the implicit links task with the Bitterlemons corpus, this involves looking at non-collective sentiment and perspective classification techniques that have been shown to work for Bitterlemons and similar tasks. See Sections 4.10 and 5.8 for details for ConVote and Bitterlemons respectively.

Another worthwhile experiment in combining methods would be to attempt ConVote classification using both similarity-based and by-name references. A positive result here would represent best ever classification performance on the ConVote corpus and demonstrate another scenario in which similarity-based links are useful.

A second major category of potential future work relates to collective classification algorithms in general. For dual classifiers and iterative classifiers, it would be interesting to explore the question of whether or not alternative base classifiers can provide better performance. Confidence-weighted linear classification is an example of a modern technique that has been shown to be highly effective on non-collective document classification tasks and could be easily adapted for use in a dual classifier or iterative classifier (Dredze et al. 2008).

Similarly, there is a need to more fully investigate the properties and performance of different feature models for representing inter-document relationships. In Chapter 4 I
showed that frequency count context window features worked best for representing the by-name references in the ConVote corpus in iterative classifiers. In Chapter 5 I introduced similarity-based inter-document relationship detection as an effective approach for iterative classification with implicit links. Future work could compare these approaches to a wider range of alternatives and establish their performance on different kinds of collective classification task.

Feature models for representing relationships in dual classifiers and global classifiers can also be investigated further. As mentioned in Section 4.9, there are a range of modern approaches that could be applied to ConVote to gain better performance than the binary bag-of-words features used Chapter 4. It would also be useful to consider classification of explicit inter-document links as a general task and to compare techniques across the full range of explicit inter-document relationships, including hyperlinks, citations and social network relationships.

A third major category for future work is trialling my techniques on other tasks. There are a considerable number of corpora that already have established benchmarks (see Section 2.3.2 and Section 2.3.3). There is also scope for producing new corpora from readily available digital content. In the case of Chapter 5 there is a strong incentive to trial the techniques on a range of additional tasks, to establish how broadly they are applicable. My hypothesis (see Section 3.1.1) is that the techniques will be effective on tasks with large proportions of task-relevant content, i.e. content that carries useful information about the dimension of classification, as opposed to content that is orthogonal to that dimension. There are two steps involved in developing this hypothesis: (1) establishing a formal measure of task-relevant content; and (2) measuring the performance of the techniques on corpora with different levels of task-relevant content and looking for a correlation.

Related to this is the work of establishing guidelines for when using a collective classifier is likely to be a better choice than alternative means of increasing performance. In some scenarios, it may be that manually labelling a little more training data might be an easier path to a performance improvement than implementing a collective technique. Similarly, there may be thresholds for low and high content-only classification performance beyond which no collective gains are likely. There is scope for future work to answer these practical questions.
A final major category for future work is developing complete explanations for the relative performance of the different approaches trialled in this thesis. There should be a theoretical explanation for why dual classifier performance is poor when link density is high. There should also an explanation for why minimum-cut is the only decoding technique that shows performance gains for a similarity-based dual classifier. I provide some analysis in Section 5.7.1 but more is possible.

One additional possibility for future work is establishing a clear best practice for experimental configuration. For my experiments with ConVote I adopted a 53-fold cross-validation approach that I judged would provide the nearest equivalent to real-world performance. This was different to the approach taken by prior work (Thomas et al. 2006; Bansal et al. 2008; Stoyanov and Eisner 2012), and led to different conclusions about the relative performance of different methods. For example, I found that the Bansal et al. (2008) approaches for representing disagreement relationships with minimum-cut were ineffective, contrary to the findings reported by the original authors. There is a need to establish a theoretical basis for judging the extent to which a collective classification experiment provides a valid estimate of real-world performance. Given that researchers have demonstrated that collective classification performance can exhibit high dependence on small factors in the test data (Bilgic and Getoor 2008; McDowell et al. 2009), a starting point would be to build some much larger corpora and compare results between different sizes of training set and test set, with K-fold cross-validation and without.

Beyond all of this, there is one more ambitious and longer term possibility for future work that involves improving on the current set of collective classification algorithms. Global approaches based on Markov random fields are the most conceptually sound of the three main approaches. They embody collective problems using a self-consistent and complete mathematical representation. This is in comparison with dual classifiers and iterative classifiers, which are essentially procedural approaches for layering collective behaviour on top of non-collective algorithms. Unfortunately global approaches suffer from slow learning and weaker performance, as shown in Chapter 4. Dual classifiers and iterative classifiers are faster to train and give better classification performance, but they have relative strengths and weaknesses. Dual classifiers have shown themselves to be superior to iterative classifiers for ConVote classification with explicit links, but inferior for ConVote
and Bitterlemons classification with implicit links.

What is desired is a new technique that combines theoretical elegance with explicit link performance equivalent to dual classifiers and implicit link performance equivalent to iterative classifiers. This is a challenging prospect for future research.
Bibliography


GOLDBERG, ANDREW, and XIAOJIN ZHU. 2006. Seeing stars when there aren’t many stars: Graph-based semi-supervised learning for sentiment categorization. In Proceedings of TextGraphs: the 1st Workshop on Graph Based Methods for Natural Language Processing, 45–52, New York, USA.


TAN, PANG-NING, MICHAEL STEINBACH, and VIPIN KUMAR. 2006. Introduction to Data Mining. Boston, USA: Addison Wesley.


Author/s:
Burford, Clinton

Title:
Collective document classification using explicit and implicit inter-document relationships

Date:
2013

Citation:

Persistent Link:
http://hdl.handle.net/11343/38110

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.