FEATURE CONSTRUCTION, SELECTION AND CONSOLIDATION FOR KNOWLEDGE DISCOVERY

by

Jiexun Li

A Dissertation Submitted to the Faculty of the
COMMITTEE ON BUSINESS ADMINISTRATION

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN MANAGEMENT

In the Graduate College

THE UNIVERSITY OF ARIZONA

2007
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Jiexun Li entitled Feature Construction, Selection and Consolidation for Knowledge Discovery and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 05/01/2007
Hsinchun Chen

Date: 05/01/2007
Jay F. Nunamaker, Jr.

Date: 05/01/2007
Zhu Zhang

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Date: 05/01/2007
Dissertation Director: Hsinchun Chen
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Jiexun Li
ACKNOWLEDGMENTS

I gratefully thank my advisor, Dr. Hsinchun Chen, for his guidance and encouragement throughout my five-year doctoral study at the University of Arizona. It has been an invaluable opportunity for me to work at the Artificial Intelligence Lab under his direction. I believe that what I have learned from him will significantly benefit my future career. Special thanks to my dissertation committee members, Dr. Jay F. Nunamaker, Jr., Dr. Zhu Zhang, and Dr. J. Leon Zhao, and my minor committee member in the School of Information Sources and Library Science, Dr. Martin Frické, for their guidance and encouragement. I also thank the department chair, Dr. Mohan Tanniru, and all other faculty members for their support. A special thank note goes to my academic advisor at Tsinghua University, Dr. Guoqing Chen, for his advice and encouragement.

My dissertation has been partly supported by National Institute of Health (NIH)/National Library of Medicine (NLM), National Science Foundation and Central Intelligence Agency. My work at the Artificial Intelligence Lab has been supported by many colleagues. I thank Dr. Jesse Martinez and Bernard W. Futscher in the Arizona Cancer Center, Dr. David W. Galbraith in the Department of Plant Science, Det. Tim Petersen in the Tucson Police Department, and Cathy Larson, Homa Atabakhsh, and Kira Joslin in the AI Lab. I also want to thank Ms. Sarah Marshall and Ms. Carol Schumaker for editing my papers.

I would like to thank my good colleagues and friends, Zan Huang, Hua Su, Rong Zheng, Michael Chau, Jennifer Xu, G. Alan Wang, Xin Li, Siddharth Kaza, Nichalin Suakkaphong, Rob Schumaker, Wei Chang, Ahmed Abbasi, Daning Hu, Mick Lin, Hsin-min Lu, Chun-Ju Tseng, Tianjun Fu, J. Harry Wang, Ming Lin, Yiwen Zhang, Gondy Leroy, Daniel McDonald, and Byron Marshall for their support to my studies.

Most of all, I am grateful for the constant support, understanding, patience, and trust of my wife, Hanmei Chen, my parents, and the rest of my family, without whom none of this work would have been possible.
DEDICATION

This dissertation is dedicated to my family.
# TABLE OF CONTENTS

LIST OF ILLUSTRATIONS .................................................................................................................. 11  
LIST OF TABLES ................................................................................................................................. 12  
ABSTRACT ............................................................................................................................................... 13  
CHAPTER 1: INTRODUCTION ............................................................................................................. 15  
  1.1 Knowledge Discovery .................................................................................................................. 15  
  1.2 Features in Knowledge Discovery .............................................................................................. 16  
  1.3 Research Framework .................................................................................................................. 19  
CHAPTER 2: FEATURE CONSTRUCTION: A PRM-BASED APPROACH FOR IDENTITY MATCHING ........................................................................................... 23  
  2.1 Introduction .................................................................................................................................. 23  
  2.2 Literature Review ....................................................................................................................... 26  
  2.2.1 Aspects of Identity .................................................................................................................. 26  
  2.2.2 Identity Matching Techniques ............................................................................................... 30  
  2.2.2.1 Heuristic Approaches for Identity Matching .................................................................. 30  
  2.2.2.2 Machine Learning for Identity Matching ..................................................................... 31  
  2.2.3 Summary and Research Gaps .............................................................................................. 35  
  2.3 Research Questions ..................................................................................................................... 36  
  2.4 A Probabilistic Relational Model for Identity Matching ............................................................ 36  
  2.4.1 Probabilistic Relational Models (PRMs) .............................................................................. 38  
  2.4.2 PRM-based Identity Matching ............................................................................................... 40  
    2.4.2.1 Feature Construction for Identity Matching ............................................................... 42  
    2.4.2.2 Similarity Measures ...................................................................................................... 45  
    2.4.2.3 PRM Learning Process ................................................................................................. 47  
  2.5 Experimental Study ..................................................................................................................... 47  
    2.5.1 Test-bed ............................................................................................................................... 48  
    2.5.2 Evaluation Metrics .............................................................................................................. 49  
    2.5.3 Hypotheses .......................................................................................................................... 50  
    2.5.4 Experimental Results ........................................................................................................ 51  
    2.5.5 Discussion ........................................................................................................................... 57  
  2.6 Conclusions .................................................................................................................................. 60  
CHAPTER 3: FEATURE CONSTRUCTION: WRITEPRINT FEATURES FOR AUTHORSHIP IDENTIFICATION OF ONLINE MESSAGES ............................................................................. 62  
  3.1 Introduction .................................................................................................................................. 62  
  3.2 Literature Review ....................................................................................................................... 65
# TABLE OF CONTENTS - Continued

3.2.1 Authorship Analysis ........................................................................................................ 65
3.2.2 Writeprint Features ......................................................................................................... 68
3.2.3 Feature Selection ............................................................................................................ 70
3.2.4 Research Gaps ................................................................................................................ 72
3.3 Research Questions ............................................................................................................. 75
3.4 Authorship Identification Based on Writeprint Features .................................................. 76
   3.4.1 A Framework for Authorship Identification of Online Messages ......................... 77
   3.4.2 Writeprint Features ..................................................................................................... 78
      3.4.2.1 Writeprint Features for English ........................................................................... 78
      3.4.2.2 Writeprint Features for Chinese ......................................................................... 82
   3.4.3 A GA-based Feature Subset Selection Model ........................................................... 83
3.5 Experimental Study ............................................................................................................ 84
   3.5.1 Data Descriptions ....................................................................................................... 85
   3.5.2 Evaluation Metric ....................................................................................................... 85
   3.5.3 Experimental Design ................................................................................................ 86
      3.5.3.1 Comparing Different Types of Features ............................................................. 87
      3.5.3.2 Writeprint Feature Selection ............................................................................ 87
   3.5.4 Experimental Results and Discussion ....................................................................... 88
      3.5.4.1 Comparing Different Types of Features ............................................................. 88
      3.5.4.2 Writeprint Feature Selection ............................................................................ 91
3.6 Conclusions ........................................................................................................................ 96

CHAPTER 4: FEATURE SELECTION: OPTIMAL SEARCH-BASED FEATURE SUBSET SELECTION FOR CANCER CLASSIFICATION ................................................................. 98
4.1 Introduction ......................................................................................................................... 98
4.2 Literature Review ................................................................................................................. 102
   4.2.1 Feature Selection ....................................................................................................... 102
      4.2.1.1 Evaluation Criterion ....................................................................................... 104
      4.2.1.1 Filter Models .................................................................................................. 104
      4.2.1.2 Wrapper Models ......................................................................................... 107
   4.2.1.2 Generation Procedure ...................................................................................... 108
      4.2.1.2.1 Individual Feature Ranking ................................................................. 108
      4.2.1.2.2 Feature Subset Selection ....................................................................... 109
   4.2.2 Gene Selection for Cancer Diagnosis ....................................................................... 112
      4.2.2.1 Individual Gene Ranking ............................................................................. 112
      4.2.2.2 Gene Subset Selection ............................................................................... 113
   4.2.3 Research gaps .............................................................................................................. 115
TABLE OF CONTENTS - Continued

4.3 Research Questions................................................................. 117

4.4 Optimal Search-based Feature Subset Selection.............................. 117
  4.4.1 Optimal Search for Feature Subset Selection............................ 118
    4.4.1.1 Genetic algorithm................................................................. 119
    4.4.1.2 Tabu search ........................................................................ 120
  4.4.2 Evaluation Criteria for Feature Subset Selection....................... 123
    4.4.2.1 Filter: MRMR ................................................................. 124
    4.4.2.2 Wrapper: an SVM classifier.............................................. 125
  4.4.4 Four Methods of Gene Subset Selection................................. 126

4.5 Experimental Study........................................................................ 127
  4.5.1 Dataset Descriptions ............................................................... 127
  4.5.2 Evaluation Metrics .................................................................. 128
  4.5.3 Experimental Results ............................................................... 129
  4.5.4 Discussion ................................................................................ 133

4.6 Conclusions and Future Directions................................................ 136

CHAPTER 5: FEATURE CONSOLIDATION: INTEGRATING GENE FUNCTIONAL
RELATIONS FROM HETEROGENEOUS DATA SOURCES............................. 138
  5.1 Introduction.................................................................................. 138
  5.2 Literature Review......................................................................... 139
    5.2.1 Data Sources .......................................................................... 140
    5.2.2 Analytical Techniques ............................................................ 141
    5.2.3 Integration Methods ............................................................... 142
    5.2.4 Research Gaps ....................................................................... 143
  5.3 Research Questions....................................................................... 144
  5.4 A Framework of Gene Relation Integration.................................... 145
    5.4.1 Data Sources and Benchmarks................................................ 146
    5.4.2 Relation Extraction ............................................................... 146
    5.4.3 Relation Integration ............................................................... 147
    5.4.4 Evaluation ............................................................................. 150
  5.5 Experimental Study....................................................................... 151
    5.5.1 Test-bed ................................................................................ 152
    5.5.2 Benchmarks .......................................................................... 153
    5.5.3 Selected Techniques for Relation Extraction............................ 154
      5.5.3.1 Gene Expression Data....................................................... 154
      5.5.3.2 Biological Literature ......................................................... 155
      5.5.3.3 Genome Sequence ........................................................... 156
TABLE OF CONTENTS - Continued

5.5.4 Relation Extraction and Integration .......................................................... 157
5.5.5 Evaluation ............................................................................................... 158
  5.5.5.1 Evaluation Metric .............................................................................. 158
  5.5.5.2 The Effect of Relation Integration .................................................... 158
  5.5.5.3 Reliability of the Integrated Network .............................................. 160
  5.5.5.4 Contribution of Data Sources ......................................................... 161
  5.5.5.5 Analysis of Gene Functional Clusters ............................................. 163

5.6 Conclusions and Future Directions .............................................................. 164

CHAPTER 6: FEATURE CONSOLIDATION: KERNEL-BASED LEARNING FOR
BIOMEDICAL RELATION EXTRACTION ...................................................... 166

6.1 Introduction ............................................................................................... 166
6.2 Literature Review ....................................................................................... 168
  6.2.1 Co-occurrence Analysis ...................................................................... 168
  6.2.2 Rule-based Approaches ...................................................................... 169
  6.2.3 Statistical Learning .............................................................................. 170
  6.2.4 Summary and Research Gaps .............................................................. 174
6.3 Research Questions ..................................................................................... 174
6.4 Kernel-based Learning for Biomedical Relation Extraction ....................... 175
  6.4.1 Entity Recognition .............................................................................. 176
  6.4.2 Relation Annotation ............................................................................ 176
  6.4.3 Kernel Construction ............................................................................ 177
    6.4.3.1 Word Kernel .................................................................................. 178
    6.4.3.2 Sequence Kernel .......................................................................... 179
    6.4.3.3 Tree Kernels ................................................................................. 180
      6.4.3.3.1 Standard Tree Kernel ............................................................... 181
      6.4.3.3.2 Trace-tree kernel ..................................................................... 182
    6.4.4.3 Composite Kernels ...................................................................... 186
  6.4.4 Learning & Evaluation ....................................................................... 186
6.5 Experimental Study .................................................................................... 186
  6.5.1 Test-bed ............................................................................................... 187
  6.5.2 Evaluation Metrics .............................................................................. 187
  6.5.3 Hypotheses ......................................................................................... 188
  6.5.4 Experimental Design ........................................................................... 189
  6.5.5 Results and Discussion ....................................................................... 190
6.6 Conclusions and Future Directions ............................................................. 195

CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS .............................. 196
TABLE OF CONTENTS - Continued

7.1 Contributions .......................................................................................................................... 196
  7.1.1 Theoretical Contributions ............................................................................................... 196
  7.1.2 Technical Contributions ............................................................................................... 197
  7.1.3 Empirical Contributions ............................................................................................... 198
7.2 Relevance to MIS Research .................................................................................................... 199
7.3 Future Directions .................................................................................................................... 201

APPENDIX A FUNCTION WORD FEATURES FOR AUTHORSHIP ANALYSIS ... 203
APPENDIX B TREE HEAD TABLE ........................................................................................... 204
REFERENCES .................................................................................................................................. 205
LIST OF ILLUSTRATIONS

Figure 1.1 A Framework of Feature Engineering for Knowledge Discovery .......... 20
Figure 2.1 A Simplified ER Diagram of a Criminal Database ............................ 37
Figure 2.2 An Extended ER Diagram for Identity Matching ............................... 41
Figure 2.3 An Example of Relational Databases for Identity Matching .............. 42
Figure 2.4 Identity Matching Performances of Three Feature Sets ...................... 55
Figure 3.1 A Framework of Authorship Identification of Online Messages .......... 76
Figure 3.2 GA-based Feature Subset Selection .................................................. 84
Figure 3.3 Experiment Results of Feature Selection on English and Chinese Test-beds . 92
Figure 3.4 Comparison of Writeprints between Three Authors .......................... 96
Figure 4.1 A Framework of Optimal Search-based Feature Subset Selection .......... 118
Figure 4.2 GA-based Feature Subset Selection .................................................. 120
Figure 4.3 TS-based Feature Subset Selection ................................................... 123
Figure 4.4 Comparing Gene Subsets Obtained by Different Methods ................. 130
Figure 5.1 A Framework of Gene Functional Relation Integration ...................... 145
Figure 5.2 A Two-layered Bayesian Network for Gene Relation Integration .......... 148
Figure 5.3 Integrated Network and Individual Evidence form Different Data Sources . 159
Figure 5.4 TP Rate of Relations vs. Joint Confidence Score $P_{cut}$ ......................... 161
Figure 6.1 A Framework of Kernel-based Learning for Relation Extraction .......... 176
Figure 6.2 Three Representations of a Relation Instance .................................... 178
Figure 6.3 An Example of a Minimum Subtree and a Trace in a Parse Tree .......... 184
LIST OF TABLES

Table 2.1 A Summary of Related Studies on Identity Matching........................................ 35
Table 2.2 Attribute Types Supported and Corresponding Similarity Measures................. 46
Table 2.3 Prediction Outcome Matrix............................................................................ 49
Table 2.4 Three Types of Features for Identity Matching............................................. 52
Table 2.5 Identity Matching Performances of Three Feature Sets................................... 55
Table 2.6 Hypotheses Testing of Different Identity Features ........................................ 57
Table 2.7 Personal and Social Features of Identity Pair: A vs. B.................................... 57
Table 2.8 Personal and Social Features of Identity Pair: C vs. D.................................... 58
Table 3.1 A Summary of Previous Studies in Authorship Identification........................ 73
Table 3.2 Writeprint Features for English Messages..................................................... 79
Table 3.3 Accuracy of Four Writeprint Feature Subsets............................................... 89
Table 3.4 Pair-wise t-tests on Accuracy of Different Feature Subsets............................. 89
Table 3.5 Comparison between Full Feature Set and Optimal Feature Subset............... 93
Table 3.6 Illustration of Key Writeprint Features......................................................... 94
Table 4.1 Taxonomy of Feature Selection ................................................................... 103
Table 4.2 A Summary of Previous Gene Selection Studies.......................................... 116
Table 4.3 Descriptions of the Three Gene Array Test-beds........................................... 127
Table 4.4 Pair-wise t-tests between Different Methods............................................... 131
Table 5.1 Relations Extracted from Different Resources Using Different Techniques .... 157
Table 5.2 Distribution of Gene Relations from Different Data Sources.......................... 162
Table 6.1 Tagged Biomedical Relations from 200 MedlineAbstracts............................. 187
Table 6.2 Kernel Methods for Biomedical Relation Detection....................................... 191
Table 6.3 Kernel Methods for Biomedical Relation Classification................................ 191
Table 6.4 Hypotheses Testing for Biomedical Relation Detection................................. 192
Table 6.5 Hypotheses Testing for Biomedical Relation Classification............................ 192
ABSTRACT

With the rapid advance of information technologies, human beings increasingly rely on computers to accumulate, process, and make use of data. Knowledge discovery techniques have been proposed to automatically search large volumes of data for patterns. Knowledge discovery often requires a set of relevant features to represent the specific domain. My dissertation presents a framework of feature engineering for knowledge discovery, including feature construction, feature selection, and feature consolidation.

Five essays in my dissertation present novel approaches to construct, select, or consolidate features in various applications. Feature construction is used to derive new features when relevant features are unknown. Chapter 2 focuses on constructing informative features from a relational database. I introduce a probabilistic relational model-based approach to construct personal and social features for identity matching. Experiments on a criminal dataset showed that social features can improve the matching performance. Chapter 3 focuses on identifying good features for knowledge discovery from text. Four types of writeprint features are constructed and shown effective for authorship analysis of online messages. Feature selection is aimed at identifying a subset of significant features from a high dimensional feature space. Chapter 4 presents a framework of feature selection techniques. This essay focuses on identifying marker genes for microarray-based cancer classification. Our experiments on gene array datasets
showed excellent performance for optimal search-based gene subset selection. Feature consolidation is aimed at integrating features from diverse data sources or in heterogeneous representations. Chapter 5 presents a Bayesian framework to integrate gene functional relations extracted from heterogeneous data sources such as gene expression profiles, biological literature, and genome sequences. Chapter 6 focuses on kernel-based methods to capture and consolidate information in heterogeneous data representations. I design and compare different kernels for relation extraction from biomedical literature. Experiments show good performances of tree kernels and composite kernels for biomedical relation extraction.

These five essays together compose a framework of feature engineering and present different techniques to construct, select, and consolidate relevant features. This feature engineering framework contributes to the domain of information systems by improving the effectiveness, efficiency, and interpretability of knowledge discovery.
CHAPTER 1: INTRODUCTION

1.1 Knowledge Discovery

The advance of information technologies has brought massive amounts of digitized data and information to various application areas. Statistics show that more information has been produced in the past three decades than in the previous five thousand years of human history (Trout, 1997). Organizations have been collecting large volumes of data and information regarding people, activities, processes, and relationships with database management systems. In the scientific domain, high-throughput experimental technologies have produced large-scale scientific data. Furthermore, large repositories of academic publications have been built to preserve and disseminate scientific knowledge in different domains. The Internet has also become a major communication channel and information repository.

Traditional technologies for data processing and analysis are no longer sufficient to handle such a massive amount of information and discover the underlying patterns. We are now facing a growing gap between the generation of data and understanding it (Witten & Frank, 2005). As the volume of data increases, the proportion of it that people understand decreases alarmingly. The underlying and potentially useful knowledge is usually hidden in the raw data but rarely made explicit or taken advantage of. People are drawing in data but starving for knowledge (Han & Kamber, 2001).
Knowledge discovery is the process of automatically searching for patterns from large volumes of data. As the key step in knowledge discovery, data mining is a set of techniques used in “the nontrivial extraction of implicit, previously unknown, and potentially useful information from data” (Fayyad, Piatetsky-Shapiro, Smyth, & Uthurusamy, 1996). The data that we are mining can be stored in either structured format, such as those stored in relational databases, or unstructured format, such as text from the literature repository and the Web. In particular, text mining refers to the process of deriving high quality information from text, whereas Web mining refers to the process of mining patterns from the Web.

1.2 Features in Knowledge Discovery

In order to provide better support for decision making, knowledge discovery techniques have been widely applied in various areas to automatically search for patterns from massive amounts of data. It has been realized that data pre-processing is a critical phase in knowledge discovery (Witten & Frank, 2005). This pre-processing phase ensures that the data fed to the data mining process is of good quality and of proper size. In data pre-processing an important task is dealing with different ways in which data are described and represented. Tools are built for various purposes and handling various data types. No matter what type of data you are dealing with, knowledge discovery often requires a good representation of the specific domain by identifying a set of relevant
features (Dash & Liu, 1997).

Features are also called attributes, variables, properties, or characteristics. A feature is an individual measurable property of the target object. It can have continuous or discrete values. A continuous feature (such as weight, temperature, and salary) has values from the domain of real numbers, whereas a discrete feature (such as gender, names of colors, and ranking of movies) often has a finite number of values. Once a collection of features are defined, a data instance can be described as a vector of corresponding feature values. Therefore, features can be regarded as a representation language. The capability of the language to describe the domain of interest can greatly impact the patterns or models we discover from the data. Developing a good set of features is often a non-trivial task in real-world applications of knowledge discovery.

Features are often defined based on prior knowledge or heuristics. However, in cases where we only have limited knowledge about a specific domain, relevant features are often unknown *a priori*. When the representation language is not sufficient to describe the problem, more features need to be constructed to enrich the language and improve the representation of the target concept. Feature construction has been recognized as an important challenge in knowledge discovery. Features for different knowledge discovery applications are often domain-specific. Furthermore, feature construction methods vary for data in different formats such as relational databases and unstructured text.
When more candidate features are constructed, we can have a richer representation of data instances. Large amounts of candidate features can lead to a feature space of high dimensionality. However, not all of these features are necessarily relevant to the target concept (Dash & Liu, 1997). A relevant feature is supposed to be neither irrelevant nor redundant to the problem. By contrast, an irrelevant feature does not affect the target concept, whereas a redundant feature does not provide any additional information to the target concept. These noisy or useless features may distract a data mining algorithm in its learning process. Moreover, it is difficult to learn meaningful knowledge from few data samples in a high dimensional feature space. This problem is known as the “curse of dimensionality” (Bellman, 1961). Furthermore, the high dimensionality can be an obstacle for some data mining algorithms with limited scalability. Therefore, selecting a subset of key features from the high dimensional feature space is regarded as another crucial challenge in knowledge discovery.

In addition, some knowledge discovery tasks deal with data from diverse sources. Even an instance from the same data source can have multiple representations, each capturing its characteristics from a certain aspect. Features defined on different data sources or data representations may vary in heterogeneous formats and therefore are not compatible. A data mining tool may only deal with a specific feature representation. We may not be able to discover reliable or complete patterns without considering all the
information represented in different languages. Thus arises a third challenge called feature consolidation, which deals with how to integrate different features in heterogeneous formats to better support decision making.

In summary, my dissertation research is aimed at addressing these three challenges by deriving relevant features, eliminating irrelevant or redundant features, and integrating heterogeneous types of features so as to improve the effectiveness, efficiency, and interpretability of knowledge discovery.

1.3 Research Framework

This dissertation presents a research framework of feature engineering for knowledge discovery. In particular, this framework consists of three major components: feature construction, feature selection, and feature consolidation. Specifically, I explored various feature engineering tasks in application areas such as identity matching, cybercrime investigation, cancer diagnosis, gene network learning, and information extraction. Figure 1.1 shows my dissertation framework and the major studies included.
Feature construction is aimed at creating or deriving new features to better represent a domain when relevant features are unknown \textit{a priori}. For data stored in relational databases, features can not only be derived from attributes of each individual table but can also be constructed based on the relational database schema. Chapter 2 focuses on a probabilistic relational model (PRM)-based approach to constructing features so as to support identity matching in large relational databases. The proposed approach can construct both common personal features (e.g., name and date-of-birth) and social contextual features that describe people’s social activities and relationship (Li, Wang, & Chen, 2006). These personal and social features together will provide a more
comprehensive representation of an individual’s identity. We conduct an experiment on a real criminal dataset to examine the discriminating power of social identity features for identity matching.

For data in unstructured textual format, various features can be constructed to capture the characteristics of instances. Chapter 3 focuses on identifying significant features as people’s “writeprint” for authorship analysis of online messages. Based on traditional stylometric theory and the unique characteristics of online messages, in this study we develop four types of writeprint features: lexical, syntactic, structural, and content-specific features. We conduct experiments on English and Chinese corpora to examine these four types of writeprint features for authorship identification (Li, Zheng, & Chen, 2006; Zheng, Li, Huang, Qin, & Chen, 2006).

*Feature selection* is aimed at identifying a subset of significant features from a high dimensional feature space containing large amounts of irrelevant or redundant features. We present a framework of feature selection methods, with a specific focus on optimal search-based feature subset selection methods that attempt to find a global optimal subset of features. Chapter 3 presents a genetic algorithm (GA)-based feature subset selection model to identify significant writeprint features for online authorship identification. Chapter 4 addresses the high dimensionality of microarray data and compares different feature selection techniques for cancer classification. Our experiments on two gene array
datasets demonstrated the effectiveness of optimal search-based feature subset selection to identify key marker genes for cancer diagnosis (Li, Su, Chen, & Futscher, 2007).

*Feature consolidation* is aimed at integrating features that are extracted from various data sources and represented in heterogeneous formats. Chapter 5 presents a Bayesian framework to integrate gene functional relations extracted from diverse data sources. Genomic data sources such as gene expression profiles, biological literature, and genome sequences capture different information about the functional relationships among genes. Our proposed approach consolidates various features based on a unified probabilistic framework into a genome-wide functional network (Li, Li, Su, Chen, & Galbraith, 2006).

Chapter 6 focuses on kernel methods which are an alternative for feature methods in machine learning and knowledge discovery. Kernel methods can capture rich information of data objects represented in complex structures without explicitly enumerating features. Moreover, kernel methods are an effective means of integrating features of heterogeneous data formats. Specifically, we design and compare different kernels for detecting and classifying biomedical relations from literature abstracts.

Chapter 7 of this dissertation highlights the major research contributions, relevance to MIS research, and some interesting future directions that are worth pursuing.
CHAPTER 2: FEATURE CONSTRUCTION: A PRM-BASED APPROACH FOR IDENTITY MATCHING

2.1 Introduction

In this chapter I present a novel approach to construct features from a relational database structure. Specifically, this study is aimed at matching identities in large databases. The constructed features are expected to capture not only people’s personal identity but also their social identity information for identity matching.

Identity management is a critical function of database systems in various organizations. For example, companies keep records of their employees, customers, and suppliers for human resource management, market analysis, and so on. Government agencies manage identity information for various purposes ranging from providing citizen services to crime investigation. Identity matching is a common practice in identity management, which is aimed at searching databases for all identity records that refer to the same real-world person. In recent years, the emphasis on e-commerce and e-government has led to increasing needs for data integration and interoperability among organizations. Identity matching has become a critical task to reduce duplications and consolidate identity records within or across databases. In the wake of the 9/11 terrorist attacks, identity matching has also become a critical issue related to national security.

The lack of a reliable unique identifier across different database makes identity
matching a non-trivial and challenging task (Kent & Millett 2002; Camp, 2003). For example, the Interval Revenue Services (IRS) uses Social Security Numbers (SSN) or Individual Taxpayer Identification Numbers (ITIN) as a unique identifier, while Motor Vehicle Division (MVD) relies on driver’s license numbers to uniquely identify its customers. Hence, identity matching is usually based on inspecting a set of identity features (e.g., name and date of birth) to distinguish a person from others (Donath, 1998). Unfortunately, such identity information is not always reliable, which may result in multiple identity representations for an individual person in one system or across multiple systems. This problem can be subject to either unintentional errors or intentional deception (Wang, Chen, & Atabakhsh, 2004).

Unintentional errors often occur in data management processes such as data entry, storage and transformation. A study showed that the data error rate in typical enterprises could be as high as 30% (Redman 1998). In addition, some information about identities may be unavailable or uncertain. Such missing or information can make identity matching more difficult.

In some cases identity information is also subject to intentional deception. A criminal or a terrorist may intentionally falsify his/her true identity, impersonate another individual’s identity, or use forged identity documents so as to mislead police investigations (Wang et al. 2004). In online auction, a customer may also use false
identities to register multiple user accounts in order to drive up the bidding prices (Snyder, 2000). Identity deception may bring significant financial loss or damaging effects to victims, public, and the national security. Deception detection has become a crucial issue of identity management in law enforcement and intelligence agencies.

Unreliable information further complicates identity matching and information sharing across organizations. For instance, crime investigators may not be able to uncover the connections among multiple crimes committed by a serial offender because he/she may use different names in historical records. Different agencies may fail to exchange information about their common targets. Therefore, identity matching relying on these unreliable features alone usually cannot achieve satisfactory performance. It is desirable to construct additional features to capture complementary identity information to improve the matching performance.

In this essay we propose to use a probabilistic relational model (PRM) based approach to match identities in databases. Specifically, this approach can not only capture common personal identity features such as name and date of birth but can also construct features that represent an individual’s social context information. By capturing these social features, the proposed approach is expected to improve the performance of identity matching.

The remainder of the chapter is organized as follows. Section 2.2 surveys the related
work on identity and identity matching techniques. Two research questions are raised in Section 2.3. Section 2.4 introduces the PRM-based approach to constructing personal and social features for identity matching. Section 2.5 presents our experimental study on a real criminal dataset to examine the proposed approach. We conclude this chapter in Section 2.6 by summarizing our research contributions and future directions.

2.2 Literature Review

In this section we review the different aspects of identity and existing identity matching techniques.

2.2.1 Aspects of Identity

The concept of identity has long been studied in philosophy, psychology, and sociology. In general, identity has two basic aspects: personal identity and social identity.

Personal identity is defined as one’s self-perception as an individual (Cheek & Briggs, 1982). It deals with the necessary and sufficient conditions under which self persists over time. For example, people often ask common questions about their personal identities: Who am I? Where did I come from?

Social identity theories diverge between psychological view and sociological view. The psychological theory defines social identity as one’s self-perception as a member of certain social groups such as nation, culture, gender, and employment (Tajfel & Turner
People within a social group may share some common characteristics such as language, interests, and values. The shared group membership becomes the identity that group members use to distinguish themselves from people in other groups. On the other hand, the sociological theory of social identity “focuses on the relationships between social actors who perform mutually complementary roles (e.g., employer-employee, doctor-patient)” (Deaux & Martin, 2003). The emphasis is on the interpersonal relationships between people and the social structure and context formed based on the relationships (Stryker & Serpe, 1982). In addition, social context also determines the specific roles an individual takes. For example, a man can take different roles in his family. He is the father of his children, the son of his parents, and the husband of his wife. An individual’s social identity, in this sense, is defined by the role-based interactions between the individual and the surrounding people. Deaux and Martin (2003) integrated the psychological view (Tajfel & Turner, 1986) and sociological view (Stryker & Serpe, 1982) into the same framework by regarding them as different levels of social context. The psychological view deals with large-scale social groups, whereas the sociological view deals with the proximate social groups in which members interact with each other. In this framework, social identity becomes a multi-level concept that can be studied at different levels of social groups.

These theories provide a sound theoretical foundation for this study. Although they
do not explicitly indicate which identity features can be used, they point to the directions in which useful information can be used to tackle the identity matching problem. For the identity matching task, the identity information of interest is not in the cognitive and psychological process of an individual’s self-perception. Instead, we are interested in external features that can be used to practically distinguish an individual from others. Clarke (1994) listed a variety of means that can be used in human identification, including appearance, social behavior, names, codes, tokens, bio-dynamics, natural physiography, and imposed physical characteristics. We categorize them into personal identity features and social identity features.

Personal identity features can be further categorized into three types: given identity features, physical characteristics, and biometric features. Given identity features consist of identifiers assigned to an individual at birth, such as name, date and place of birth, mother’s maiden name, and social security number, as indicated by certain tokens (e.g., passport, driver’s license, or membership cards). Physical characteristics include weight, height, color of skin, hair and eye, gender, race, and visible physical markings. Biometric features are natural physiographic characteristics that are unique to an individual such as fingerprints, DNA, iris, hand geometry, voice, etc.

Social identity features deal with an individual’s social activities and social relations. Social activity features describe an individual’s biographical experiences, while social
relation features include the social structure of the group, the relationships between the individual and other members, the roles the individual, etc. Examples of biographical features include education and employment background, credit, medical and crime history, etc. Theories on social identity suggest that social contextual information can be accessed at a large-scale level or a proximate level (Deaux & Martin, 2003). However, large-scale social group features such as nation and race often are too broad to be practically useful in distinguishing individuals. We thus focus on the proximate social context of individuals. At this level, a social group around an individual is defined by the immediate people interacting with him/her.

In reality, different kinds of personal information and social information vary in their availability and reliability. Even in law enforcement databases, identity records often contain only individuals’ simple given identity features, physical characteristics (e.g. weight and height), and sometimes biometric features. The given identity information is subject to deception and many other aforementioned data quality issues. The physical characteristics are not reliable since they often can be altered easily. The hair color, for example, can be easily changed from time to time. Although biometric features such as fingerprints and DNA are the most difficult to falsify and can reliably identify an individual, they are not always available. The social identity of an individual, on the contrary, usually cannot be easily altered or falsified because such information is
embedded in the social context formed through people’s social behaviors and the interactions of group members. The social contextual information can provide additional information for distinguishing an individual from others.

2.2.2 Identity Matching Techniques

Based on the way that a matching decision model is constructed, identity matching can be categorized into heuristic approaches and machine learning approaches.

2.2.2.1 Heuristic Approaches for Identity Matching

Heuristic approaches often require domain experts to manually specify decision rules, based on which the decision models are expected to perform as well as human experts. For instance, in a study on cross-jurisdictional information integration, Marshall et al. (2004) encode domain experts’ heuristics into a simple rule: a pair of identity records are considered a “match” only if their first name, last name, and DOB values are identical. However, in cases of data entry errors (Redman, 1998) or intentional deception (Wang et al., 2004), such exact-match heuristics can not identify records with disagreeing values in any of the three attributes. Most heuristic approaches allow partial match to reduce false negatives in identity matching. The IBM DB2 Identity Resolution (EAS) is a leading commercial product designed to manage identity records (Jonas, 2006). It provides a rule-based matching method that resolves identity records representing the
same person into one entry. For a pair of identity records, a resolution score is calculated by following a set of predefined rules. Some example matching rules are: if the DOB and the last name values of two identity records are identical and the matching score of their first names is above 70/100, then the two records are resolved into one. These heuristic techniques rely on experts’ involvement in defining the rules for optimal matching performance. This process can be time-consuming and has low portability when using the technique in different settings. Moreover, few heuristic approaches have considered social contextual information as a determinant for identity matching.

2.2.2.2 Machine Learning for Identity Matching

Machine learning approaches automatically build decision models by learning from a training dataset of known identity matching cases. Unlike heuristic techniques, machine learning usually requires no or little human intervention. However, the training data instances are still labeled by human experts and the data quality largely determines the effectiveness of the decision models. Machine learning for identity matching can be further categorized into distance-based methods and probabilistic methods.

Distance-based methods define distance measures on different types of descriptive attributes and combine them into a weighted average distance or total similarity score. An identity pair whose overall distance is below a pre-defined threshold will be regarded as a match. Dey, Sarkar, and De (2002) proposed an integer programming approach for record
linkage across two databases. This approach calculates the similarity scores of corresponding personal feature values and combines them into an overall similarity rating as a weighted average. The integer programming model assumes 1-to-1 match of records from two databases and is to minimize the total cost of type-I and type-II errors. However, such an assumption is rarely true in the real world. Brown and Hagen (2003) proposed a data association method for linking criminal records that possibly refer to the same suspect. This method compares two records and calculates a total similarity measure (TSM) as a weighted-sum of the similarity measures of all corresponding feature values. However, this method lacks a training process to determine the threshold for a matching decision. Wang et al. (2004) proposed a record comparison algorithm for detecting deceptive identities by comparing four personal features (name, DOB, SSN, and address) and combining them into an overall similarity score. A supervised learning process determines the threshold of matching decision by training on a set of identity pairs labeled by an expert. However, missing values could significantly affect the performance of the record comparison algorithm (Wang, Chen, Xu, & Atabakhsh, 2006).

Probabilistic methods for identity matching root in the seminal work by Fellegi and Sunter (1969). By posing record linkage as a probabilistic classification problem, they proposed a formal framework to label pairs of identities from two different datasets as “match” or non-match” on the basis of agreement among different features. By assuming
conditional independence among features given the class, the probabilities can be estimated in an unsupervised fashion. Many later studies were built based upon this work to enrich the probabilistic model (Dey, Sarkar, & De, 1998; Winkler, 2002; Ravikumar & Cohen, 2004; Wang, Chen, & Atabakhsh, 2006). Dey et al. (1998) formalized entity matching as an integer programming problem by minimizing the cost of type I and type II errors. For cases when the conditional independence assumption cannot be made, Winker (2002) introduced a latent match variable into the probabilistic model and used an Expectation Maximization algorithm for parameter estimation. Ravikumar and Cohen (2004) and Wang et al. (2006) proposed hierarchical probabilistic models to separate match variable for each feature and an overall match variable for these lower level matches. These studies have shown good performances of probabilistic methods for identity matching. However, probability parameters cannot be easily estimated in the absence of sufficient training data.

Most studies abovementioned consider only personal information for identity matching. Solving such a complex problem requires a combination of multiple techniques and needs to be viewed from a network perspective (Mumford, 1999). Individuals are not isolated but interrelated to each another in the society. Social contexts associated with individuals can provide additional information that can reveal people’s identity. Several recent studies have focused on the use of linkage and contextual information for
improved identity matching. For distance-based methods, researchers have introduced various relational distance measures in graphs and combined them with attribute distance for improved matching performance. Ananthakrishna, Chaudhuri and Ganti (2002) introduce a deduplication method using the dimensional hierarchy over the link relations in data warehouse. This method enhances the personal feature similarity with the similarity between two identity records’ foreign key relations across the relational hierarchy. Bhattacharya and Getoor (2004) proposed relational distance measures to capture the co-authorship patterns in collaboration networks to improve entity resolution. Kalashnikov, Mehrotra, and Chen (2005) incorporated contextual information such as affiliation and co-authorship to disambiguate references using a nonlinear optimization model. In addition, probabilistic models that capture relational structures have been proposed for entity resolution. Pasula et al. (2003) proposed a probabilistic relational model (PRM) for citation matching. This model captures the dependence between identities of co-authors over the relational database structure and predicts whether two different references are the same paper. Culotta and McCallum (2005) used a conditional random field model to capture linked dependency for data deduplication. Bhattacharya and Getoor (2006) adapted a Latent Dirichlet Allocation (LDA) model for collective entity resolution. This model assigns a latent group variable for each reference and predicts its value based on the collaborative patterns.
2.2.3 Summary and Research Gaps

Table 2.1 summarizes related work on identity matching. Specifically, the use of identity features and matching techniques in different studies are shown in this table.

<table>
<thead>
<tr>
<th>Studies</th>
<th>Identity Features</th>
<th>Matching Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Personal</td>
<td>Social</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Marshall et al., 2004)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Jonas, 2006)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Dey et al., 2002)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Brown &amp; Hagen, 2003)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Wang et al., 2004)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Ananthakrishna et al., 2002)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>(Bhattacharya &amp; Getoor, 2004)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>(Kalashnikov et al., 2005)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>(Fellegi &amp; Sunter, 1969)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Dey et al., 1998)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Winkler, 2002)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Ravikumar &amp; Cohen, 2004)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Wang et al., 2005)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Pasula et al., 2003)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>(Culotta &amp; McCallum, 2005)</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>(Bhattacharya &amp; Getoor, 2006)</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

As shown in Table 2.1, many studies use machine learning approaches (either distance-based or probabilistic methods) to address the identity matching problem. However, most existing studies only use personal identity features to match identity
records. Although several recent studies have taken relational information into account and shown good matching performance, we have not seen a systematic methodology of constructing personal and social features for identity matching. Moreover, most studies abovementioned focus on entity resolution in literature repository which mainly deals with ambiguous references of authors and papers. In contrast, identity matching in databases of organizations like law enforcement often deals with data with more missing values, uncertainty, and even intentional deception. Therefore, capturing social contextual information is of critical importance to support identity matching.

2.3 Research Questions

In order to address these research gaps, this study is aimed at constructing features to capture not only personal but also social identity features to improve identity matching in databases. Specifically, we focus on the following two research questions:

Q1. How can we construct features to capture an individual’s social context information (social activities and social relations) from databases for identity matching?

Q2. Can social identity features improve the performance of identity matching?

2.4 A Probabilistic Relational Model for Identity Matching

Personal identity information and social contextual information about individuals are stored in different tables of relational databases. Figure 2.1 shows a simplified
entity-relation (ER) diagram of a criminal database as an example. In this ER diagram, **Person** and **Incident** are entity classes while **Participation** is the relationship between **Person** and **Incident**. Such a relational database structure contains rich information about individuals’ identity. Considering only attributes in the “**Person**” table is insufficient for identity matching in that relational information from the tables “**Participation**” and “**Incident**” is lost. Identity matching requires capturing personal and social information about individuals from different related tables.

![Figure 2.1 A Simplified ER Diagram of a Criminal Database](image)

Standard data mining algorithms learn from a flat representation of independent instances, each with its own separate attributes. Few of these algorithms are capable of handling data in relational form. Relational learning is aimed at extracting patterns from multiple related tables in a database structure (Dzeroski & Lavrac, 2001). The assumption
that data instances are independent from each other is dropped in relational learning. Instead, the central focus of relational learning is the relations among data instances.

A formal approach for relational learning is called probabilistic relational models (PRMs) (Friedman, Getoor, Koller, & Pfeffer, 1999; Getoor, Friedman, Koller, & Taskar, 2002). As a relational version of Bayesian networks, PRMs are generative models of joint probability distribution capturing probabilistic dependencies between data instances and between the properties of related instances in a relational domain. Applications of relational learning and PRMs span the realms of link prediction, social network modeling, citation matching, recommender systems, and so on (Getoor & Sahami, 1999; Pasula et al. 2003; Popescul & Ungar, 2003; Huang, Zeng, & Chen, 2004). These studies have shown the PRMs’ capability of capturing the entire rich structure of patterns encoded by the relational database schema. Therefore, in this study we adopt the PRMs to learn dependencies between identities and their related data objects in relational databases for identity matching. Specifically, we show how PRMs can be customized for making matching decision between identity pairs and how these models are well-suited for this task.

2.4.1 Probabilistic Relational Models (PRMs)

A probabilistic relational model consists of a set of classes (tables) $X_1, \ldots, X_n$ and a set of relations $R_1, \ldots, R_m$ between the entities. Each class $X$ is associated with a set of
descriptive attributes $A(X)$ and a set of reference slots (foreign keys) $R(X)$. We denote the attribute $A$ of $X$ as $XA$ and the reference slot $R$ as $XR$. $X$ is called the domain of $R$, while the corresponding class $Y$ that $XR$ refers to is called the range of $R$. Each reference slot $\rho$ denotes a mapping function from $\text{Domain}[\rho] = X$ to $\text{Range}[\rho] = Y$, while $\rho^{-1}$, called an inverse reference slot, maps from $\text{Range}[\rho] = Y$ to $\text{Domain}[\rho] = X$. A slot chain is composed of multiple slots defined as $\tau = \rho_1.(\ldots)\rho_k$, where $\text{Range}[\rho_1] = \text{Domain}[\rho_{i+1}]$. Since a database contains multiple related entities tables, various dependencies between the attributes of related entities can be explored through different slot chains.

As an extension of Bayesian network learning, PRM learning is to extract the probabilistic dependencies of various descriptive attributes and reference slots over a database. A probabilistic relational model $\Pi$ is composed of an acyclic directed graph, $S$, and the parameters associated with it, $\Theta_S$. In particular, $S$ describes the dependency structure of attributes and slots by assigning a set of parents $\text{Pa}(XA)$ to each $XA$. $\Theta_S$ represents the parameters characterizing the conditional probabilistic distributions (CPDs) (Friedman et al., 1999). Each $XA$ is associated with a conditional probability distribution that specifies $P(XA \mid \text{Pa}(XA))$. Given a complete initialization $I$ of objects in each class $X$ as well as their values for each attribute and references slot, a PRM can be learned by finding the model $\Pi^*(S^*, \Theta^*)$ that best fits $I$. A search-and-scoring approach is a standard process to find the best PRM. A commonly scoring metric is:
\[ \log P(S \mid I) = \log P(I \mid S) + \log P(S) + C \]

where \( P(I \mid S) \) is the marginal likelihood \( P(I \mid S) = P(I \mid S, \Theta S) P(\Theta S \mid S) d\Theta S \). To constrain the computational complexity, standard greedy search algorithms can be used to search for the optimal structural \( S^* \). Given the optimal structure, parameters of CPDs, \( \Theta S^* \) can be estimated to complete the model specification.

2.4.2 PRM-based Identity Matching

Identity matching is to determine whether two identities refer to the same person. Conceptually, this problem can be regarded as an application of relational learning. The linkage (match or non-match) between each identity pair is the modeling focus. PRMs with existence uncertainty are able to model the existence of certain records (Getoor et al. 2002). For identity matching, we introduce a class of \textit{Match} (\texttt{pid1}, \texttt{pid2}, exist) to model the undetermined match relationship between two individuals. Figure 2.2 illustrates the added class “\textit{Match}” in the ER diagram of the criminal database. In this \textit{Match} class, \texttt{pid1} and \texttt{pid2} are foreign keys to the \texttt{Person} class; \texttt{exist} is an existence attribute whose value is from \{true, false\}. Specifically, \texttt{Match.exist} equals \textit{true} if the pair of identities refer to the same person and equals \textit{false} otherwise. The goal of identity matching essentially is to determine the value of \texttt{Match.exist} for each identity pairs in the database.
Figure 2.3 shows an example of the four relational tables from a criminal database. The three identities (John, Jenny, and Jon) in the Person table have different name, dob, and ssn. The Participation table indicates their involved incidents and their roles. The Incident table shows the crime type and reported time of each incident. Records in these tables are related by slot chains of foreign keys, as denoted by arrows. To determine the values of Match.exist of each identity pair in the Match table, we need to follow the slot chains to capture information in the three tables: Person, Participation, and Incident. In this example, high similarity of personal features (name, dob, and ssn) between P1 (John) and P3 (Jon) suggests that they are the same person. Following the slot chains, more information about P1 and P3 can be extracted: they were both “arrestees” in two “assault” incidents (C1 and C2) reported at night (“21:40” and “20:35”), respectively; P2
(Tom) was the “victim” in both incidents. Such information reveals their identity from a social perspective and further confirms the matching decision. This section introduces how to use PRM to construct features from such a relational structure and learn a decision model for identity matching.

![Figure 2.3 An Example of Relational Databases for Identity Matching](image)

2.4.2.1 Feature Construction for Identity Matching

Using the PRM notation introduced previously, the dependency structure $S$ of PRM defines the parents $Pa(X.A)$ for each attribute $X.A$. Because identity matching is aimed at predicting the value of $Match.exist$, we only need to focus on the partial dependency
structure for `Match.exist`. Starting from a target pair of identity pairs (pid1, pid2) in the `Match` class, various potential features involved in this dependency structure can be constructed through reference slot chains for identity matching.

- **Personal Identity Features**

  A slot chain of length = 1 starts from the `Match` class to the `Person` class. Descriptive attributes of the `Person`, such as `name`, `dob`, `ssn`, and so on, are the simplest and most straightforward features derived from slot chains. For instance, we use `[Match.pid1].dob` to denote a target individual pid1’s date of birth. These features compose the personal identity features that have been commonly used in identity matching based on feature similarity.

- **Social Identity Features**

  As the length of the slot chain increases and inverse reference slots are introduced, more complex features can be constructed to capture information from other classes such as `Participation` and `Incident`. Unlike personal identity features, these new derived features could reveal an individual’s social behavior and contextual information. These social identity features can be further divided into social activity features and social relation features.

  By extending the slot chain to the class of `Participation` and `Incident`, we can construct new features that represent the target individual’s social activities, i.e., the
incidents in which he or she is involved in. For example, 
\[ \text{Match.pid1}.[\text{Participation.pid}]^{-1}.iid \], represents the set of incidents in which the target individual \text{pid1} is involved; \[ \text{Match.pid1}.[\text{Participation.pid}]^{-1}.role \] represents the roles of the target person in his/her involved incidents; 
\[ \text{Match.pid1}.[\text{Participation.pid}]^{-1}.[\text{Participation.iid}].\text{crimetype} \] represents the crime types of the \text{pid1}’s involved incidents.

These features describe the social activities of the target individual. We are also interested in the social relationship and social group of an individual. We consider the collection of individuals who are directly related to the person to be his/her social group. In other words, an individual’s social relation features represent the characteristics of his/her immediate “neighbors.” We can further extend the slot chains back to classes \text{Participation} and \text{Person} to construct these social relation features. For example, 
\[ \text{Match.pid1}.[\text{Participation.pid}]^{-1}.[\text{Participation.iid}].[\text{Participation.iid}]^{-1}.[\text{Participation.pid}].\text{pid} \] represents individuals who are involved in at least one incident with \text{pid1}. 
\[ \text{Match.pid1}.[\text{Participation.pid}]^{-1}.[\text{Participation.iid}].[\text{Participation.iid}]^{-1}.[\text{Participation.pid}].[\text{Participation.pid}]^{-1}.\text{role} \] represents the roles of \text{pid1}’s neighbors in their involved incidents. 
\[ \text{Match.pid1}.[\text{Participation.pid}]^{-1}.[\text{Participation.iid}].[\text{Participation.iid}]^{-1}.[\text{Participation.pid}].[\text{Participation.pid}]^{-1}.[\text{Participation.iid}].\text{crimetype} \] represents the crime types of the incidents that \text{pid1}’s neighbors are involved.
These social activity and relation features represent an individual’s identity from a social perspective. We believe that the incorporation of these features could improve the performance of identity matching. In theory, an infinite number of potential features can be constructed by extending the length of slot chains. However, more features will also increase the computation complexity while searching for the optimal PRM. In addition, as the length of reference slot chain increases, the derived features tend to become less interpretable. In this study we constrain the length of the slot chain and only focus on these three major types of features.

2.4.2.2 Similarity Measures

For PRM-based identity matching, the value of \texttt{Match.exist} to predict is between a pair of identity records, \texttt{pid1} and \texttt{pid2}. Each of them is represented by a set of features constructed separately from slot chains. Prediction in identity matching is based on the similarity between each pair of feature values. For single-valued features in different data types (e.g., numeric, textual, binary, or nominal), appropriate similarity measures need to be selected. Table 2.2 lists the four data types and the corresponding similarity measures.
Table 2.2 Attribute Types Supported and Corresponding Similarity Measures

<table>
<thead>
<tr>
<th>Attribute Types</th>
<th>Similarity Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
<td>$1 - \frac{</td>
</tr>
<tr>
<td>Binary</td>
<td>$\begin{cases} 1, \text{when } s_1 \text{ and } s_2 \text{ agree} \ 0, \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>Nominal</td>
<td>$\begin{cases} 1, \text{when } s_1 \text{ and } s_2 \text{ agree} \ 0, \text{otherwise} \end{cases}$</td>
</tr>
<tr>
<td>Textual strings</td>
<td>$1 - \frac{ED(S_1, S_2)}{\max(\text{length}(S_1), \text{length}(S_2))}$</td>
</tr>
</tbody>
</table>

*: ED($S_1$, $S_2$) is the Levenshtein edit distance function (Levenshtein, 1966)

Social activity and relation features are often derived through long reference slot chains. In cases where chains contain one-to-many mappings, the derived features can be of multiple values. For example, `[Participation.pid]^{-1}` indicates a mapping function from an individual to all of his/her involved incidents. The notion of aggregation from database theory is the proper tool to address this issue by converting a multi-valued set into a single-valued feature. There are many useful notions of aggregation, such as cardinality (the number of distinct values in the set), mode (the most frequently occurring value), mean, median, maximum, minimum, etc. For example, $\text{mode}\{[\text{Match.pid1}].[\text{Participation.pid}]^{-1}.[\text{Participation.iid}].[\text{Participation.iid}]^{-1}.[\text{Participation.pid}].\text{pid}\}$ represents the most frequent neighbor of the target individual. In addition, we can also jointly use aggregation operator such as cardinality and multi-set
operations such as intersection and union to derive the Jaccard’s coefficient, a commonly used similarity measure. For example, the similarity between the neighbors of pid1 and pid2 can be computed by $\frac{\text{cardinality}\{\text{intersection}\{[\text{Match.pid1}].\tau, [\text{Match.pid2}].\tau\}\}}{\text{cardinality}\{\text{union}\{[\text{Match.id1}].\tau, [\text{Match.pid2}].\tau\}\}}$, where $\tau = \text{[Participation.pid]}^{-1}$. [Participation.iid].[Participation.iid]$^{-1}$.[Participation.pid].pid. In this formula, the numerator denotes the number of the two individuals’ common neighbors and the denominator denotes the total number of their neighbors.

2.4.2.3 PRM Learning Process

With the three types of features and their corresponding similarity measures, PRM learning is to find the optimal partial dependency structure involving Match.exist. Standard hill-climbing greedy search algorithms can be employed to search for the optimal structural $S$. With the optimal dependency structure, maximum likelihood parameter estimation can be performed to complete the model specification. In this study we used a naive Bayesian classifier for binary prediction (Langley & Sage, 1994) to estimate the probability $P(\text{Match.exist} = 1 \mid \text{derived features of Match.exist})$ for undetermined identity pairs.

2.5 Experimental Study

In order to examine the effectiveness of our proposed PRM-based approach, we
conducted an experimental study on a real criminal dataset. In particular, we compared the predictive power of the three types of constructed features for identity matching.

2.5.1 Test-bed

We used real criminal identity records from the Tucson Police Department (TPD) as our experimental test-bed. The TPD dataset contains information on 2.03 million individuals involved in illegal activity over 16 years. As our domain expert (a TPD detective with over 20 years of experience) suggested, in this study we selected a subset of individuals involved in gang activity. Criminals related with gangs often undergo intensive investigation and therefore their data is more complete. In total, this gang subset contained 3,694 identity records. In order to capture the social groups of these criminals, we also incorporated 34,364 individuals who were involved in any incidents related to these 3,694 criminals into our test-bed.

A gold standard for labeling identity pairs is required for learning a decision model of identity matching. From the TPD dataset, the gold standard includes the pre-associations between the records that have been identified by previous police investigations. During an investigation, if it is found that a single individual has multiple records in the database, then these records are merged into a single identity and given an identifier. Thus, in the dataset, multiple records may be associated with a single identifier, which denotes that expert knowledge was used to merge the identities. In addition to such
mergers, the gold standard was also refined to include records that were merged based on first name, last name, and date-of-birth. Any individual records that had exactly the same values for these fields were considered the same identity and assigned the same identifier. This heuristic was used at the suggestion of domain experts. Based on this gold standard, these 3,694 identities records refer to 2,119 unique individuals.

2.5.2 Evaluation Metrics

We treat identity matching as a pair-wise classification. Choice of evaluation metrics is an important issue for identity matching. The distribution of matches and non-matches tend to be highly screwed in databases. Often only less than 1% of all pairs are duplicates. Therefore, accuracy may not be the best metric because simply labeling all pairs as non-match would give 99% accuracy. Instead, we followed most identity matching studies and chose precision, recall and F-measure as the evaluation metrics (Bhattacharya & Getoor, 2005). Table 2.3 shows four outcomes of prediction compared with truth.

<table>
<thead>
<tr>
<th>Truth Prediction</th>
<th>Identities of the same person</th>
<th>Identities of different persons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match</td>
<td>True Positive (TP)</td>
<td>False Positive (FP)</td>
</tr>
<tr>
<td>Non-match</td>
<td>False Negative (FN)</td>
<td>True Negative (TN)</td>
</tr>
</tbody>
</table>

Based on the prediction outcomes, precision and recall as follows:
Specifically, precision indicates the correctness of match prediction, whereas recall indicates the completeness of correct match prediction. F-measure gives the weighted harmonic mean of precision and recall:

\[ F\text{-}\text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \]

2.5.3 Hypotheses

Based on the relational database schema, we can construct features through slot chains to predict the value of \textbf{Match.exist} for identity pairs. As we extend the slot chains, more complex features can be derived to capture the identity information of individuals. By incorporating additional features into the decision model, we expect to achieve improved identity matching performance.

We denote personal identity features, social activity features, and social relation features as \( F_p \), \( F_{sa} \), and \( F_{sr} \), respectively. To examine the discriminating power of the social features for identity matching, we focus on the following two hypotheses in our experimental study.

\[ \text{H1}: F_p + F_{sa} > F_p \]
In H1, we hypothesize that identity matching using both personal identity features and social activity features can achieve better performance than using personal identity features alone. Social activity features describe the characteristics of an individual’s involved activities as well as his/her social roles in these activities. They can potentially reveal individuals’ identity from a social perspective.

\[ H2: F_p + F_{sa} + F_{sr} > F_p + F_{sa} \]

In H2, we hypothesize that identity matching using personal identity features, social activity features, and social relation features can achieve better performance than using personal identity features and social activity features. Social relation features capture an individual’s relationships with other people and the characteristics of his/her social groups. These features are expected to provide additional information and further improve the identity matching performance.

2.5.3 Experimental Design

Table 2.4 summarizes the features we constructed using the PRM-based approach in our experiments. We have personal identity features, social activity features, and social relation features. Because these social features derived from slot chains are often of multiple values, we collectively use aggregation operators such as mode, intersection, union and cardinality to derive single-valued features: mode{} for the most frequent item, cardinality{intersection{}} for the number of common items in two sets,
cardinality\{union\{} for the number of all items in two sets, and
\[
\frac{\text{cardinality}\{\text{intersection}\}}}{\text{cardinality}\{\text{union}\}}
\]
for Jaccard’s similarity of two sets.

Table 2.4 Three Types of Features for Identity Matching

<table>
<thead>
<tr>
<th>Feature Annotations</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Personal Identity Features</strong> ((F_p))</td>
<td></td>
</tr>
<tr>
<td>[Match.pid1].fn</td>
<td>First name</td>
</tr>
<tr>
<td>[Match.pid1].mn</td>
<td>Middle name</td>
</tr>
<tr>
<td>[Match.pid1].ln</td>
<td>Last name</td>
</tr>
<tr>
<td>[Match.pid1].dob</td>
<td>Data of birth</td>
</tr>
<tr>
<td>[Match.pid1].ssn</td>
<td>SSN</td>
</tr>
<tr>
<td>[Match.pid1].address</td>
<td>Address</td>
</tr>
<tr>
<td><strong>Social Activity Features</strong> ((F_{sa}))</td>
<td></td>
</tr>
<tr>
<td>[Match.pid1].[Participation.pid]^{-1}.role</td>
<td>(\text{pid1}’\text{s role in previous incidents})</td>
</tr>
<tr>
<td>[Match.pid1].[Participation.pid]^{-1}.[Participation.iid].crimetype</td>
<td>The crime type of (\text{pid1}’\text{s involved incidents})</td>
</tr>
<tr>
<td>[Match.pid1].[Participation.pid]^{-1}.[Participation.iid].time</td>
<td>The time of (\text{pid1}’\text{s involved incidents})</td>
</tr>
<tr>
<td><strong>Social Relation Features</strong> ((F_{sr}))</td>
<td></td>
</tr>
<tr>
<td>[Match.pid1].[Participation.pid]^{-1}.[Participation.iid].[Participation.iid]^{-1}.[Participation.pid].pid</td>
<td>(\text{pid1}’\text{s neighbors})</td>
</tr>
<tr>
<td>[Match.pid1].[Participation.pid]^{-1}.[Participation.iid].[Participation.iid]^{-1}.[Participation.pid].[Participation.pid]^{-1}.role</td>
<td>The role of (\text{pid1}’\text{s neighbors in their involved incidents})</td>
</tr>
<tr>
<td>[Match.pid1].[Participation.pid]^{-1}.[Participation.iid].[Participation.iid]^{-1}.[Participation.pid].[Participation.pid]^{-1}.[Participation.iid].crimetype</td>
<td>The crime type of (\text{pid1}’\text{s neighbors’ involved incidents})</td>
</tr>
<tr>
<td>[Match.pid1].[Participation.pid]^{-1}.[Participation.iid].[Participation.iid]^{-1}.[Participation.pid].[Participation.pid]^{-1}.[Participation.iid].time</td>
<td>The time of (\text{pid1}’\text{s neighbors’ involved incidents})</td>
</tr>
</tbody>
</table>
In this study identity matching is formalized as a pair-wise classification problem. However, due to the large amount of identity records, comparing all the identity pairs in a database is often impractical because of its high computational complexity. In our experiment we used an adaptive detection algorithm (Monge, 1997; Wang et al., 2006) to pre-cluster potentially matched identities based on the personal identity features only. Specifically, this algorithm first sorts all identity records by a certain attribute; each record is only compared with one representative record from each cluster within a window size \( w \); a record is merged into a cluster if a match or a partial match is found. In our case of criminal identity matching, identity records are sorted by last name because last name is a relatively reliable identifier and usually has no missing values (this is often enforced by police record management systems). The adaptive window size for this study was set at 20. It can be adjusted according to the size of the dataset. A Euclidean function was used to measure the similarity between two identity records.

The adaptive detection algorithm grouped the 3,694 identities from the gang dataset into 812 clusters, including 803 clusters that contain multiple identities. We conducted with-cluster pair-wise comparisons for these 803 clusters. In total, we have 9,551 identity pairs, including 2,661 matches and 6,890 non-matches according to the gold standard. Each instance in this dataset is a pair of identity records represented by their personal and social features. We adopted a Bayesian classifier to learn the conditional dependencies
between the value of Match.exist and the corresponding identity features. To examine the discriminating power of different features, we first used only the personal identity features \( F_p \) to learn the classifier; next, social activity features \( F_{sa} \) and social relation features \( F_{sr} \) were incorporated separately into the model. The standard 10-fold cross-validation method was used to estimate the performances of classification models.

2.5.4 Experimental Results

The performance of identity matching is summarized in Figure 2.4 and Table 2.5. The classification model based on only the six personal identity features \( F_p \) achieved 97.48% precision, 95.09% recall, and 96.27% F-measure. By incorporating the social activity features \( F_{sa} \) into the classification model, we found that the precision increased to 99.59%, recall to 96.12%, and F-measure to 97.82%. Furthermore, when social relation features were added, the classification model achieved the best performances for all the three evaluation metrics: 99.81% precision, 96.13% recall, and 97.94% F-measure.
Further analysis on the number of false positives (FPs) and false negatives (FNs) can bring us more insights into how the social features helped improve identity matching. An FP is an unmatched identity pair that is predicted as a “match,” while an FN is a matched identity pair that is predicted as a “non-match.” The classification model based on only personal identity features resulted in 67 FPs and 134 FNs. When social activity features
were added into the classification model, we found that the number of FPs was reduced from 67 to 11 and the number of FNs was reduced from 134 to 107. In other words, several wrong matching decisions based on personal features only were corrected by the social activity features. Furthermore, when the social relation features are added, FP was further reduced from 11 to 5, which indicated that social relational information could also help detect mismatched identity pairs by other features. However, we did not observe any additional reduction of FN in the experiment.

We conducted paired t-tests to test the three hypotheses. Table 2.6 shows the hypotheses testing results, where p-values with * and ** indicates significant difference at the level of $\alpha = 0.05$ and 0.01, respectively. According to the paired t-tests, H1 was supported. By incorporating social activity features into the classification model, we observed significant improvement in precision, recall, and F-measure ($p = 0.0018$, 0.0167, and 0.0003). However, the incorporation of social relation features did not achieve significant improvement in the three evaluation metrics with all p-values greater than 0.1. Therefore, H2 was not confirmed. The experiments showed that personal identity features can provide strong evidence for matching identity records and achieve high performance. The constructed social features provide complementary information about individuals’ identity. Although social activity features were shown to effectively rectify wrong matching decisions by reducing FP and FN, the improvement that resulted from adding
social relation features was limited.

Table 2.6 Hypotheses Testing of Different Identity Features

<table>
<thead>
<tr>
<th>Hypotheses</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1 $F_p &lt; F_p + F_{sa}$</td>
<td>0.0018**</td>
<td>0.0167*</td>
<td>0.0003**</td>
</tr>
<tr>
<td>H2 $F_p + F_{sa} &lt; F_p + F_{sa} + F_{sr}$</td>
<td>0.1318</td>
<td>0.1280</td>
<td>0.1080</td>
</tr>
</tbody>
</table>

2.5.5 Discussion

To illustrate how social features helped match identities, we examined the identity matching results and found several examples where social features rectified the false predictions made by personal features alone. Tables 2.7 and 2.8 show two identity pairs and some of their corresponding feature values. Due to data confidentiality consideration, the real personal information in the following examples is not revealed.

Table 2.7 Personal and Social Features of Identity Pair: A vs. B

<table>
<thead>
<tr>
<th>Person #</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Zeno, Andrew</td>
<td>Zeno, Andrew</td>
</tr>
<tr>
<td>DOB</td>
<td>19650105</td>
<td>19821004</td>
</tr>
<tr>
<td>Address</td>
<td>3131 N EL CAPITAN</td>
<td>2121 N AVENIDA EL CAPITAN</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>CrimeType</td>
<td>Drug offences</td>
<td>Assaults</td>
</tr>
<tr>
<td>Neighbor</td>
<td>{E, P, Q}</td>
<td>{E, M}</td>
</tr>
<tr>
<td>Neighbor's CrimeType</td>
<td>Drug offences</td>
<td>Assaults</td>
</tr>
<tr>
<td>Neighbors’ Role</td>
<td>Arrested</td>
<td>Victim</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Table 2.8 Personal and Social Features of Identity Pair: C vs. D

<table>
<thead>
<tr>
<th>Person #</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Smith, Matthew</td>
<td>Smith, Liss</td>
</tr>
<tr>
<td>DOB</td>
<td>19740627</td>
<td>19740627</td>
</tr>
<tr>
<td>Address</td>
<td>NULL</td>
<td>NULL</td>
</tr>
<tr>
<td>Role</td>
<td>Arrested</td>
<td>Arrested</td>
</tr>
<tr>
<td>CrimeType</td>
<td>Auto theft, drug offences</td>
<td>Auto theft, drug offences</td>
</tr>
<tr>
<td>Neighbors</td>
<td>{R, S, T}</td>
<td>{R, S, T}</td>
</tr>
</tbody>
</table>

The first example illustrates how social contextual information corrected a false positive predicted by personal features alone. In this example, two identity records A and B had identical first and last names: “Zeno, Andrew.” Their DOB and address values were very similar as well. Based on personal features alone the classification model predicted them to be the same person. When their social features were taken into consideration, however, many discrepancies were found between the two persons. In particular, person A was mostly involved in “drug offences,” while person B was mainly involved in other “assaults.” Although A and B happened to have a common neighbor E who was involved in some incidents with each of them respectively, their social groups in general had very different characteristics. In particular, most of A’s “neighbors” were often suspects or arrestees in drug-related crimes, while B’s “neighbors” were mostly
victims in assaults. These disagreements in social features represented their different social behavior and therefore differentiated A from B. The similar personal feature values might be caused by one person intentionally concealing his identity by using the other person’s identity information.

The second example illustrates how social features helped detect two matching identities when personal identity feature values suggested a non-matching prediction (i.e., FN). In this example, persons C and D shared the same last name (“Smith”) but had different first name (“Matthew” and “Liss”). Their address information was not available. The classification model based on personal features alone considered them as an unmatched identity pair. However, their social features showed that they played the same social roles (“arrested”) in the same types of crimes (“auto theft” and “drug offences”) and they shared common neighbors (R, S, and T). Therefore, when these social features were considered, C and D were classified as a matched identity pair. This example shows that when personal features contain intentional errors or unintentional deception, social features can provide additional information and help detect matched identities.

In summary, the social activity and relation features can rectify wrong matching decision based on personal features alone by identifying both FP and FN pairs. For identity matching in databases, especially criminal databases, the capability of detecting errors or deceptions in personal features is critical. These social features provide
complementary information about individuals’ true identity to support matching decision.

2.6 Conclusions

Identity matching is an important and challenging task in database management. Personal identity features such as name and date of birth have been commonly used in identity matching but cannot always provide sufficient information about people’s identity. The major contribution of this study is that we adopt and customize PRMs to construct features through slot chains in the relational database for identity matching. Not only can PRMs derive personal identity features, but also construct some novel features (i.e., social activity and social relation features) that can capture the social contextual information of individuals. The experimental results on a criminal dataset were promising and supported our postulation that combining social features with personal identity features could improve the performance of identity matching.

The proposed approach can be extended in the following directions. Under the probabilistic relational models, an infinite number of features can be derived by extending slot chains and applying aggregations. We are interested in identifying the key features from these derived features so as to improve the performance and efficiency of identity matching. Besides, in this study we treat matching between each identity pair as an independent decision. In reality, determining that two identities as a match may affects the values of corresponding social features and therefore allow us to make additional
inferences. Furthermore, the learned matching decision model may also help us refine the gold standard of labeling matching pairs in the training dataset. Therefore, the identity matching process should be performed in an iterative manner. Finally, we plan to extend the PRM-based feature construction approach to other applications such as link prediction in social networks.
CHAPTER 3: FEATURE CONSTRUCTION: WRITEPRINT FEATURES FOR AUTHORSHIP IDENTIFICATION OF ONLINE MESSAGES

3.1 Introduction

The proposed approach in Chapter 2 is aimed at constructing informative features from structured data in relational databases. In contrast, a more challenging task is constructing and extracting features from unstructured data such as text. Therefore, this chapter focuses on constructing features from data in textual format. In particular, we develop a framework of authorship identification of online messages based on writeprint features.

The rapid development and proliferation of Internet technologies and applications have created a new way for information sharing in the cyberspace. A wide range of activities have evolved over the Internet, ranging from simple information exchange and resource sharing to virtual communications and e-commerce activities. In particular, online messages have been extensively used to distribute information over Web-based channels such as e-mail, Web sites, newsgroups, and chat rooms. Unfortunately, online messages can also be misused for the distribution of unsolicited, inappropriate information such as junk mail (commonly referred to “spamming”) and offensive/threatening messages. Moreover, criminals have been using online messages to distribute illegal materials, including pirated software, child pornography materials, and
stolen property (Power, 2002). In addition, criminals and terrorist organizations also use online messages as one of their major media for communication and propaganda. These activities have spawned the concept of “cybercrime” (Chang, Chung, Chen, & Chou, 2003). Cybercrime was defined by Thomas and Loader (2000) as illegal computer-mediated activities which can be conducted through global electronic networks.

A common characteristic of online messages is anonymity. People usually do not need to provide their real identity information, such as name, age, gender, and address. In many misuse or crime cases of online messages, the sender will attempt to hide his/her true identity to avoid detection. For example, the sender’s address can be forged or routed through an anonymous server, or the sender can use multiple usernames to distribute online messages via different anonymous channels. Therefore, the anonymity of online messages imposes unique challenges to identity tracing in cyberspace. As a result of the sheer growth of cyber users and activities, efficient automated methods for identity tracing are becoming imperative. Since people are not usually required to provide their real identity in cyberspace, the anonymity makes identity tracing a critical problem in cybercrime investigation. This problem is further complicated by the sheer amount of cyber users and online activities.

Fingerprint-based identification has been the oldest biometric technique successfully used in conventional crime investigation. The unique, immutable patterns of a fingerprint,
i.e., the pattern of ridges and furrows as well as the minutiae points, can help a crime investigator determine the identities of suspects. However, there are no fingerprints to be found in the cybercrime scene, i.e., the internet. Fortunately, there is another type of print, which we call writeprint, hidden in people’s writings. Similar to fingerprints, writeprint is composed of multiple features, such as vocabulary richness, length of sentence, use of function words, layout of paragraphs, and key words. These writeprint features can represent an author’s writing style, which is usually consistent across his or her writings, and further become the basis of authorship analysis.

This essay is aimed at constructing and examining writeprint features for authorship identification of online messages. Furthermore, we introduce a method of identifying the key features for online authors to facilitate identity tracing in cybercrime investigation. Due to the international and multi-lingual nature of the Internet, we also evaluate the performance of writeprint-based authorship identification for different languages.

The remainder of the chapter is organized as follows. Section 3.2 surveys related studies on authorship analysis and feature selection methods. I raise four research questions in Section 3.3. In Section 3.4 I develop an authorship identification framework and introduce features constructed for online message. Furthermore, I introduce a GA-based feature selection method to identify key writeprint features. Section 3.5 presents an experimental study on both English and Chinese corpora. We conclude this
essay in Section 3.6 by summarizing our research contributions.

3.2 Literature Review

In this section we review previous research of authorship analysis with a specific focus on writeprint features. In addition, we have a brief review of different feature selection methods.

3.2.1 Authorship Analysis

Authorship analysis is a process of examining the characteristics of a piece of writing in order to draw conclusions on its authorship. Its roots are from a linguistic research area called stylometry (Holmes, 1998). It has its most extensive applications to historical literature (Mendenhall, 1887; Mosteller & Wallace, 1964). Some recent studies introduced this approach to software forensics (Gray, Sallis, & MacDonell, 1997) and online messages (de Vel, 2000; de Vel, Anderson, Corney, & Mohay, 2001) and showed promising results. This research field can be categorized into authorship identification, authorship characterization, and similarity detection.

Authorship identification, also called “authorship attribution,” determines whether a piece of work was written by a particular author by examining other works produced by that author. The origins of this field date back to the eighteenth century when English logician Augustus de Morgan suggested that authorship might be settled by determining
if one text contained more long words than another. This hypothesis was investigated by Mendenhall (1887) who subsequently published his results of authorship attribution among Bacon, Marlowe and Shakespeare. Mosteller and Wallace (1964) applied authorship attribution to solve the mystery of the Federalist Papers. Authorship characterization summarizes the characteristics of an author and generates the author profile based on his/her writings. Some of these characteristics include gender (Koppel, Argamon, & Shimoni., 2002; Argamon et al., 2003), educational and cultural background (Corney, de Vel, Anderson, & Mohay, 2002), and language familiarity or habits (Crag, 1999). Similarity detection compares multiple pieces of writing and determines whether or not they were produced by a single author without actually identifying the author. Most studies in this category are related to plagiarism detection, which attempts to detect plagiarism through examining the similarity between two pieces of writings (Clough, 2000).

This study particularly focuses on authorship identification because it is the most relevant to cybercrime investigation. The essence of authorship identification is to identify a set of features that remain relatively constant among a number of writings by a particular author. Given \( n \) predefined features, each piece of writing can be represented by an \( n \)-dimension feature vector. Various analytical techniques can be used to determine the authorship of unknown documents. Statistical analysis techniques for authorship
identification include statistical naive Bayes methods (Mendenhall, 1987), cumulative sum (CUSUM) (Farringdon, 1996), principle component analysis (PCA) (Burrows, 1987), factor analysis (Biber, 1995), cluster analysis (Holmes, 1992), discriminant analysis (Ledger & Merriam, 1994). The advent of powerful computers instigated the extensive use of machine learning techniques in authorship analysis (Tweedie, Singh, & Holmes, 1996; Lowe & Matthews, 1995; Khmelev & Tweedie, 2001; Diederich, Kindermann, Leopold, & Paass, 2000; Argamon et al., 2003). In general, machine learning methods can achieve higher accuracies than statistical methods. Many supervised learning techniques such as decision tree, neural network (NN) and support vector machine (SVM) can train and generate a classifier so as to determine label (i.e., the authorship of an article). In particular, SVM has been frequently used in previous authorship identification studies due to its advantage in handling high dimensional feature space (Diederich et al., 2000; de Vel et al., 2001; Zheng, Qin, Huang, & Chen, 2003).

The performance of authorship identification is also affected by several parameters, such as the number of candidate authors and the training sample size. Hoorn, Frank, Kowalczyk, and Ham (1999) suggested that the reliability of the classification would decrease as the number of authors increased to over three. Stamatatos Fakotakis, and Kokkinakis (2001) observed that the classification performance was improved as the number of writings by each author in the training dataset increased.
3.2.2 Writeprint Features

The writeprint features to represent authors’ writing style are of critical importance to authorship identification. Feature proposed in previous studies can be categorized into the following four types.

*Lexical features*, the earliest features used in authorship analysis, represent an author’s lexicon-related writing styles. Most lexical features are based on character or word usage in writing. For example, Elliot and Valenza (1991) conducted modal testing based on word usage to compare the poems of Shakespeare with those of Edward de Vere, the leading candidate as the true author of the works credited to Shakespeare. In Yule’s early work some more generic features were employed, such as sentence length (Yule, 1938) and vocabulary richness (Yule, 1944). Later Burrows (1987) developed a set of more than 50 high-frequency words which were tested on the Federalist Papers. Holmes (1998) analyzed the use of "shorter" words (2 or 3 letter words) and "vowel words" (words beginning with a vowel). Such features require intensive efforts in selecting the most appropriate set of words that best distinguish a given set of authors (Holmes & Forsyth, 1995). The effectiveness of this approach is limited since word usage is highly dependent on the topic of the article.

*Syntactic features*, including punctuation and function words, can capture an author’s writing style at the sentence level. They are often “content-free” features derived
from people’s personal habits of organizing sentences. In the seminal work conducted by Mosteller and Wallace (1964), they first used the frequency of occurrence of thirty function words (e.g. “while” and “upon”) to clarify the disputed work, Federalist Papers. Subsequently different function words were examined and showed good discriminating capability (Baayen, Halteren, & Tweedie, 1996; Burrows, 1989; Holmes & Forsyth, 1995; Tweedie & Baayen, 1998). Baayen et al. (1996) concluded that incorporating punctuation frequency as a feature can improve the performance of authorship identification. Stamatatos et al. (2001) introduced more complex syntactic features such as passive count and part-of-speech tags. Binongo and Smith (1999) used the frequency of occurrence of 25 prepositions to discriminate between Oscar Wilde’s plays and essays. These studies demonstrated that syntactic features might be more reliable than lexical features in authorship identification.

*Structural features*, represent the author’s habits when organizing a piece of writing. Habits such as paragraph length, use of indentation, and use of signature can be strong authorial evidence of personal writing style. These features are more prominent in online documents, which have less content information but more flexible structures or richer stylistic information in another word. Structural layout traits and other features have been introduced by de Vel et al. (2000) for email author identification and achieved high identification performance.
Content-specific features refer to keywords in a specific topic. Although seldom used in previous studies, these features could complement “content-free” features to improve the performance of authorship identification for particular applications. Usually, these features can express personal interests in a specific domain. One successful application of content-specific features was conducted by Martindale and McKenzie (1995). They correctly attributed 12 disputed Federalist Papers and concluded that content features outperformed lexical features but not function word features. Zheng et al. (2003) introduced around ten content-specific features in a cybercrime context and the results showed that they are helpful in author identification.

From a multilingual perspective, different languages may share similar writeprint features, such as structural features. However, due to the uniqueness of language, some features are not generic. For example, while most Western languages have boundaries between words, most Oriental languages do not. In addition, different languages can have different function words and word-based features.

3.2.3 Feature Selection

Various writeprint features have been developed for authorship analysis. Rudman (1998) summarized almost one thousand writeprint features for English used in authorship analysis applications. Numerous studies have shown the discriminating power of different types of features. However, not all of these features necessarily contribute to
prediction. Some features may be irrelevant or redundant, hence reducing the prediction accuracy. For instance, de Vel et al. (2001) observed a reduction in performance when the number of function word features was increased from 122 to 320. Therefore, feature selection should be undertaken to identify the key writeprint features with significant discriminating power. Since features are regarded as an abstract of authors’ writing style, feature selection is a critical issue in authorship identification.

Feature selection is aimed at identifying a subset of features that are relevant to the target concept, i.e., authorship in our case. There have been a variety of well developed feature selection methods in the pattern recognition and data mining domains (Liu & Motoda, 1998). Based on the evaluation criterion, feature selection methods are often categorized into filter models and wrapper models (Dash & Liu, 1997).

A filter model uses a certain relevance measure (e.g., correlation, information gain, and F-statistic) as a filter to screen out relevant features from all candidate features (Dash & Liu 1997; Hall, 2000; Dash & Liu, 2003). These criteria are independent of any inductive learning algorithm. Features with highest relevance scores are regarded relevant and significant. However, most filters evaluate features individually and therefore cannot capture features that are complementary to each other for class distinction. Moreover, filters also tend to retain highly relevant but redundant features.

By contrast, wrappers utilize a learning algorithm “wrapped” in the feature selection
process to evaluate feature subsets according to the prediction accuracy (Kohavi & John, 1997). Wrappers often select features with higher accuracy but are often criticized for high computational cost and low generality. The process of wrapper-based feature selection can be viewed as a search problem in feature space. Complete search and heuristic search are two major search strategies. Exhaustive search tries every feature combination to achieve the optima but is computationally infeasible for high dimensional feature spaces. Heuristic search uses certain rules to guide the direction of the search. The heuristic search strategy reduces the size of the search space and therefore speeds up the process significantly. Deterministic heuristic search algorithms, such as sequential forward selection (SFS) and sequential backward elimination, are efficient but often lead to local optimum (Pudil, Novovicova, & Kittler, 1994). Nondeterministic heuristic search algorithms, instead, seek for the globally optimal solution. For example, generic algorithm (GA) behaves like a metaphor of the processes of evolution in nature (Holland, 1975). The optimal solution with the highest fitness value can be achieved via a number of generations by applying genetic operators such as selection, crossover, and mutation. GA has been introduced to feature selection and shown good performance.

3.2.4 Research Gaps

We summarize major studies in authorship identification since 1960s in Table 3.1. We are particularly interested in two aspects: (1) what writeprint features were used, and
(2) whether the study dealt with multiple languages. We denote the four types of features
(lexical, syntactic, structural, and content-specific features) by F1, F2, F3, and F4,
respectively. In the multi-language category, Y or N indicates whether the study covered
multiple languages or not.

Table 3.1 A Summary of Previous Studies in Authorship Identification

<table>
<thead>
<tr>
<th>Previous studies</th>
<th>Features</th>
<th>Multi-language</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Mosteller &amp; Wallace, 1964)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Ledger &amp; Merriam, 1994)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Merriam &amp; Matthews, 1994)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Kjell, 1994)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Martindale &amp; McKenzie, 1995)</td>
<td>√ √ √</td>
<td></td>
</tr>
<tr>
<td>(Mealand, 1995)</td>
<td>√ √ √</td>
<td></td>
</tr>
<tr>
<td>(Holmes &amp; Forsyth, 1995)</td>
<td>√ √ √</td>
<td></td>
</tr>
<tr>
<td>(Farrington, 1996)</td>
<td>√ √ √</td>
<td></td>
</tr>
<tr>
<td>(Baayen et al. 1996)</td>
<td>√ √ √</td>
<td></td>
</tr>
<tr>
<td>(Tweedie et al., 1996)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Tweedie &amp; Baayen, 1998)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Craig, 1999)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Hoorn et al., 1999)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Binongo &amp; Smith, 1999)</td>
<td>√ √ √</td>
<td></td>
</tr>
<tr>
<td>(Diederich et al. 2000)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(De Vel et al., 2001)</td>
<td>√ √ √ √</td>
<td></td>
</tr>
<tr>
<td>(Stamatatos et al., 2001)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Khmelev &amp; Tweedie, 2001)</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>(Corney et al., 2002)</td>
<td>√ √ √ √</td>
<td></td>
</tr>
<tr>
<td>(Baayen et al., 2002)</td>
<td>√ √ √ √</td>
<td></td>
</tr>
<tr>
<td>(Peng et al., 2003)</td>
<td>√ √ √</td>
<td></td>
</tr>
<tr>
<td>(Zheng et al., 2003)</td>
<td>√ √ √ √</td>
<td></td>
</tr>
<tr>
<td>(Argamon et al., 2003)</td>
<td>√ √ √ √</td>
<td></td>
</tr>
</tbody>
</table>
From Table 3.1, we have the following observations. First, lexical and syntactic features were most commonly used in previous studies, while the effects of structural and content-specific features have not been fully investigated. Second, most studies only focused on authorship analysis in one particular language. Few researchers have addressed the multilingual issues in authorship identification.

Although authorship identification methods have achieved successes in many literary and forensic applications, very limited studies have been undertaken specifically on online messages. The unique characteristics of online messages have brought us with some new challenges in authorship identification. First, online messages are often shorter than conventional documents. As Ledger and Merriam (1994) claimed, authorship characteristics would not be strongly apparent below 500 words. Forsyth and Holmes (1996) found that it was very difficult to attribute a text of less than 250 words to an author. The short length of online messages may cause some good features in normal texts to be ineffective. For example, since the vocabulary used in short documents is usually limited and relatively unstable, measures such as vocabulary richness may be not as effective as in previous studies on literary works. Second, online messages also have some special characteristics which may help reveal the writing style of the author. Since Web-based channels are relatively more informal and casual, authors are more likely to leave their own “writeprints” in their articles. For example, the structure or composition
style used in online messages is often different from normal text documents. Some features, such as structural layout traits, unusual language usage, and unusual content markers, may contribute more to author distinction. Third, since the Internet is a global network, cyber users can distribute online messages in any language over the cyber space. We need to take into account the multilingual nature of online messages.

3.3 Research Questions

Considering these challenges brought by the unique characteristics of online messages, we need to examine the effectiveness of different writeprint features and identify the key features for authorship identification of online messages. Specifically, this study focuses on the following research questions:

Q1: How can we apply authorship identification to online messages in a multilingual context?

Q2: Which types of writeprint features are effective for authorship identification of online messages?

Q3: How can we identify a set of important writeprint features for authorship identification?

Q4: What are the key writeprint features for authorship identification of online messages in different languages?
3.4 Authorship Identification Based on Writeprint Features

In this section we develop a framework for authorship identification of online messages (Figure 3.1). Specifically, we describe four types of writeprint features to represent online authors’ writing styles and a GA-based feature selection method to identify the key features.

![Figure 3.1 A Framework of Authorship Identification of Online Messages](image-url)
3.4.1 A Framework for Authorship Identification of Online Messages

In our overall framework, the process of authorship identification can be divided into the following four major steps: message collection, feature extraction, model generation, and author identification.

- **Step 1: Message Collection**

  In order to capture the writing styles of online authors, we first need a collection of their previous messages. For each active author of interest, a certain number of messages written by this author can be retrieved from the Web channels such as email, newsgroup, forums, and chat rooms.

- **Step 2: Feature Extraction**

  Online messages collected from the Web are in unstructured text format. Based on related works and the uniqueness of online messages, we develop a set of writeprint features to represent the authors’ writing styles. These writeprint features will be described in detail in next section. We developed a feature extractor to automatically extract writeprint features from each online message. Thus, each message is converted from textual format into a vector of writeprint features.

- **Step 3: Model Generation**

  As in a typical classifier learning process, the online message collection is divided into two subsets. The first subset, called training set, is used to train the classification
model using an inductive learning algorithm. The other subset is called testing set, which is used to validate the prediction power of the authorship identification model generated by the classification model. By comparing the prediction with the true authorship of the messages in testing set, we can estimate the quality of the classification model.

- Step 4: Author Identification

After the authorship identification model is validated to be effective, it can be used to predict the authorship of unknown online messages. The result of authorship identification will help investigators narrow down their scope to a small number of potential authors.

3.4.2 Writeprint Features

As an important component of our framework, the writeprint features may significantly affect the performance of authorship identification. By following previous studies and considering the unique characteristics of online messages, we define a feature set of four types of writeprint features: lexical, syntactic, structural, and content-specific features.

3.4.2.1 Writeprint Features for English

For English messages, we developed 270 writeprint features: 87 lexical features, 158 syntactic features, 14 structured features, and 11 content-specific features (Table 3.2):
Table 3.2 Writeprint Features for English Messages

<table>
<thead>
<tr>
<th>Features</th>
<th>Descriptions/Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lexical features</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Character-based features</strong></td>
<td></td>
</tr>
<tr>
<td>1. Total number of characters (C)</td>
<td></td>
</tr>
<tr>
<td>2. Total number of alphabetic characters /C</td>
<td></td>
</tr>
<tr>
<td>3. Total number of upper-case characters/C</td>
<td></td>
</tr>
<tr>
<td>4. Total number of digit characters/C</td>
<td></td>
</tr>
<tr>
<td>5. Total number of white-space characters/C</td>
<td></td>
</tr>
<tr>
<td>6. Total number of tab spaces/C</td>
<td></td>
</tr>
<tr>
<td>7 ~ 32. Frequency of letters (26 features)</td>
<td>A ~ Z</td>
</tr>
<tr>
<td>33 ~ 53. Frequency of special characters (21 features)</td>
<td>~ , @, #, $, %, ^, &amp;, *, -, _, = ,+, &gt;, &lt;, [ , ], { , }, /, ,</td>
</tr>
<tr>
<td><strong>Word-based features</strong></td>
<td></td>
</tr>
<tr>
<td>54. Total number of words (M)</td>
<td></td>
</tr>
<tr>
<td>55. Total number of short words (less than four characters) /M</td>
<td>e.g., and, or</td>
</tr>
<tr>
<td>56. Total number of characters in words/C</td>
<td></td>
</tr>
<tr>
<td>57. Average word length</td>
<td></td>
</tr>
<tr>
<td>58. Average sentence length in terms of character</td>
<td></td>
</tr>
<tr>
<td>59. Average sentence length in terms of word</td>
<td></td>
</tr>
<tr>
<td>60. Total different words/M</td>
<td></td>
</tr>
<tr>
<td>61. Hapax legomena *</td>
<td>Frequency of once-occurring words</td>
</tr>
<tr>
<td>62. Hapax dislegomena *</td>
<td>Frequency of twice-occurring words</td>
</tr>
<tr>
<td>63. Yule’s K measure *</td>
<td>A vocabulary richness measure defined by Yule</td>
</tr>
<tr>
<td>64. Simpson’s D measure *</td>
<td>A vocabulary richness measure defined by Simpson</td>
</tr>
<tr>
<td>65. Sichel’s S measure *</td>
<td>A vocabulary richness measure defined by Sichele</td>
</tr>
<tr>
<td>66. Brunet’s W measure *</td>
<td>A vocabulary richness measure defined by Brune</td>
</tr>
<tr>
<td>67. Honore’s R measure *</td>
<td>A vocabulary richness measure defined by Honore</td>
</tr>
<tr>
<td>68 ~ 87. Word length frequency distribution /M (20 features)</td>
<td>Frequency of words in different length</td>
</tr>
<tr>
<td><strong>Syntactic Features</strong></td>
<td></td>
</tr>
<tr>
<td>88 ~ 95. Frequency of punctuations (8 features)</td>
<td>“” “” “” “” “” “” “” “”</td>
</tr>
<tr>
<td>96 ~ 245. Frequency of function words (150 features)</td>
<td>The list of function words is in Appendix A.</td>
</tr>
<tr>
<td><strong>Structural Features</strong></td>
<td></td>
</tr>
<tr>
<td>246. Total number of lines</td>
<td></td>
</tr>
<tr>
<td>247. Total number of sentences</td>
<td></td>
</tr>
<tr>
<td>248. Total number of paragraphs</td>
<td></td>
</tr>
<tr>
<td>249. Number of sentences per paragraph</td>
<td></td>
</tr>
<tr>
<td>250. Number of characters per paragraph</td>
<td></td>
</tr>
<tr>
<td>251. Number of words per paragraph</td>
<td></td>
</tr>
<tr>
<td>252. Has a greeting</td>
<td></td>
</tr>
<tr>
<td>253. Has separators between paragraphs</td>
<td></td>
</tr>
<tr>
<td>254. Has quoted content</td>
<td>Cite original message as part of replying message</td>
</tr>
<tr>
<td>255. Position of quoted content</td>
<td>Quoted content is below or above the replying body</td>
</tr>
<tr>
<td>256. Indentation of paragraph</td>
<td>Has indentation before each paragraph</td>
</tr>
<tr>
<td>257. Use Email as signature</td>
<td></td>
</tr>
<tr>
<td>258. Use telephone as signature</td>
<td></td>
</tr>
<tr>
<td>259. Use url as signature</td>
<td></td>
</tr>
<tr>
<td><strong>Content-specific Features</strong></td>
<td></td>
</tr>
<tr>
<td>260~270. Frequency of content specific keywords (11 features)</td>
<td>deal, obo, sale, wtb, thx, paypal, check, windows, software, offer, Microsoft</td>
</tr>
</tbody>
</table>

The definitions of measures with “*” can be found in (Tweedie & Baayen, 1998).
Lexical Features:

Lexical features can be further divided into character-based and word-based features. In this research, we included character-based lexical features used in (de Vel, 2000; Forsyth & Holmes, 1996; Ledger & Merriam, 1994), vocabulary richness features in (Tweedie & Baayen, 1998), and word length frequency features used in (Mendenhall, 1887; de Vel et al., 2000). In total, we adopted 87 lexical features for English messages.

Syntactic Features:

Syntactic features, including function words, punctuation, and part of speech, can capture an author’s writing style at the sentence level. The discriminating power of syntactic features is derived from people’s different habits of organizing sentences. We do not use POS tags as features in this study because POS tagging is still immature for some languages such as Chinese.

Different sets of function words, ranging from 12 to 122, have been tested in various studies (Baayen et al., 1996; Burrows, 1989; Holmes & Forsyth, 1995; Tweedie & Baayen, 1998; de Vel et al., 2001). There is no generally accepted good set of function words for authorship identification because of the varying discriminating power of function words in different applications. In this research, we adopted a set of 150 function words, which is selected based on previous research (see Appendix B).

We also adopted the punctuation features suggested by Baayen et al. (1996).
Combing the function words and punctuation features, we considered 158 syntactic features for English online messages in our framework.

- **Structural Features:**

  In general, structural features represent the way an author organizes the layout of an article. De Vel (2000) introduced several structural features specifically for email. Since email contains many general structural features of online messages, we adopted those features applicable for online messages. In addition, we added features such as paragraph indentation and signature-related features. In total, we adopted 14 structured features, including ten features from (de Vel et al., 2001) and four new features.

- **Content-specific Features:**

  In addition to “content-free” features, content-specific features are important discriminating features for online messages. The selection of such features is dependent on specific application domains. On the Web one user may often post online messages involving a relatively small range of topics whereas different users may distribute messages on different topics. For this reason, special words or characters closely related to specific topics may provide some clue about the identity of the author. For example, a criminal selling pirate software may use such words as “obo” (or best offer), “for sale,” etc. When applied to a different application domain, different content-specific features need to be developed based on specific application characteristics. For example, based on
suggestions from domain experts and manual inspection of historical messages, we examine eleven keywords as content-specific features specifically for English “for-sale” online messages. These keywords either represent characteristics of the materials for sale or the writing habits of the author when he/she was selling something online.

3.4.2.2 Writeprint Features for Chinese

Although some languages share similar style features, many languages exhibit unique characteristics, such as function words. Most Western languages use a blank space as a boundary to segment two words. By contrast, in many Oriental languages such as Chinese, such word boundaries often do not exist and words are adjacent to each other in a sentence. These require different writing style features for different languages.

We investigate the writing style features in Chinese in attempt to examine the capability of our framework in a multiple language context. The reason why we choose Chinese because Chinese is the second most popular language on the web (14.1%, English is 35.8%, http://www.glreach.com/globstats/). Also, Chinese is a typical oriental language that differs from English at some important aspects (e.g., word boundary). Due to the language differences, some English features (e.g., word length or frequency of the 26 different English letters) do not exist in Chinese. We remove such features. Based on analysis of comparable English features, 117 features from Table 3.2 are selected for Chinese online messages, including 16 lexical features (Features 1 ~ 6, Features 54 and
Features 59 ~ 67), 77 syntactic features (Features 88 ~ 95 and 69 Chinese function words, see Appendix B), 14 structural features (Features 246 ~ 259), and 10 content-specific features.

3.4.3 A GA-based Feature Subset Selection Model

Not all of these features are relevant or informative to authorship identification. In this study we use a GA-based feature selection model to identify writeprint features. We chose a feature subset selection method because it can capture the complementary effects among features. A genetic algorithm (GA) is used to search the feature space and generate candidate feature subsets (Holland, 1975).

In such a model each chromosome represents a feature subset, where its length is the total number of candidate features and each bit indicates whether a feature is selected or not. Specifically, 1 represents a selected feature while 0 represents a discarded one. For example, a chromosome representing five candidate features, “10011,” means that the first, fourth and fifth features are selected, while the other two are discarded. In the first generation each bit of chromosomes is assigned to 0, meaning that none of the features is selected. Each chromosome, i.e., a feature subset, can be employed to train a classifier. Thus, the fitness value of each chromosome is defined as the accuracy of the corresponding classifier. By applying genetic operators in the successive generations, the GA model can generate different combinations of features to achieve the highest fitness
value. Therefore, the feature subset corresponding to the highest accuracy of classification along all the generations is regarded as the optimum. The selected features in this subset are the key writeprint features to discriminate the writing styles of different authors. The process of this GA-based feature selection is shown in Figure 3.2.

![Figure 3.2 GA-based Feature Subset Selection](image)

3.5 Experimental Study

In our experimental study, we examined the performance of authorship identification on both English and Chinese corpora. In particular, we are interested in evaluating the effectiveness of different writeprint features for authorship distinction.
3.5.1 Data Descriptions

To test the feasibility of the authorship identification and to identify the key writeprint features for online messages, two test-beds of online messages (English and Chinese) were created. Among various types of online messages, personal emails and chat messages often involve privacy issues and are difficult to collect. Therefore, we collected publicly available newsgroup messages as the test-bed in our experiments. Since illegal online messages are of particular interest in this study, we collected messages that were involved in selling pirated software/CDs from misc.forsale.computers.* (including 27 sub-groups) in Google newsgroups to create the English test-bed. The Chinese test-bed is composed of online messages from the two most popular Chinese Bulletin Board Systems (smth.org and mitbbs.com), involving seven different topics (e.g., movie, music, and novel). For each of the test-beds, we identified ten of the most active authors with 30-40 messages collected for each of them. The average length of the messages written by each author is 169 words for the English test-bed and 807 characters for the Chinese test-bed.

3.5.2 Evaluation Metric

To evaluate the performance of authorship identification, we used the classification accuracy as the evaluation metric, which has been commonly used in data mining and authorship analysis. Accuracy indicates the overall prediction correctness of a classifier.
In our study accuracy can be defined as follows:

\[
\text{Accuracy} = \frac{\text{number of messages whose author was correctly identified}}{\text{total number of messages}}
\]

3.5.3 Experimental Design

Many classification techniques can be adopted to perform the authorship identification task. In this research, we chose a support vector machine (SVM) classifier (Cristianini & Shawe-Taylor, 2000). SVM is a novel learning machine first introduced by Vapnik (1995). It is based on the Structural Risk Minimization principle from the computational learning theory. Due to its capability of handling high dimensional feature space and good performance reported in various applications (Joachims, 2002; Cristianini & Shawe-Taylor, 2000), it has been introduced to authorship analysis research (Diederich et al., 2000; de Vel et al., 2001; Argamon, 2003). In this study we choose the WEKA data mining package, which implements the sequential minimal optimization (SMO) algorithm to train a SVM classifier (Platt, 1998; Keerthi, Shevade, Bhattacharyya, & Murthy, 2001).

In this study we designed two experimental studies to examine the discriminating power of different writeprint features. The first experiment focuses on comparing the different types of features, whereas the second is aimed to identify the more important features for authorship identification.
3.5.3.1 Comparing Different Types of Features

To examine different features and techniques, we designed several author identification tasks. Firstly, four feature sets were created. Here, we use F1, F2, F3 and F4 to denote lexical, syntactic, structural, and content-specific features, respectively. The first feature set contained lexical features (F1) only. Syntactic features were added to F1 to form the second feature set (F1+F2). Structural features were added to form the third feature set (F1+F2+F3). The fourth feature set contains all the four types of features (F1+F2+F3+F4). The reason why we choose this incremental method in such order is that the order represents the evolutionary sequence of style features and we intend to examine the effect of adding relatively new features on top of existing ones. For each feature set, we trained an SVM classifier to authorship identification. A 10-fold cross-validation was used to estimate the accuracy of the classification model. The same procedure was repeated for both the English and Chinese test-beds.

3.5.3.2 Writeprint Feature Selection

In addition to examine the effectiveness of the four types of writeprint features, we are also interested in which are the key features for authorship identification of online messages. We applied the GA-based feature selection model on both the English and Chinese test-beds. The selection process started with an empty feature set, i.e., no feature was used. In the successive generations, the GA model conducted a global search for the
optimal feature subset by applying the selection, crossover, and mutation operators. The classification accuracy of the SVM classifier was used as the fitness value in the GA-based feature selection process. We compared the classification accuracy of the full feature set and the selected feature subset.

3.5.4 Experimental Results and Discussion

3.5.4.1 Comparing Different Types of Features

The results for the comparison of different feature types and techniques are summarized in Table 3.3. We observed that the accuracy increased as more types of features were incorporated. For both the English and Chinese test-beds, the best accuracy was achieved when all feature types were used. Our experiments showed that these writeprint features can distinguish authors of online messages with satisfactory accuracy. In particular, when all features were used, the SVM classifier achieved accuracy of 97.69% and 88.33% for the English and Chinese test-beds, respectively. Several factors may account for the discrepancy of performance between the two languages. Our writeprint features for Chinese were not as comprehensive as those for English. Better predictions would be achieved if more Chinese features were developed. Despite the differences in English and Chinese, our proposed approach appears promising in a multilingual context.
Table 3.3 Accuracy of Four Writeprint Feature Subsets

<table>
<thead>
<tr>
<th>Feature sets</th>
<th>English</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>89.36%</td>
<td>57.78%</td>
</tr>
<tr>
<td>F1+F2</td>
<td>90.03%</td>
<td>69.16%</td>
</tr>
<tr>
<td>F1+F2+F3</td>
<td>94.66%</td>
<td>82.77%</td>
</tr>
<tr>
<td>F1+F2+F3+F4</td>
<td>97.69%</td>
<td>88.33%</td>
</tr>
</tbody>
</table>

Furthermore, we conducted pair-wise t-tests to examine the effect of adding one type of features on the accuracy of authorship identification for both languages. Table 3.4 summarizes the $t$-stat and p-values of each comparison for the English and Chinese test-beds. Significant differences in pair-wise $t$-tests are highlighted in bold ($\alpha = 0.05$).

Table 3.4 Pair-wise t-tests on Accuracy of Different Feature Subsets

<table>
<thead>
<tr>
<th>Feature Types</th>
<th>English</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 &lt; F1+F2</td>
<td>0.4005</td>
<td>0.0176</td>
</tr>
<tr>
<td>F1+F2 &lt; F1+F2+F3</td>
<td>0.0282</td>
<td>0.0068</td>
</tr>
<tr>
<td>F1+F2+F3 &lt; F1+F2+F3+F4</td>
<td><strong>0.0277</strong></td>
<td>0.1274</td>
</tr>
</tbody>
</table>

- **Lexical Features (F1):**

When using lexical features alone, a SVM classifier achieved 89.36% accuracy for the English test-bed. Although for online messages of short length, some lexical features such as vocabulary richness may not be as useful as for normal documents (Tweedie & Baayen, 1998), we believe that most lexical features are still promising to identifying
authorship of English messages. For the Chinese test-bed, the SVM classifier achieved 57.78% accuracy when using lexical features alone. This indicates that lexical features are not effective enough for authorship identification of Chinese messages. More features need to be used to improve the performance.

- Syntactic Features (F2):

  For the English test-bed, when the syntactic features were added, the classification accuracy was improved but not significantly (p=0.4005). This result did not fully confirm the good discriminating power of function words reported in previous studies. Online messages in our test-bed were too short to represent authors’ usage of function words. Moreover, compared with the small number of words used in one message, the number of function words in our feature set may be too large and some of them could be irrelevant or redundant. Some previous studies have examined the impact of the number of function words for authorship identification (Stamatatos et al., 2001; de Vel et al., 2001). Having too many function words could decrease the prediction accuracy. Further experiments need to be conducted to identify a suitable set of function words for online messages. For the Chinese test-bed, adding syntactic features improved the performance significantly (p=0.0176). Since the average length (807 words) of the Chinese messages is longer than that of the English messages (74 words), it seems that syntactic features are more effective in distinguishing authorship of longer messages.
• Structural Features (F3):

For both the English and Chinese test-beds, adding structural features improved the performance significantly (p=0.0282 for English; p=0.0068 for Chinese). These results demonstrate that structural features are effective discriminators for authors of online messages. For example, some authors like to use long sentences or paragraphs, while some authors like to have their signature at the end of their messages.

• Content-specific Features (F4):

Content-specific features also improved the classification accuracy significantly for the English test-bed (p=0.0277) but not for Chinese (p=0.1274). Authors seem to have certain content-specific keywords in their messages. For example, some preferred check as the payment method and some people mostly sell Microsoft products.

3.5.4.2 Writepint Feature Selection

The comparative experiments on different types of features demonstrated the distinguishing power of each feature type. Furthermore, we used the GA-based feature subset selection model to identify key writeprint features for online messages. The GA-based feature selection process starts from an empty set. We observed a significant increase of fitness value as features were incorporated at several early generations and a relatively constant accuracy afterwards. The number of selected features in the best chromosome (i.e., subset) of each generation increased until stabilized around half of the
full feature set.

Figure 3.3 shows the change of accuracy and the number of selected features along 500 generations for the two test-beds. The evolutionary process of GA converged after about 50 generations for the English dataset and 120 generations for the Chinese dataset. Among all the chromosomes in the 500 generations, the one with the highest accuracy corresponded to the optimal feature subset.

Figure 3.3 Experiment Results of Feature Selection on English and Chinese Test-beds

To compare the discriminating power of the full feature set and the optimal set,
10-fold pair-wise $t$-tests were conducted respectively for the English and Chinese datasets. As shown in Table 3.5, the GA-based model identified a feature subset with only about half of the full set as the key features, i.e., 134 out of 270 for English, and 56 out of 117 for Chinese. For the English dataset, the optimal feature set achieved a classification accuracy of 99.01%, which is significantly higher than 97.96% achieved by the full set ($p=0.0417$). For the Chinese dataset, the optimal feature set achieved a classification accuracy of 90.56%, which is higher than 88.33% achieved by the full set but not significantly ($p=0.1270$). In general, using the optimal feature subset, we can achieve a comparable (if not higher) accuracy of authorship identification.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Feature set</th>
<th># of Features</th>
<th>Accuracy</th>
<th>Variance</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>Full set</td>
<td>270</td>
<td>97.96%</td>
<td>0.002</td>
<td>0.0417</td>
</tr>
<tr>
<td></td>
<td>Optimal subset</td>
<td>134</td>
<td>99.01%</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>Full set</td>
<td>117</td>
<td>88.33%</td>
<td>0.023</td>
<td>0.1270</td>
</tr>
<tr>
<td></td>
<td>Optimal subset</td>
<td>56</td>
<td>90.56%</td>
<td>0.026</td>
<td></td>
</tr>
</tbody>
</table>

The effect of feature selection is significant and promising. Furthermore, we discovered that the selected key feature subset included all four types of features. This also confirmed that each type of features contributes to the predictive power of the classification model. In particular, the relatively high proportion of selected structural and
content-specific features suggests their useful discriminating power for online messages. Table 3.6 illustrates several key features identified from the full feature set.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>English</th>
<th>Chinese</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lexical</td>
<td>Total number of upper-case letters /total number of characters; Frequency of character “@” and “$”; Yule’s K measure (vocabulary richness); 2-letter word frequency.</td>
<td>Total number of English characters /total number of characters; Total number of digits /total number of characters; Honore’s R measure (vocabulary richness).</td>
</tr>
<tr>
<td>Syntactic</td>
<td>Frequency of punctuation “!” and “:”; Frequency of function word “if” and “can”</td>
<td>Frequency of function word “然后(then)” and “我想(I think)”</td>
</tr>
<tr>
<td>Structural</td>
<td>Number of sentences per paragraph; Has separators</td>
<td>Number of sentences per paragraph; Has separators</td>
</tr>
<tr>
<td>Content-specific</td>
<td>Frequency of word “check” and “sale”</td>
<td>Frequency of “音乐(music)” and “小说(novel)”</td>
</tr>
</tbody>
</table>

The results from Table 3.6 have some interesting implications. Since some features in the full feature set may be irrelevant for online messages, the frequency of characters related to online messages (e.g., “@,” “$”) instead of other common ones (e.g., “A,” “E”) were selected. In addition, since some features may only provide redundant information, the total number of upper-case letters/ total number of characters was identified as a key
feature, while the frequency of lower-case letters was discarded. Similarly, only one vocabulary richness measure, e.g., Yule’s K or Honore’s R, was selected and others were ignored. Since online messages are often short in length and flexible in style, structural layout traits such as the average length of paragraphs became more useful. In addition, content-specific features are highly related to their context. Hence features such as “sale” and “check” were identified as the key content-specific features for the English dataset based on sales of pirated software/CDs. In other contexts, different content-specific features should be identified and used accordingly.

These selected key features of writeprint can effectively represent the distinct writing style of each author and further assist us to identify the authorship of new messages. Figure 3.4 exemplifies a comparison of writeprints between three authors in the English dataset using five of the key features, where feature values were normalized to [0, 1]. Clearly, Mike’s distinct writeprint from the other two indicates his unique identity. The high similarity between the writeprints of Joe and Roy suggests that these two IDs might be the same person.
Figure 3.4 Comparison of Writeprints between Three Authors

3.6 Conclusions

The absence of fingerprints in cyberspace leads law enforcement and intelligence community to seek for new approaches to trace criminal identity in cybercrime investigation. To address this problem, we developed a framework for authorship identification of online messages. We defined four types of writeprint features and introduced a GA-based feature selection model to identify the key features. We conducted experiments on English and Chinese online messages to examine these writeprint features. The experimental results demonstrated the capability of our approach to distinguish authors of online messages. The satisfactory performance we achieved for both English and Chinese test-beds showed a promising application of this approach in a multilingual context. Structural features and content-specific features showed particular discriminating power for online messages. The key writeprint features identified by the GA-based model
could achieve higher classification accuracy in distinguishing online authors. We believe the proposed authorship identification framework can potentially assist identity tracing in the cybercrime investigation.

Several interesting issues in this research domain are still open. Given the key features selected, we will continue to rank and cluster them based on their functional traits, and further provide a visual representation of an author’s writeprint. Due to the multinational nature of cybercrime, we plan to employ this feature selection model to identify the key writeprint features in other languages. In addition, we are also interested in applying the writeprint identification approach to other related problems such as plagiarism detection and intellectual property checking.
CHAPTER 4: FEATURE SELECTION: OPTIMAL SEARCH-BASED FEATURE SUBSET SELECTION FOR CANCER CLASSIFICATION

4.1 Introduction

Chapters 2 and 3 present two studies on constructing informative features from structured and unstructured data for knowledge discovery. However, in many applications, a large amount of features does not necessarily lead to better performance. Chapter 3 has shown that feature selection can improve prediction accuracy by removing irrelevant and redundant information. In this chapter I develop a framework of feature selection for classification. Specifically, I examine optimal search-based feature subset selection methods for gene array cancer classification.

Knowledge discovery in bioinformatics is largely driven by the available biological experimental data and knowledge. Cancer research is one of the major research areas in the medical field. An accurate classification of different tumor types is of great importance in cancer diagnosis and drug discovery (Lu & Han, 2003). Traditional cancer classification has always been based on morphological and clinical measurements, but these methods are limited in their diagnostic ability. In order to achieve more accurate cancer classification, researchers have proposed approaches based on global genetic data analysis.

Cells are the fundamental working units for every living system. The
deoxyribonucleic acid (DNA) contains all the instructions to direct cell activities. The entire DNA sequence that codes for a living thing is called its genome. A gene is a defined section of the entire genomic sequence, with a specific and unique purpose. DNA acts as a blueprint for a molecule called ribonucleic acid (RNA). The process of transcribing a gene’s DNA sequence into RNA is called gene expression. The expression of a gene provides a measure of gene activity under certain biochemical conditions. It is known that certain diseases, such as cancer, are reflected in the changes of the expression values of certain genes. Due to some practical problems of gene expression, such as the instability of messenger RNA, biomedical researchers also study the relationships between diseases and DNA methylation as an alternative. Methylation is a modification of cytosine that occurs either with or without a methyl group attached. This methylation of cytosine can only appear together with guanine as CpG. The methylated CpG can be seen as a fifth base and is one of the major factors responsible for expression regulation.

The advent of DNA microarray techniques has supported effective identification of different gene functions for cancer diagnosis. Microarray analysis is a relatively new molecular biology methodology that expands on classic probe hybridization methods to provide access to thousands of genes at once, therefore allowing the simultaneous measuring of thousands of genes in a cell sample. With this abundance of gene array data, biomedical researchers have been exploring their potential for cancer classification and
have seen promising results.

For microarray-based cancer classification, the input variables are measurements of genes and the outcomes are tumor class. Let $X_1, X_2, \ldots, X_n$ be random variables for genes $g_1, g_2, \ldots, g_n$, respectively. In particular, $X_i$ represents the expression or methylation level of gene $g_i$. Let $C$ be the random variable for the class labels, and domain($C$) = \{1, \ldots, K\}, where $K$ denotes the total number of classes. Given a training set $T = \{(t_1, c_1), (t_2, c_2), \ldots, (t_m, c_m)\}$ of $m$ tuples, where $t_i = \{t_i.X_1, t_i.X_2, \ldots, t_i.X_n\}$ is a set of measurement of $n$ genes in one sample and $c_i$ is its corresponding class label. Microarray-based cancer classification is to construct a classification model from such a training set and to predict the class label of unknown samples in the test set $S = \{t_1, t_2, \ldots, t_l\}$, where $l$ is the size of the test set. The classification accuracy is defined as the percentage of correct predictions made by the classifier on the test set. Various classification algorithms can be applied to cancer classification problems, ranging from the decision-tree method, the linear discrimination analysis, the nearest neighbor analysis, to the new support vector machines (SVMs). However, the unique nature of microarray data poses some new challenges to cancer classification.

The major problem of all classification algorithms for gene expression and methylation analysis is the high dimensionality of input space compared to the relatively small number of available samples (Model, Adorján, Olek, & Piepenbrock, 2001). There
are usually thousands to hundreds of thousands in each sample and fewer than one hundred samples. Most classification algorithms may suffer from the high dimensionality due to overfitting. The problem of learning from few data samples in a high-dimensional feature space is often known as the “curse of dimensionality” (Bellman, 1961). In addition, the large number of features would increase the computational cost significantly. Another challenge arises from irrelevant genes. Among the huge number of genes in microarray data, cancer-related genes only occupy a small portion while most of the genes are irrelevant for cancer distinction. Not only does the presence of these irrelevant genes interfere with the discrimination power of the relevant ones but also incur the difficulty in gaining insights about genes’ biological relevance. A critical concern for biomedical researchers is to identify the marker genes that can discriminate tumors for cancer diagnosis. Therefore, gene selection is of great importance for accuracy and interpretability of microarray-based cancer classification.

The remainder of this chapter is organized as follows. In Section 4.2 we review existing feature selection methods and their applications to gene selection. In Section 4.3 we raise our research questions to address the limitations in previous studies. In Section 4.4 we develop a framework of optimal search-based feature subset selection. Under this framework, in Section 4.5 we present a comparative study of several gene selection methods applied to two real microarray datasets. We conclude in Section 4.6 by
summarizing key insights and future directions.

4.2 Literature Review

Identification of good marker genes for cancer diagnosis is a typical feature selection problem. In this section we survey different feature selection techniques and their applications for gene array data.

4.2.1 Feature Selection

Feature selection is aimed at identifying a minimal-sized subset of features that are relevant to the target concept (Dash & Liu, 1997). The objective of feature selection is three-fold: improving the prediction accuracy, providing faster and more cost-effective prediction, and providing a better understanding of the underlying process that generated the data (Guyon & Elisseeff, 2003).

A feature selection method generates different candidates from the feature space and assesses them based on some evaluation criterion to find the best feature subset (Dash & Liu, 1997). Based on the evaluation criterion and generation procedure of candidates, we can categorize various feature selection methods into a taxonomy as shown in Table 4.1, which includes examples of the major feature selection methods in each category. We introduce these methods in detail below.
### Table 4.1 Taxonomy of Feature Selection

<table>
<thead>
<tr>
<th>Evaluation Criterion</th>
<th>Model</th>
<th>Measure</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Filter</td>
<td>Distance - the degree of separation between classes.</td>
<td>Fisher criterion (Bishop, 1995), test statistics (Mendenhall &amp; Sincich, 1995), Relief (Kira &amp; Rendell, 1992)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consistency - finds a minimum number of features that can distinguish classes.</td>
<td>Inconsistency rate (Dash &amp; Liu, 2003)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Correlation - measures the ability to predict one variable from another.</td>
<td>Pearson correlation coefficient (Hall, 2000), information gain (Quinlan, 1993)</td>
</tr>
<tr>
<td></td>
<td>Wrapper</td>
<td>Classification - the performance of an inductive learning algorithm.</td>
<td>Decision tree &amp; naïve Bayes (Kohavi &amp; John, 1997)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generation Procedure</th>
<th>Type</th>
<th>Search</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Individual Ranking</td>
<td>Measures the relevance of each individual feature.</td>
<td>Most filters (Guyon &amp; Elisseeff, 2003)</td>
</tr>
<tr>
<td></td>
<td>Subset Selection</td>
<td>Complete - traverses all the feasible solutions.</td>
<td>Branch and bound (Chen, 2003), best-first search (BFS) (Ginsberg, 1993)</td>
</tr>
<tr>
<td></td>
<td>Heuristic Deterministic – uses a greedy strategy to select feature according to local change.</td>
<td>Sequential forward selection (SFS), sequential backward selection (SBS), sequential floating forward selection (SFFS), sequential floating backward selection (SFBS) (Pudil et al., 1994)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Non-deterministic - attempts to find the optimal solution in a random fashion.</td>
<td>Simulated annealing (SA) (Kirkpatrick et al. 1983), Las Vegas Filter (LVF) (Liu &amp; Setiono, 1996), genetic algorithms (GA) (Holland, 1975), tabu search (TS) (Glover &amp; Laguna, 1999)</td>
<td></td>
</tr>
</tbody>
</table>
4.2.1.1 Evaluation Criterion

An evaluation criterion is used to measure the discriminating ability of candidate features. Based on the evaluation criterion, feature selection methods can be divided into filter models and wrapper models (Kohavi & John, 1997). A filter model selects good features as a preprocessing step without involving any learning algorithm. In contrast, a wrapper model utilizes a learning algorithm as a black box “wrapped” in the feature selection process to score feature subsets according to the prediction accuracy.

4.2.2.1.1 Filter Models

Filter models select good features based on a certain data intrinsic measure (Dash & Liu, 1997; Hall, 2000; Dash & Liu, 2003). These measures show the relevance of a feature to the target class. These relevance measures can be further grouped into distance, consistency, and correlation measures (Dash & Liu, 1997).

- Distance Measures

Distance measures attempt to quantify the ability of a feature or a feature subset to separate different classes from each other. A classical distance measure to assess the degree of separation between two classes is given by the Fisher criterion (Bishop, 1995):

\[ J(k) = \frac{(\mu'_k - \mu''_k)^2}{\sigma'^2_k + \sigma''^2_k} \]

where \((\mu'_k, \mu''_k)\) is the mean and \((\sigma'^2_k, \sigma''^2_k)\) is the standard deviation of all features in
instances within class I and II, respectively. It gives a high score for features where the
two classes are far apart compared to the within class variances. In addition, based on the
assumption of normal distribution of feature values within a class, test statistics can also
be used as a distance measure. A $t$-statistic value can measure the significance of the
difference between two class means (Mendenhall & Sincich, 1995). Similarly, F-statistic
and $\kappa^2$-statistic can be used for multi-class problems. A well-known feature selection
algorithm, Relief, also uses a distance measure that estimate the ability of a feature to
distinguish two instances that are close to each other but of opposite classes (Kira &
Rendell, 1992).

- Consistency Measures

Consistency measures attempt to find a minimum number of features that can
distinguish classes as consistently as a full feature set (Dash & Liu, 2003). An
inconsistency is defined as two instances having the same pattern: the same feature
values, but different class labels. Since the full feature set always has the lowest
inconsistency rate, feature selection thus attempt to minimize the number of features in
subset $S$ to achieve a certain inconsistency rate. Consistency measures have been shown
effective in removing these undesired features. However, these measures are often limited
to discrete data to make the patterns of feature values countable. Furthermore, finding the
best feature subset based on consistency measures is very computationally costly (Dash
Correlation Measures

Correlation measures quantify the ability to predict the value of one variable from the value of another (Dash & Liu, 1997). These correlation measures are often based on linear correlation or information theory.

Under the first type of linear correlation-based measures, the most well known measure is the *Pearson correlation coefficient*. For two continuous variables \(X\) and \(Y\), the correlation coefficient is defined as:

\[
r_{XY} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2} \sqrt{\sum_i (y_i - \bar{y})^2}}
\]

where \(\bar{x}\) (or \(\bar{y}\)) is the mean of variable \(X\) (or \(Y\)). This coefficient can also be extended to measure the correlation between a continuous variable and a discrete variable or between two discrete variables (Hall, 2000). Linear correlation measures are simple and easy to understand. However, since they are based on the assumption of linear correlation between features, they may not be able to capture nonlinear correlations in nature.

In information theory, *entropy* is a measure of the uncertainty of a random variable. The amount of entropy reduction of variable \(X\) given another variable \(Y\) reflects additional information about \(X\) provided by \(Y\) and is called *information gain* (Quinlan, 1993). Information-theory based correlation measures can capture the different correlation between features in nature. Furthermore, they are applicable to multi-class
problems. However, most entropy-based measures require nominal features. They can be applied to continuous features only if the features have been discretized properly in advance, which may lead to information loss.

4.2.2.1.2 Wrapper Models

None of the evaluation criteria introduced above is dependent on any inductive learning algorithm. Differently, in wrapper models classification accuracy is used as an evaluation criterion. Wrapper models (Kohavi & John, 1997) employ – as a subroutine – a statistical re-sampling technique (such as cross validation) using an inductive learning algorithm to estimate the accuracy of candidate feature subsets. The one with the highest classification accuracy will be identified as the best feature subset.

Since the features selected using the classifier are then used to predict the class labels of unseen instances, accuracy is often very high. However, wrapper models are often criticized for two disadvantages: high computational cost and low generality. First, since the inductive learning algorithm needs to be called as a subroutine repeatedly along the number of iterations, wrappers often suffer from high time complexity. Second, since candidate subsets are assessed based on the error rate of a predetermined classifier, the optimal feature subset is only suitable for this specific classifier. For the same dataset, wrappers may get different optimal feature subsets by applying different classifiers.
4.2.1.2 Generation Procedure

Based on the generation procedure of candidate subsets (i.e., whether features are evaluated individually or collectively), we categorize generation procedures into individual feature ranking (IFR) and feature subset selection (FSS) (Blum & Langley, 1997; Guyon & Elisseeff, 2003).

4.2.1.2.1 Individual Feature Ranking

In individual feature ranking approaches, each feature is measured for its relevance to the class according to a certain criterion. The features are then ranked and the top ones are selected as a good feature subset. Most filters which only aim at removing irrelevant features belong to this category. Individual feature ranking is commonly used for feature selection because of its simplicity, scalability, and good empirical success (Guyon & Elisseeff, 2003). In particular, it is computationally advantageous since it requires only the computation of $N$ scores and sorting the scores, where $N$ is the total number of features.

However, since no correlations among features are exploited, individual feature rankings have several disadvantages. First, the selected features may have high correlation with the class as an individual, but acting together might not give the best classification performance. Second, some features may contain the same correlation information thus introducing redundancy. Third, features that are complementary to each
other in determining the class labels may not be selected if they do not exhibit high individual correlation. Finally, the number of features retained in the feature subset is difficult to determine, often involving human intuition with trial and error.

4.2.1.2.2 Feature Subset Selection

To overcome these shortcomings of individual feature ranking, feature subset selection attempts to find a set of features that serve together to achieve the best classification performance. The advantage of this approach is that it considers not only the relevance of features and the target concept but also the inter-correlation between different features. However, feature subset selection often requires searching a large number of candidate feature subsets. This search process requires great computation expense, especially for data with high dimensionality. Different approaches were proposed to solve this problem: complete search and heuristic search (Dash & Liu, 1997).

- Complete Search

Ideally, feature selection should traverse every single feature subset in the \( N \)-dimensional feature space and try to find the best one among the competing \( 2^N \) candidate subsets according to a certain evaluation function. However, this exhaustive search is known to be \( \text{NP}\)-hard and the search becomes quickly computationally intractable. For a search to be complete does not mean that it must be exhaustive. Different techniques, such as branch & bound (B&B) (Narendra & Fukunaga, 1977) and
best-first search (BFS) (Ginsberg, 1993), were developed to reduce the search without jeopardizing the chances of finding the optimal subset. By applying these search techniques fewer subsets are evaluated, although the order of the search space is $O(2^N)$. Nevertheless, even with these more efficient techniques, complete search is still impractical in cases of high dimensional data.

- Heuristic Search

Other search algorithms generate candidate solutions based on certain heuristics. Heuristic searches can be categorized further as deterministic and non-deterministic.

*Deterministic heuristic search* methods are basically a “hill-climbing” approach which selects or eliminates features in a stepwise manner. At each step of this search process, only remaining features yet to be selected (or rejected) are available for selection (or rejection). One considers local changes to the current state to decide on selecting or removing a specific feature. This search strategy comes in the two forms: *sequential forward selection* (SFS) starts from an empty set and progressively incorporates features; its backward counterpart, *sequential backward selection* (SBS) starts with the full set of features and progressively removes the least promising. These deterministic search methods are computationally advantageous and robust against over-fitting (Guyon & Elisseeff, 2003). However, due to the greedy strategy adopted, they often find local optimal solutions, sometimes called the “nesting effect” (Guyon & Elisseeff, 2003). Even
their extended versions, such as sequential floating forward selection (SFFS) and sequential floating backward selection (SFBS) (Pudil et al., 1994), cannot overcome this drawback completely.

To jump out of the nesting effect without exploring each of the $2^N$ feature subsets, non-deterministic heuristic search methods were introduced to select the optimal or suboptimal feature subsets in a random fashion within a pre-specified number of iterations. A representative algorithm in this category is Las Vegas filter (Liu & Setiono, 1996), which randomly searches the space of subsets using a Las Vegas algorithm that makes probabilistic choices to help guide the search for an optimal set. Simulated annealing (SA) algorithms are based on the analogy between a combinatorial optimization problem and the solid annealing process (Kirkpatrick, Gelatt, & Vecchi, 1983). SA avoids local optima by allowing backtracking according to Metropolis’s criterion based on Boltzman’s probability ($e^{-\Delta E/T}$). A genetic algorithm (GA) is an optimal search technique (Holland, 1975) that behaves like the processes of evolution in nature. GA can find the global (sub)optimal solution in complex multi-dimensional spaces by applying genetic operators such as selection, crossover, and mutation. A tabu search (TS) algorithm is a meta-heuristic method that guides the search for the optimal solution making use of flexible memory which exploits the history of the search (Glover & Laguna, 1999). Many studies have shown that TS can compete and, in many cases,
surpass the best-known techniques such as SA and GA (Glover & Laguna, 1999). These methods are also called \textit{optimal search} because of their ability to find global optimal or suboptimal solutions. In recent years they have been introduced to feature selection and have shown good performances (Siedlecki & Sklansky, 1989; Zhang & Sun, 2002).

4.2.2 Gene Selection for Cancer Diagnosis

Various feature selection approaches have been applied to gene selection for cancer classification.

4.2.2.1 Individual Gene Ranking

Due to its simplicity and scalability, individual gene ranking is the most commonly used in gene selection. A well-known example is the GS method proposed by Golub et al. (1999). The GS method defines a correlation metric to measure the relative class separation produced by the values of a gene. It favors genes that have a large between-class mean value and a small within-class variation. For gene $j$, let $[\mu_{+}(j), \sigma_{+}(j)]$ and $[\mu_{-}(j), \sigma_{-}(j)]$ denote the means and standard deviations of the expression levels of $j$ in the two classes, respectively. A correlation metric $P(j, c)$, so called the signal-to-noise ratio (SNR), is defined as $[\mu_{+}(j) - \mu_{-}(j)] / [\sigma_{+}(j) + \sigma_{-}(j)]$. This metric reflects the difference between the two classes relative to the standard deviation within the classes. Larger values of $|P(j, c)|$ indicate a strong correlation between the gene expression and
the class distinction. Thus, genes are grouped into positive and negative value groups and ranked according to their absolute values. The top $k/2$ genes from the two groups are selected as the informative genes. Similar distance measures such as the Fisher criterion, t-statistic, and median vote relevance (MVR) have also been applied to identification of marker genes (Chow, Moler, & Mian, 2001; Li, Zhang, & Ogihara, 2004; Liu, Li, & Wong, 2002). These measures are often used for binary classification (i.e., distinguishing normal and cancerous tissues). For multi-class cancer classification, metrics such as $F$-statistic and BSS/WSS can be used (Dudoit, Fridlyand, & Speed, 2002). For instance, BSS/WSS is the ratio of a gene’s between-groups to within-groups sum of squares. For a gene $j$, the ratio is

$$\frac{BSS(j)}{WSS(j)} = \frac{\sum_i \sum_{k} I(c_i = k)(\bar{x}_{kj} - \bar{x}_j)^2}{\sum_i \sum_{k} I(c_i = k)(x_{ij} - \bar{x}_{kj})^2}$$

where $\bar{x}_j$ denotes the average value of gene $j$ across all samples, $\bar{x}_{kj}$ denotes the average value of gene $j$ across samples belonging to class $k$, and $I(\cdot)$ denotes the indicator function: 1 if the condition in parentheses is true and 0 otherwise.

4.2.2.2 Gene Subset Selection

Although these IFR methods have been shown to eliminate irrelevant genes effectively, they do not exploit the interaction effects among genes. By contrast, gene subset selection takes into account gene interactions and group performance for cancer
classification. Bø and Jonassen (2002) proposed a gene pair ranking method, which evaluates how well a gene pair can separate two classes. Each gene pair is evaluated by computing the projected coordinates of each experimental sample on the diagonal linear discriminant (DLD) axis. The evaluation score is defined as the two sample $t$-statistic between the two groups of data points. This method can identify cancer-related genes or gene pairs that are not among the top genes when ranked individually. However, this method can capture only pairwise correlations and is limited in binary classification. Ding and Peng employed an approach of minimum redundancy – maximum relevance (MRMR) to find the optimal subset of multiple genes (Ding & Peng, 2003; Peng, Ding, & Long, 2005). Mutual information and F-statistics can be used for discrete and continuous variables, respectively. They used SFS to find the optimal set. This greedy strategy is simple but may result in a local optimal solution.

Unlike filter models, wrapper models use estimated accuracy of a specific classifier to evaluate candidate subsets. Guyon et al. (2002) proposed a support vector machine based recursive feature elimination (RFE) approach to select genes. This approach works as follows: starting with the full gene set, it progressively computes the change in classification error rate for the removal of each gene, then finds and removes the gene with the minimum error rate change. This process tries to retain the gene subset with the highest discriminating power, which may not necessarily be those with highest individual
relevance. Similar approaches can be also found in other studies (Marchevsky, Tsou, & Laird-Offringa, 2004; Model, Adorján, Olek, & Piepenbrock, 2001; Xiong, Fang, & Zhao, 2001; X. Zhang et al., 2006). Some wrappers utilized optimal search instead of greedy search. Li, Weinberg, Darden, and Pedersen (2001) proposed a GA/kNN method to identify genes that can jointly discriminate normal and tumor samples. It ranked genes by their frequency of selection through the iterations of GA and the top ones were selected. However, since it “broke up” subsets, it essentially became an individual ranking approach and unreliable for multi-class classification. Unlike (Li et al., 2001), Ooi and Tan (2003) chose the gene subset with the best fitness among all generations of GA as the optimal subset. This method is shown to achieve high accuracy for multi-class classification. Saeys, Degroeve, Aeyels, de Peer, and Rouze (2003) used an Estimation of Distribution Algorithm (EDA), a general form of GA, to select marker genes and reported good performance. However, optimal search methods such as tabu search have not yet been examined for gene selection from array data.

4.2.3 Research gaps

Table 4.2 summarizes some major studies in gene selection. Many gene selection methods can significantly reduce the number of features and achieve satisfactory accuracy for cancer classification. However, we lack a framework of feature selection for cancer classification. Individual feature ranking have been frequently used but cannot
capture the interactions among genes. Feature subset selection methods based on greedy search methods often can only achieve local optimality. Although some optimal search algorithms such as GA have been applied in gene subset selection, other optimal search methods (e.g., TS) have not been fully investigated for gene selection.

<table>
<thead>
<tr>
<th>Studies</th>
<th>Evaluation Criterion</th>
<th>Generation Procedure</th>
<th>FSS</th>
<th># Classes</th>
<th># Genes</th>
<th># Selected Genes</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Golub et al., 1999)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>6817</td>
<td>50</td>
<td>&gt;90</td>
</tr>
<tr>
<td>(Dudoit et al., 2002)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2~9</td>
<td>6817</td>
<td>30~50</td>
<td>75</td>
</tr>
<tr>
<td>(Guyon et al., 2002)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>2000~7129</td>
<td>8</td>
<td>90</td>
</tr>
<tr>
<td>(Chow et al., 2001)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>7070</td>
<td>50</td>
<td>99</td>
</tr>
<tr>
<td>(Li et al., 2001)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>2000~4026</td>
<td>50~200</td>
<td>97</td>
</tr>
<tr>
<td>(Li et al., 2001)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>81</td>
<td>2~5</td>
<td>90~95</td>
</tr>
<tr>
<td>(Xiong et al., 2001)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2~3</td>
<td>2000~8000</td>
<td>3~4</td>
<td>90</td>
</tr>
<tr>
<td>(Bø &amp; Jonassen, 2002)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>2000~6817</td>
<td>20</td>
<td>80~90</td>
</tr>
<tr>
<td>(Liu et al., 2002)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>12000</td>
<td>20</td>
<td>97</td>
</tr>
<tr>
<td>(Ooi &amp; Tan, 2003)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>9~14</td>
<td>1000</td>
<td>32</td>
<td>82</td>
</tr>
<tr>
<td>(Li et al., 2004)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>3~14</td>
<td>1000</td>
<td>150</td>
<td>93</td>
</tr>
<tr>
<td>(Marchevsky et al. 2004)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2</td>
<td>200</td>
<td>5</td>
<td>80~90</td>
</tr>
<tr>
<td>(Peng et al., 2005)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>2~9</td>
<td>900~9000</td>
<td>≤100</td>
<td>60~90</td>
</tr>
<tr>
<td>(Zhang et al., 2006)</td>
<td>√</td>
<td>Filter, Wrapper</td>
<td>IFR</td>
<td>≥1000</td>
<td>90</td>
<td>85</td>
<td></td>
</tr>
</tbody>
</table>
4.3 Research Questions

In order to address the research gaps above, this study is aimed at investigating feature selection for cancer classification, with emphasis on optimal search-based feature subset selection methods. Specifically, we focus on the following research questions.

Q1. Can feature subset selection methods identify marker genes for cancer classification?

Q2. Will feature subset selection outperform individual feature ranking for cancer classification?

Q3. Will wrappers outperform filters for gene subset selection?

Q4. Which optimal search algorithm will perform better for gene subset selection?

4.4 Optimal Search-based Feature Subset Selection

We develop a framework of optimal search-based feature subset selection for cancer classification. In feature subset selection, for a full set of $N$ features, each feature subset is represented as a string of length $N$ as $[f_1, f_2, \ldots, f_N]$, where each element takes a Boolean value (0 or 1) to indicate whether or not a gene is selected. Specifically, for a feature $i$, $f_i$ equals 1 if it is selected in the subset and equals 0 otherwise.

As shown in Figure 4.1, the overall framework of feature subset selection is composed of two major modules: an optimal search method to generate candidate gene subsets and an evaluation criterion to assess the candidate subsets. The candidate feature
subset with the highest goodness score is regarded as the optimal subset.

Figure 4.1 A Framework of Optimal Search-based Feature Subset Selection

4.4.1 Optimal Search for Feature Subset Selection

As reviewed in Section 4.2, several optimal search methods (e.g., simulated annealing, genetic algorithm, tabu search) can be chosen to generate candidate feature subsets. In this study we choose two optimal search methods, genetic algorithm and tabu search, for their good performance reported in previous studies.
4.4.1.1 Genetic algorithm

Genetic algorithm (GA) is an optimal search method which behaves like evolution processes in nature (Holland, 1975). GA has been used successfully in many applications such as Internet search engines and intelligent information retrieval (Chen, Shankaranarayanan, She, & Iyer, 1998). GA has also been introduced to feature selection (Siedlecki & Sklansky, 1989).

In GA, each solution to a problem is represented in as a chromosome, which in our case is the string representing a gene subset. A pool of strings forms a population. A fitness function is defined to measure the goodness of a solution. A GA searches for the optimal solution by iteratively executing genetic operators to realize evolution. Based on the principle of “survival of the fittest,” strings with higher fitness are more likely to be selected and assigned a number of copies into the mating pool. Next, crossovers randomly choose pairs of strings from the pool with probability $P_c$ and produce two offspring strings by exchanging genetic information between the two parents. Mutations are performed on each string by changing each element at probability $P_m$. Each string in the new population is evaluated based on the fitness function. By repeating this process for a number of generations, the string with the best fitness of all generations is regarded as the optimum. Following is the principal GA for feature subset selection.

$S$ the feature space
120

\( k \)  the current number of iterations

\( x \)  a solution of feature subset

\( x^* \)  the best solution so far

\( f \)  a fitness/objective function

\( f(x) \)  the fitness/objective value of solution \( x \)

\( V_k \)  the \( k \)-th population of solutions

\( P_c \)  the probability of crossover

\( P_m \)  the probability of mutation

1. Generate an initial population \( V_0 \) of feature subset from \( S \) (population size = \text{pop\_size}). Set \( k = 0 \).
2. Evaluate each feature subset in \( V_k \) with respect to the fitness function.
3. Choose a best solution \( x \) in \( V_k \). IF \( f(x) > f(x^*) \) THEN set \( x^* = x \).
4. Based on the fitness value, choose solutions in \( V_k \) to generate a new population \( V_{k+1} \). Set \( k = k+1 \).
5. Apply crossover operators on \( V_k \) with probability \( P_c \).
6. Apply mutation operators on \( V_k \) with probability \( P_m \).
7. IF a stopping condition is met THEN stop ELSE go to Step 2.

Figure 4.2 GA-based Feature Subset Selection

4.4.1.2 Tabu search

GA explores the feature space and generates new solutions by randomly manipulating solutions in the current population. Gene selection is aimed at identifying a small subset of marker genes from a high dimensional feature space. Therefore, for gene
subset selection task, a “smarter” search method is more preferable than the random strategy employed by GA.

Tabu search (TS) algorithm is a meta-heuristic method that guides the search for the optimal solution making use of flexible memory which exploits the search history (Glover & Laguna, 1999). Numerous studies have shown that TS can compete and often surpass the best-known techniques such as GA (Glover & Laguna, 1999). Zhang and Sun used TS for feature selection and showed that the tabu search had a high possibility of obtaining the optimal solution (Zhang & Sun, 2002). However, little study has been conducted to examine its performance on high dimensional data.

Tabu search is based on the assumption that solutions with higher objective value have a higher probability of either leading to a near optimal solution, or leading to a good solution in a fewer number of steps. In each iteration a tabu search moves to the best admissible neighboring solution, either with the greatest improvement or the least deterioration. A tabu list records the reverse of the most recent T moves to avoid cycling. A move in the tabu list is forbidden until it exits the tabu list in a FIFO procedure or it satisfies an aspiration criterion. An aspiration criterion is used to free a tabu move if it is of sufficient quality in terms of objective.

Starting with an initial solution, a TS randomly picks and evaluates a certain number of neighboring solutions, which can be reached by a single move from the current
solution. In particular, for a gene subset, its neighboring solutions are generated by adding or deleting a gene. If the best move is not in the tabu list, or if it is tabu but satisfies the aspiration criterion, then it is picked and made the new solution. The aspiration criterion chosen here is that a move in the tabu list can be taken if it results in a solution of the highest objective value so far. In addition, the tabu list is updated by “remembering” this move and “forgetting” the oldest one if the “memory” is full. If a gene is added (or deleted) at iteration $i$, then deleting (or adding) this gene is incorporated in the tabu list and forbidden in the subsequent $T$ iterations. Not only can a tabu list prevent search from returning to a visited solution, but also help guide the search to achieve the optimal solution more quickly. By repeating this process for a number of iterations, the best solution of all is regarded as the optimum. Following is the principal TS for feature subset selection.

- $S$ the feature space
- $k$ the current number of iterations
- $x$ a solution of feature subset
- $x_k$ the $k$-th solution
- $x^*$ the best solution so far
- $f$ objective function
- $f(x)$ the objective value of solution $x$
\[ N(x_k) \quad \text{all the neighboring solutions of } x_k \]
\[ V(x_k) \quad \text{a random generated subset of } N(x_k) \]
\[ m(x, x') \quad \text{the move from } x \text{ to } x', \text{i.e., adding or deleting a feature} \]
\[ TL \quad \text{a tabu list} \]
\[ T \quad \text{the total length of the tabu list} \]
\[ t \quad \text{the current number of tabu moves in the tabu list} \]

1. Choose an initial feature subset \( x_0 \) in \( S \). Set \( x^* = x_k \), \( k=0 \), and \( t=0 \).
2. Set \( k=k+1 \) and randomly generate a subset \( V(x_k) \) from \( N(x_k) \).
3. Evaluate each feature subset with respect to the objective function \( f \).
4. Choose a best \( x \) in \( V(x_k) \).
5. IF \( m(x_k, x) \) \( TL \) THEN
   IF \( f(x) > f(x^*) \) THEN remove \( m(x_k, x) \) from \( TL \).
   ELSE remove \( x \) from \( V(x_k) \) and go to Step 4.
6. Set \( k=k+1 \) and \( x_k = x \).
7. IF \( t < T \) THEN set \( t=t+1 \) ELSE remove the first item \( m \) from \( TL \).
8. Add \( m(x_k, x_{k-1}) \) to \( TL \).
9. IF \( f(x_k) > f(x^*) \) THEN \( x^* = x_k \).
10. IF a stopping condition is met THEN stop ELSE go to 2.

Figure 4.3 TS-based Feature Subset Selection

4.4.2 Evaluation Criteria for Feature Subset Selection

In order to assess the candidate gene subsets, different evaluation criteria can be used so as to serve the particular decision-making tasks. Because the major objective of gene selection is to improve the accuracy of cancer classification in this study, we mainly focus on evaluation criteria that assess classification performance. Specifically, both filter
and wrapper models are adopted and examined for gene subset selection.

4.4.2.1 Filter: MRMR

A good gene subset contains genes highly relevant with the class, yet uncorrelated with each other. We chose the minimum redundancy-maximum relevance (MRMR) approach because it can remove both irrelevant and redundant genes (Ding & Peng, 2003; Peng, 2005).

For the objective of maximum relevance, an $F$-statistic between a gene and the class label could be adopted as relevance score. The $F$-statistic value of gene $x$ in $K$ classes denoted by $h$ is denoted as $F(x, h)$. Hence, for a feature set $\Theta$, the objective function of maximum relevance can be written as:

$$\max V := \frac{1}{|\Theta|} \sum_{x \in \Theta} F(x, h)$$

where $V$ is the average $F$ value of all the features in $\Theta$ and $|\Theta|$ is the cardinality of the feature subset. Similarly, mutual information can be adopted to measure the relevance between genes and class distinction for discrete variables.

For the other objective of minimum redundancy, the Pearson correlation coefficient between two genes $x$ and $y$, denoted as $r(x, y)$, can be used as the score of redundancy. Regarding both high positive and high negative correlation as redundancy, we take the absolute value of correlation. For a feature subset $\Theta$, the objective of minimum
redundancy can be written as:

$$\min W := \frac{1}{|\Theta|^2} \sum_{x, y \in \Theta} |r(x, y)|$$

where \( W \) is the average correlation coefficient between any two features in \( \Theta \).

These two objectives can be combined in different ways such as difference and quotient. For instance, we could choose a quotient of the two objectives due to its good performances reported in (Ding & Peng, 2003):

$$\max \frac{V}{W} := \frac{1}{1} \sum_{x} \frac{F(x, h)}{\sum_{x, y \in \Theta} |r(x, y)|}.$$ 

4.4.2.2 Wrapper: an SVM classifier

The evaluation criterion in most wrappers is the classification accuracy of a learning algorithm. In this study we chose the support vector machine (SVM) classifier due to its good performance and robustness to high dimensional data (Christianini & Shawe-Taylor, 2000). Initially, SVM is a data-driven method for solving binary classification tasks. Recently it has been modified for multi-class classification problems. A standard SVM separates the two classes with a hyperplane in the feature space such that the distance of either class form the hyperplane, i.e., the margin, is maximal. The prediction of an unseen instance \( z \) is either 1 (a positive instance) or \(-1\) (a negative instance), given by the decision function.
\[ h = f(z) = \text{sgn}(w \cdot z + b). \]

The hyperplane is computed by maximizing a vector of Lagrange multipliers \( \alpha \) in

\[ L(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j h_i h_j K(x_i, x_j) \]

where \( \alpha_1, \alpha_2, \ldots, \alpha_n \geq 0 \), and \( \sum_{i=1}^{n} \alpha_i h_i = 0 \). Function \( K \) is a kernel function and maps the features in the input space into a feature space (possibly of a higher dimension) in which a linear class separation is performed. For a Linear SVM (LSVM), the mapping of \( K \) is a linear mapping: \( K(x_i, x_j) = x_i \ast x_j \).

For each candidate subset, a 10-fold cross validation can be performed to estimate the classification performance of SVM. In particular, all the samples are randomly divided into ten folds. One fold of samples is excluded from the training set, and a classifier is built on the remaining nine folds and used to classify the left-out fold. By repeating this procedure for all ten folds, we can estimate the classification accuracy for the candidate subset.

4.4.4 Four Methods of Gene Subset Selection

By combining the two optimal search algorithms (GA and TS) with the two evaluation criteria (MRMR and SVM), we have four gene subset selection methods: GA/MRMR, TS/MRMR, GA/SVM, and TS/SVM. The former two are filters and the latter two are wrappers. They all consider the group performance of multiple genes and
use optimal search to find the best gene subsets.

4.5 Experimental Study

4.5.1 Dataset Descriptions

This study compared these optimal search-based gene subset selection methods on two datasets of gene arrays (Table 4.3).

Table 4.3 Descriptions of the Three Gene Array Test-beds

<table>
<thead>
<tr>
<th>Type</th>
<th>DNA Methylation</th>
<th>Gene Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Arizona Cancer Center</td>
<td>Alon et al. (1999)</td>
</tr>
<tr>
<td>Test-bed</td>
<td>METH-2</td>
<td>METH-5</td>
</tr>
<tr>
<td></td>
<td>COLON</td>
<td></td>
</tr>
<tr>
<td># Gene</td>
<td>678</td>
<td>678</td>
</tr>
<tr>
<td># Sample</td>
<td>55</td>
<td>43</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Name</th>
<th># S</th>
<th>Name</th>
<th># S</th>
<th>Name</th>
<th># S</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Normal</td>
<td>10</td>
<td>AML</td>
<td>3</td>
<td>Normal</td>
<td>22</td>
</tr>
<tr>
<td>C2</td>
<td>Tumor</td>
<td>45</td>
<td>CMML</td>
<td>10</td>
<td>Tumor</td>
<td>40</td>
</tr>
<tr>
<td>C3</td>
<td></td>
<td></td>
<td>RA</td>
<td>17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td></td>
<td></td>
<td>RAEB</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td></td>
<td></td>
<td>RARS</td>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The first dataset is DNA methylation arrays from the Arizona Cancer Center. It is derived from the epigenomic analysis of bone marrow specimens from healthy donors and individuals with myelodysplastic syndrome (MDS). The MDSs are a heterogeneous and complex group of hematologic disorders and it is estimated that 20% of patients with MDS will evolve to acute myeloid leukemia (AML). Some genetic and epigenetic
aberrations have been identified for MDS. Recent work by Silverman’s group has shown that a DNA methyltransferase inhibitor induces hematologic improvement in 60% of patients and delays conversion to AML, strongly suggesting that aberrant methylation is an important yet reversible pathoepigenetic lesion in MDS thus provides promising therapeutic options. This dataset contained 678 genes and 55 samples. Based on this dataset, we created two test-beds to perform a binary and a multi-class classification, respectively. METH-2 is used to discriminate normal from tumor tissues and METH-5 is used to discriminate five subtypes of tumors.

The second dataset is experimental measurements of gene expression with Affymetrix oligonucleotide arrays of colon cancer tissues (Alon et al., 1999). Colon cancer is the 3rd most common and the 2nd lethal cancer. This dataset contains measurements of 2,000 human genes in 62 tissue samples (40 tumor and 22 normal tissues). The third test-bed (COLON) is used to discriminate normal from tumor tissues.

4.5.2 Evaluation Metrics

To compare different methods, we used the accuracy of an SVM classifier using 10-fold cross validation as the evaluation metric. Cross validation provides a more realistic assessment of classifiers which generalize well to unseen data. A sequential minimal optimization (SMO) method for training a SVM classifier, implemented in WEKA (Garner, 1995), can construct a multi-class classifier and was used in this study.
The formula used to calculate the accuracy is stated below:

\[
\text{Accuracy} = \frac{\text{Number of correctly classified instances}}{\text{Total number of Instances}}
\]

4.5.3 Experimental Results

In experiments we choose $F$-statistic as a baseline individual ranking method. For each test-bed we rank all the genes by their $F$-statistic value and generate gene subsets by picking the top $m$ genes, where $m = 10, 20, \ldots, 100$. The one that achieves the highest accuracy for an SVM classifier is selected as the best subset. For METH-2, the top 20 genes achieved the highest accuracy of 94.364%; for METH-5, the top 40 genes achieved the highest accuracy of 53.333%; and for COLON, the top 70 genes achieved the highest accuracy of 87.581%.

We applied the four methods of optimal search-based gene subset selection on the three test-beds. Then, 10-fold cross validation with an SVM classifier was performed on these gene subsets as well as the full gene set and those obtained from $F$-statistic ranking. For each gene subset, we ran a 10-fold cross validation with SVM classifier 30 times by randomly reconstructing the 10-folds. Figure 4.2 summarizes the classification accuracy and the number of features for each gene subset on the three test-beds.
### METH-2

<table>
<thead>
<tr>
<th>Gene set</th>
<th>#G</th>
<th>Mean</th>
<th>StDev</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full set</td>
<td>678</td>
<td>92.424</td>
<td>0.689</td>
<td>(--*--)</td>
</tr>
<tr>
<td>F-stat</td>
<td>20</td>
<td>94.364</td>
<td>1.809</td>
<td>(--*--)</td>
</tr>
<tr>
<td>GA/MRMR</td>
<td>23</td>
<td>94.424</td>
<td>0.461</td>
<td>(--*--)</td>
</tr>
<tr>
<td>TS/MRMR</td>
<td>6</td>
<td>95.818</td>
<td>1.521</td>
<td>(--*--)</td>
</tr>
<tr>
<td>GA/SVM</td>
<td>47</td>
<td>95.697</td>
<td>1.391</td>
<td>(--*--)</td>
</tr>
<tr>
<td>TS/SVM</td>
<td>20</td>
<td>96.121</td>
<td>0.923</td>
<td>(--*--)</td>
</tr>
</tbody>
</table>

Pooled StDev = 1.229

### METH-5

<table>
<thead>
<tr>
<th>Gene set</th>
<th>#G</th>
<th>Mean</th>
<th>StDev</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full set</td>
<td>678</td>
<td>25.891</td>
<td>3.158</td>
<td>(*)</td>
</tr>
<tr>
<td>F-stat</td>
<td>40</td>
<td>53.333</td>
<td>2.360</td>
<td>(*)</td>
</tr>
<tr>
<td>GA/MRMR</td>
<td>19</td>
<td>46.124</td>
<td>0.882</td>
<td>(*)</td>
</tr>
<tr>
<td>TS/MRMR</td>
<td>9</td>
<td>47.364</td>
<td>1.779</td>
<td>(*)</td>
</tr>
<tr>
<td>GA/SVM</td>
<td>156</td>
<td>54.186</td>
<td>3.621</td>
<td>(*)</td>
</tr>
<tr>
<td>TS/SVM</td>
<td>86</td>
<td>64.729</td>
<td>3.867</td>
<td>(*)</td>
</tr>
</tbody>
</table>

Pooled StDev = 2.815

### COLON

<table>
<thead>
<tr>
<th>Gene set</th>
<th>#G</th>
<th>Mean</th>
<th>StDev</th>
<th>95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full set</td>
<td>2000</td>
<td>83.710</td>
<td>2.002</td>
<td>(--<em>-</em>)</td>
</tr>
<tr>
<td>F-stat</td>
<td>70</td>
<td>87.527</td>
<td>0.941</td>
<td>(--*)</td>
</tr>
<tr>
<td>GA/MRMR</td>
<td>7</td>
<td>86.720</td>
<td>0.694</td>
<td>(--*)</td>
</tr>
<tr>
<td>TS/MRMR</td>
<td>17</td>
<td>87.473</td>
<td>0.917</td>
<td>(--*)</td>
</tr>
<tr>
<td>GA/SVM</td>
<td>245</td>
<td>90.161</td>
<td>2.133</td>
<td>(--*)</td>
</tr>
<tr>
<td>TS/SVM</td>
<td>38</td>
<td>90.430</td>
<td>2.317</td>
<td>(--*)</td>
</tr>
</tbody>
</table>

Pooled StDev = 1.640

**Figure 4.4 Comparing Gene Subsets Obtained by Different Methods.**

- **#G**: number of genes in the gene set;
- **Mean**: mean of classification accuracy;
- **StDev**: standard deviation of classification accuracy.

For the three test-beds, gene subsets obtained by different methods all achieved higher classification accuracy than a full gene set. TS/SVM performed the best (96.121% for METH-2, 64.729% for METH-5, and 90.430% for COLON).

Furthermore, in order to answer research questions Q1~Q4, we conducted pair-wise
$t$-tests to compare different methods (Table 4.4).

### Table 4.4 Pair-wise $t$-tests between Different Methods

<table>
<thead>
<tr>
<th>Comparison</th>
<th>METH-2</th>
<th>METH-5</th>
<th>COLON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene subset vs. full gene sets</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GA/MRMR vs. Full set</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
</tr>
<tr>
<td>TS/MRMR vs. Full set</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
</tr>
<tr>
<td>GA/SVM vs. Full set</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
</tr>
<tr>
<td>TS/SVM vs. Full set</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
</tr>
<tr>
<td>Gene subset selection vs. individual gene ranking</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GA/MRMR vs. F-statistic</td>
<td>0.4299 (&gt;)</td>
<td>0.0000 (&lt;)</td>
<td>0.0002 (&lt;)</td>
</tr>
<tr>
<td>TS/MRMR vs. F-statistic</td>
<td>0.0006 (&gt;)</td>
<td>0.0000 (&lt;)</td>
<td>0.3371 (&lt;)</td>
</tr>
<tr>
<td>GA/SVM vs. F-statistic</td>
<td>0.0011 (&gt;)</td>
<td>0.1425 (&gt;)</td>
<td>0.0000 (&gt;)</td>
</tr>
<tr>
<td>TS/SVM vs. F-statistic</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
</tr>
<tr>
<td>Wrappers vs. filters for gene subset selection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GA/SVM vs. GA/MRMR</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.0000 (&gt;)</td>
</tr>
<tr>
<td>TS/SVM vs. TS/MRMR</td>
<td>0.1778 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.0142 (&gt;)</td>
</tr>
<tr>
<td>Tabu search vs. genetic algorithm for gene subset selection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TS/MRMR vs. GA/MRMR</td>
<td>0.0000 (&gt;)</td>
<td>0.0007 (&gt;)</td>
<td>0.0004 (&gt;)</td>
</tr>
<tr>
<td>TS/SVM vs. GA/SVM</td>
<td>0.0850 (&gt;)</td>
<td>0.0000 (&gt;)</td>
<td>0.3209 (&gt;)</td>
</tr>
</tbody>
</table>

(>): the former outperforms the latter; (<): the latter outperforms the former.

- Gene subsets vs. full set of genes:

  For the three test-beds, gene subsets obtained by GA/MRMR, TS/MRMR, GA/SVM, and TS/SVM all achieved classification accuracy significantly higher than the full gene set ($p = 0.0000$ for all the four methods). These demonstrated the effectiveness of these optimal search-based gene selection methods in identification of marker genes for cancer diagnosis.
• Gene subset selection vs. individual gene ranking:

Compared with the baseline method, $F$-statistic-based individual gene ranking, for METH-2, TS/MRMR, GA/SVM, and TS/SVM identified gene subsets with significantly higher classification accuracy ($p = 0.0006$, $0.0011$, and $0.0000$ for TS/MRMR, GA/SVM, and TS/SVM, respectively) and GA/MRMR also achieved accuracy comparable to the baseline method ($p = 0.4299$). For METH-5, TS/SVM achieved significantly higher accuracy than the baseline method ($p = 0.0000$) and GA/SVM did not outperform the baseline method significantly ($p = 0.1425$). For COLON, GA/SVM and TS/SVM achieved significantly higher accuracy than the baseline method ($p = 0.0000$). These results demonstrated that overall optimal search-based gene subset selection methods tend to outperform individual feature ranking. However, for MRMR, their performance is not as good as individual ranking. $F$-statistic ranking significantly outperformed GA/MRMR and TS/MRMR ($p = 0.0000$) for METH-5 and GA/MRMR for COLON ($p = 0.0002$).

• Wrappers vs. filters:

For all three test-beds, GA/SVM significantly outperformed GA/MRMR ($p = 0.0000$); TS/SVM also achieved better or comparable performance to TS/MRMR ($p = 0.1778$, $0.0000$, and $0.0142$ for METH-2, METH-5, and COLON, respectively). These results are not surprising because wrappers use classification accuracy as the evaluation
criterion whereas filters do not.

- Tabu search vs. genetic algorithm:

  We conducted pair-wise t-tests of TS/MRMR vs. GA/MRMR and TS/SVM vs. GA/SVM. For METH-2, METH-5, and COLON, TS/MRMR significantly outperformed GA/MRMR (p = 0.0000, 0.0007, and 0.0004, respectively). For METH-2 and METH-5, TS/SVM significantly outperformed GA/SVM (p = 0.0850 and 0.0000, respectively). Only for COLON, TS/SVM did not significantly outperform GA/SVM (p = 0.3209). These results showed that tabu search is promising for gene subset selection.

4.4.4 Discussion

Our comparative study demonstrated that the optimal search-based gene subset selection is effective to identify a small subset of marker genes. For example, TS/SVM identified 20 out of 678 genes as marker genes for METH-2, 86 out of 678 for METH-5, and 38 out of 2000 for COLON. These small gene subsets could be used to distinguish tumors with significantly higher accuracy than the full gene set. Furthermore, optimal search-based wrappers often achieved significantly better or comparable performance than individual gene ranking. Gene selection is aimed at identifying the most important genes for cancer diagnosis. In our implementation of genetic algorithm and tabu search, given the same fitness value, gene subsets of smaller size are preferred. Therefore, the result gene subsets tend to be the minimal subset of genes that can achieve the highest
It is interesting that the marker genes identified by optimal search-based selection methods contains several genes that are not among the top genes when ranked individually. Only 2 out of the 20 marker genes identified by TS/SVM for METH-2 are among the top 20 genes ranked by F-statistic; 10 out of 86 genes identified by TS/SVM for METH-5 are among the top 40 genes ranked by F-statistic; and 1 out of 38 genes identified by TS/SVM for COLON are among the top 70 genes ranked by F-statistic. Therefore, taking into account genes’ group performance, optimal search-based gene subset selection can identify marker genes that work collaboratively for cancer distinction. Yet these genes may not be identified by individual ranking.

Statistical comparison in Section IV demonstrated the effectiveness of selected genes for better classification performance. In addition, we had two cancer biologists to evaluate biological relevance of the selected genes for cancer diagnosis. Based on the expert judgment, several cancer-related genes were identified among the gene subset. For instance, in our experiment HOXA1 is identified only by TS/SVM as a marker gene for METH-2. Homeobox genes encode DNA-binding transcriptional regulators that contain a highly conserved motif (the “homeobox”). It has been proposed that deregulation of such genes would participate in human carcinogenesis. In the human genome there are about 200 homeobox containing genes, of which 39 are members of the HOX gene superfamily.
HOXA1 is a member of the A cluster of Hox genes and has been indicated to act as a human mammary epithelial oncogene with aggressive in vivo tumor formation (Zhang et al., 2003). It is worth noting that some recent studies alert to the problem of multiplicity of marker gene subsets (Ein-Dor, Kela, Getz, Givol, & Domany, 2005; Somorjai, Dolenko, & Baumgartner, 2003). Hence, further biological validation is needed to examine the optimality of the selected genes.

Optimal search-based gene subset selection methods also suffer from high dimensionality, which increases the difficulty for GA and TS to find the optimal solution. They also require more computationally expense than the individual ranking because they iteratively evaluate all the candidate gene subsets. The complexity is even higher for wrappers which iteratively call an inductive learning algorithm as a subroutine. Guyon et al. (2002) suggested trading accuracy for speed by initially removing chunks of genes with lower relevance. This process may lose some good genes but can reduce the feature space and make the optimal search easier.

The two optimal search-based MRMR methods do not iteratively train a classifier and therefore have less computational cost. They often identify a smaller gene subset than individual ranking by removing redundant genes. However, these methods often did not achieve higher accuracy than others. It may be questionable to regard highly correlated genes as redundant because they may provide gene interaction information for cancer
diagnosis. In addition, although MRMR assesses the goodness of a gene subset, it essentially combines independent evaluations of individual gene. Therefore, MRMR may not capture strongly interacting genes as gene pair ranking does (Bø & Jonassen, 2002).

The experiments also showed the effectiveness of tabu search for gene selection. TS achieved comparable and often better performance than GA. Due to its use of flexible memory, tabu search is guided by the tabu list which forbids non-promising moves, whereas GA searches in a more random manner. However, since TS only changes one feature a time, it is more time consuming to find the optimal solution.

4.6 Conclusions and Future Directions

In order to identify marker genes from high dimensional gene array data for cancer classification, we developed a framework of optimal search-based feature subset selection. These methods use an optimal search algorithm to generate candidate subsets and evaluate the goodness of each gene group. In this study we used MRMR as the evaluation criterion for a filter and SVM classifier for a wrapper. Genetic algorithm and tabu search were used as the optimal search algorithms. Our comparative study on experimental gene array data demonstrated the effectiveness of optimal search-based gene subset selection. In terms of classification accuracy, optimal search-based wrappers often outperformed the individual ranking. Particularly, tabu search often achieved comparable or higher performance than genetic algorithm. Therefore, tabu search can be
a promising alternative to genetic algorithm for gene selection.

We are in the process of extending our work in the following directions. (1) Optimal search-based feature selection outperforms individual ranking in terms of prediction performance but requires much higher computational expense. We attempt to improve the efficiency of optimal search-based gene subset selection. (2) We will conduct deeper analysis of the biological relevance of the selected genes so as to address the problem of multiplicity of marker gene subsets (Ein-Dor et al., 2005; Somorjai et al., 2003). (3) Furthermore, we will study gene interactions in detail to see whether incorporation of gene interaction information can improve cancer classification.
CHAPTER 5: FEATURE CONSOLIDATION: INTEGRATING GENE FUNCTIONAL RELATIONS FROM HETEROGENEOUS DATA SOURCES

5.1 Introduction

The previous three chapters deal with features extracted from only one single data source. However, in many real-world applications, we need to analyze features from multiple data sources to discover patterns. This chapter focuses on consolidating features from heterogeneous data sources so as to improve the performance of knowledge discovery. Specifically, I develop a framework of integrating gene functional relations extracted from diverse data sources.

Uncovering gene functional relations is one of the major goals of biological studies. These relations can be extracted from a variety of data sources. Many studies have developed data or text mining techniques to exploit special characteristics of different biological data. However, most studies only focus on one single type of data sources (Friedman, Linial, Nachman, & Pe'er, 2000; Jenssen, Laegreid, Komorowski, & Hovig, 2001). Each single data source often can only reveal a certain perspective of the underlying complex biological mechanism. Furthermore, many single source based approaches are criticized for their low reliability as well as limited coverage of genes and relations.

Features from each data source capture a certain type of information and provide a
particular form of evidence about gene relations. Integrating evidence of gene relations from multiple data sources is believed to provide a means to overcome these drawbacks, and thereby benefit studies of genomic functions. By combining multiple forms of evidence, we expect to provide a complete genome-wide functional network and more accurate inferences of unknown gene functions.

With specific interests and experiences in plant science research, we focus on relation extraction and integration for *Arabidopsis thaliana* (or *Arabidopsis*). *Arabidopsis* is one of the model organisms for studying plant genetics and development. The genome of *Arabidopsis*, the first to be sequenced in higher plants, is believed to comprise at least 30,700 genes. Of these genes, the function of approximately one-third (9,194) remain unknown according to the functional Gene Ontology (GO) category listed by the *Arabidopsis* Information Resource (TAIR). Of the remainder, a large proportion lack complete or adequate functional annotation. This study is aimed at constructing a genome-wide functional network of *Arabidopsis* by integrating relations extracted from diverse data sources.

5.2 Literature Review

In previous studies, several approaches have been proposed to integrate gene functional relations extracted from various data sources. We survey the related studies of relation integration from the following three perspectives: data sources, analytical
techniques, and integration methods.

5.2.1 Data Sources

In general, previous studies have involved examination of three types of genomic data. The first type of data source comprises experimental measurements of genes or proteins, such as microarray-based gene expression data (Jansen et al., 2003), yeast two-hybrid experimental data (Uetz et al., 2000), and in vivo pull down experimental data (Gavin et al., 2002). These experimental datasets contain rich biological information but oftentimes also involve noises and errors. The second type of data consists of various genomic and proteomic features, such as gene sequence (Marcotte et al., 1999), gene localization (Yanai & DeLisi, 2002), and metabolic pathway (Marcotte, Pellegrini, Thompson, Yeates, & Eisenberg, 1999). The third type of data is the body of biological knowledge, which contains mostly known and validated gene/protein relations. Some of these known relations are stored in structured format, e.g., Database of Interacting Proteins (DIP) and Munich Information Center for Protein Sequences (MIPS) (Marcotte et al., 1999). There are also other known relations that are represented in unstructured textual format, e.g., Gene Ontology (GO) annotation (Jansen et al., 2003) and biological literature (von Mering et al., 2005). Automatic extraction of relations from text in semi- or non-structural format is a nontrivial task.
5.2.2 Analytical Techniques

These data sources described above provide diverse insights of gene functions. Some functional interactions can be directly inferred from experimental results, such as yeast two-hybrid interactions (Uetz et al., 2000) and *in vivo* pull down experiments (Gavin et al., 2002). For most biological data sources, various analytical techniques must be used to analyze features from the data and extract gene functional relations.

Many studies discover gene relations based on certain criteria of correlation or similarity between genes within a particular data source. For instance, in microarray data comprising simultaneous measurements of thousands of transcripts, the observed correlation of the expression levels of different genes can indicate co-expression and regulatory relations (von Mering et al., 2005). Clustering methods, such as $K$-means, self-organizing maps (SOM), and hierarchical clustering, have been devised to identify the similarity of expression of different genes across multiple samples (Gasch & Eisen, 2002).

Co-occurrence analysis can also be applied to various data sources. These methods assume that the co-occurrence of two items in a certain source indicates their relationships. Phylogenetic profiling (PP) (Pellegrini et al., 1999) and Rosetta Stone (RS or domain fusion) method (Marcotte et al., 1999) are co-occurrence-based methods for gene sequence analysis. Some sources may not directly infer protein interactions *per se,*
but they also contain information associated with the interactions. Marcotte et al. (1999) extracted protein pairs that catalyze sequential reactions in metabolic pathways. Jansen et al. looked at whether two proteins are either both essential or both non-essential for survival (Jansen et al., 2002; Jansen et al., 2003). Co-occurrence analysis has also been applied to extract gene pairs from literature (von Mering et al., 2005).

5.2.3 Integration Methods

The combination of relations from different sources could provide a unified view of gene functional networks with large coverage and high reliability. Previous relation integration methods can be categorized into set-based methods and scoring-based methods.

A set-based method combines different relations using set operations such as intersection or union. Assuming links corroborated by multiple methods are of good quality, intersection can be used to achieve higher reliability (Marcotte et al., 1999). However, intersection often leads to small fractions of relation sets. Alternatively, union can be used to achieve larger coverage of evidence but often lower reliability (Yanai & DeLisi 2002). Since they treat each relation equally, the difference of relations in “quality” is not taken into account in the integration process.

To overcome these pitfalls, more recent studies have used scoring-based methods for relation integration. These methods are often composed of two steps. First, relations
extracted from different resources are scored based on a certain benchmark. Second, based on their quality scores, the relations can be integrated together using different methods. A quality score could be assigned to each relation subset (Jansen, Lan, Qian, & Gerstein, 2002; Troyanskaya, Dolinski, Owen, Altman, & Botstein, 2003; von Mering et al., 2005). For example, the subset profiling method (Jansen et al., 2002) partitions relations from different sources into subsets based on their overlapping patterns and assigns a score to each subset. Subsets are sorted by the score in descending order and added into the network one by one until a good balance of coverage and reliability is achieved. Lee, Date, Adai, and Macotte (2004) assigned a score to each relation and integrated relations using a weighted sum scoring method. Bayesian methods have been shown to be useful in integrating different types of evidence. Naïve Bayes methods assume conditional independence among evidence and are mainly used to integrate relations from unrelated or weakly related data sources (Jansen et al., 2003; Troyanskaya et al., 2003; von Mering et al., 2003; von Mering et al., 2005). Fully connected Bayesian networks (BNs) capture the interdependence and therefore could achieve a more accurate accommodation of correlated evidence (Jansen et al., 2003; Troyanskaya et al., 2003). However, full BNs often require higher computational costs.

5.2.4 Research Gaps

In previous studies, various approaches have been applied to integrate relations
extracted from multiple data sources to improve the reliability and/or the coverage of gene functional networks. However, few studies have conducted a systematic evaluation of the relation integration method and the integrated network. Therefore, designing a framework for extracting and integrating relations from diverse biological sources and evaluating the integrated network remains a challenge. Furthermore, most related studies focused on yeast genes and proteins. For *Arabidopsis*, there has been no study that constructed a genome-wide network by integrating relations from diverse sources. This research gap motivates us to choose and focus on *Arabidopsis* in this study.

5.3 Research Questions

The essence of feature consolidation is to integrate different types of information to improve the performance of knowledge discovery. Specifically, for gene network learning, a successful integration of heterogeneous data sources is expected to bring more insights into biological functions. This study focuses on five main research questions in this study.

Q1. How can gene relations extracted by different techniques from diverse data sources be combined into a genome-wide functional network?

Q2. Can relation integration improve the reliability of a gene functional network?

Q3. How can we evaluate the reliability of the integrated network?

Q4. How does each data source contribute to the integrated gene network?

Q5. How can the integrated network help biological researchers identify gene
5.4 A Framework of Gene Relation Integration

In this study, we propose a framework of integrating gene functional relations (Figure 5.1). The framework contains four major modules: (1) Data sources and benchmarks, (2) Relation extraction, (3) Relation integration, and (4) Evaluation. Gene functional relations are extracted from diverse data sources and integrated into a genome-wide network. Benchmark resources are used to score and validate the gene relations in relation integration and evaluation process.
5.4.1 Data Sources and Benchmarks

Our proposed framework involves three types of data sources: experimental data, biological knowledge, and biological features. In particular, we mainly focus on gene expression data, biological literature, and genome sequence information as the representatives for the three major data sources, respectively.

In our framework, benchmark sources of trusted relations are important to evaluate the relations and validate the integrated network. Benchmarks for this purpose should be as accurate as possible and are usually human-curated. An ideal benchmark set should be independent from the evidence sources, sufficiently large for reliable statistics, and free of systematic biases (Jansen et al., 2003). Some benchmarks are used to score relations supported by different evidence in the relation integration process. Such benchmarks can be called “scoring benchmarks.” The others that are independent of the scoring benchmark are used as a gold standard to validate the correctness of the integrated network. We call these benchmarks as “evaluation benchmarks.”

5.4.2 Relation Extraction

From each data source, e.g., gene expression data ($E$), biological literature ($L$), and genome sequence ($S$), several analytical techniques can be applied to extract functional relations. In Figure 5.1, ($E_1…E_{n_{e}}$), ($L_1…L_{n_{l}}$), and ($S_1…S_{n_{s}}$) denote the relations extracted by a certain technique from the three data sources, where $E_i$, $L_i$ or $S_i$ represents relations
extracted by a particular technique from the corresponding data source; \( n_e, n_l, \) and \( n_s \) denote the number of techniques applied to each data source, respectively. Each technique applied to a particular data source provides certain evidence of functional relations among genes and proteins.

Many analytical techniques can be used for relation extraction. For example, from gene expression data, correlation coefficient analysis (Jansen et al., 2003; Lee et al., 2004; von Mering et al., 2003; von Mering et al., 2005) and Bayesian network learning (Friedman et al., 2000; Huang, Li, Su, Watts, & Chen, 2006) can be used to extract co-expression relationship between genes. Gene relations in the body of biological literature can be extracted by co-occurrence analysis (Jenssen et al., 2001; Lee et al., 2004; Marcotte et al., 1999) or NLP-based parsers (McDonald et al., 2004). For genome sequence information, various sequence analysis methods, such as phylogenetic profiling (PP), Rosetta Stone (RS), gene cluster (GC), and gene neighbor (GN) analysis, are all possible methods for relation extraction (Bowers et al., 2004; Lee et al., 2004).

5.4.3 Relation Integration

Relations extracted by different analytical techniques from diverse data sources are of different quality and reliability. We propose a two-layered Bayesian network for integrating these relations into a genome-wide functional network. We chose a BN approach for its following advantages (Jansen et al., 2003). BNs allow for the combining
of heterogeneous data by converting them into a common probabilistic framework. They weigh evidence according to its reliability. They are readily interpretable as they present conditional probability relationships among different evidence. The structure of our proposed Bayesian network is shown in Figure 5.2.

![Integrated Relations](image)

**Figure 5.2 A Two-layered Bayesian Network for Gene Relation Integration**

In order to integrate relations extracted by different techniques from diverse data sources, we need a unified scoring scheme to measure their quality. In previous literature, a confidence score has been used for this purpose (von Mering et al. 2003; von Mering et al. 2005). We adopt this score as the evaluation criterion for different evidence in this study. Consider evidence $f$ expressed in binary terms (i.e., “present” or “absent”). Each evidence $f$ is evaluated for its ability to reconstruct known gene pathways by comparing its predictions to a common benchmark. A relation that matches a known one in the benchmark is called a true positive. This confidence score, $P(f)$, measures the percentage
of true positives among the total number of relations support by evidence $f$. A greater score indicates a higher degree of reliability of the evidence.

As shown in Figure 5.2, the Bayesian network of relation integration is composed of two layers. At the bottom layer of the Bayesian network are gene functional relations extracted by different analytical techniques from each data source. Since these relations are from the same resource and provide correlated evidence, these multiple forms of evidence from the same data source are combined using fully connected Bayesian networks. Specifically, each relation extracted from a data source $D$ is supported by a particular combination of evidence forms, $f = (f_1, ... , f_{n_d})$, where $n_d$ is the number of techniques used to extract relations from $D$. For each combination of evidence form $f$, we can assign it with a joint confidence score, $P(f) = P(f_1, ... , f_{n_d})$, by benchmarking the supported relations against the common reference set.

Next, the second layer of the Bayesian network combines relations from different data sources into one integrated network. Under the assumption of independence among various data sources, relation integration at this layer is performed in a naïve Bayes fashion. This assumption of independence is valid because relations extracted from the same data source have been joined previously in the first layer. Given $N$ independent forms of evidence $(f_1, ... , f_N)$, the joint confidence score is derived as follows (von Mering et al. 2003; von Mering et al. 2005):
The joint confidence score is often higher than the individual sub-scores, indicating higher reliability of relations supported by multiple forms of evidence.

5.4.4 Evaluation

To examine its validity and to uncover the underlying biological information, we evaluate the integrated network from the following four perspectives.

- **The effect of relation integration**

  We examine the effect of relation integration by comparing the integrated network against relations of each evidence form based on an evaluation benchmark. A successful integration should provide a network of relations with higher reliability than relations supported by individual evidence forms.

- **The reliability of the integrated network**

  Each relation in the integrated network is assigned with a joint confidence score that measures the reliability of this relation. Based on the evaluation benchmark, we evaluate the reliability of the integrated network by analyzing the correlation between the true positive (TP) rate of top relations and the confidence score. A positive correlation between the TP rate and the confidence score can confirm that a higher score indicates higher reliability.

- **Contribution of data sources**
Furthermore, a threshold for reliable relations can be derived by comparing the evaluation benchmark against the scoring benchmark. Relations with a confidence score greater than this threshold are regarded high-quality relations while the others are regarded as noise and removed. We follow the same terminology used in (Lee et al., 2004) and call such a network of reliable relations a “confident network.” We evaluate the confident network by analyzing the distribution of gene functional relations from different data sources. Specifically, we analyze the relations that involve gene of unknown functions. Such relations are interesting to biological researchers because they may infer new functions of genes.

- **Analysis of gene functional clusters**

We analyze the network structure and identify gene clusters by grouping genes according to their connectivity. Based on expert judgments, we can match each gene cluster to a specific biological function. Analysis of these gene functional clusters can further validate the integrated network and bring the researchers more insights into their biological functions.

5.5 Experimental Study

In order to validate our proposed framework for relation integration, we conducted an experimental study on *Arabidopsis* and evaluate the integrated functional network.
5.5.1 Test-bed

Our experimental study involves the three major types of data sources for gene functional relation extraction.

As a senior plant scientist suggested, we chose two high-quality microarray series of Arabidopsis available at http://www.weigelworld.org as the test-bed of gene expression data. These two microarray series include experiments for development (referred as “dev”) and abiotic stress (referred as “abio”). These two datasets both measure the expression levels of 22,810 Arabidopsis genes in 237 and 298 samples, respectively.

The second data source is literature abstracts related to Arabidopsis from PubMed, the online portal of MedLine literature database. We used the MeSH (Medical Subject Headings) terms, “Arabidopsis” and “Arabidopsis Proteins,” to create a sub-collection for Arabidopsis. By April 2005, we collected 10,548 Arabidopsis-related abstracts from PubMed.

For the data source of genome sequence information, we identified and chose the well-known online database named Prolinks (http://dip.doe-mbi.ucla.edu/pronav), constructed by the University of California, Los Angeles (Bowers et al., 2004). This database spans 83 organisms including Arabidopsis thaliana. It combines results of four gene sequence analysis techniques in this database and includes over one million links for Arabidopsis.
5.5.2 Benchmarks

In our framework, benchmarks of known gene functional relations are needed to evaluate the relations and to validate the relation integration approach. Based on the suggestion of domain experts, two reference benchmarks, namely KOG and AraCyc, were selected in this study. The eukaryotic orthologous groups (KOGs) are within the Clusters of Orthologous Groups of proteins (COGs), which define the orthologous proteins among different species with gene functions assigned to 23 broad categories (Tatusov et al., 2003). Of the 25,749 annotated Arabidopsis proteins 53% are claimed to be clustered in KOG. The identification of any pair of proteins as belonging to the same KOG clusters serves as a benchmark for the evaluation of relations. This benchmark has also been used in previous studies such as (Bowers et al., 2004). The second benchmark, AraCyc, is an Arabidopsis pathway database as a part of the BioCyc projects (Mueller, Zhang, & Rhee, 2003). It is generated from genome annotations, and is verified and corrected by human curators. By January 2005, AraCyc covered 186 genetic and metabolic pathways. About 53% of reactions in these pathways have enzymes/genes annotated to them. For AraCyc, each pair of two genes belonging to the same pathway will be regarded as a benchmark relation.

In total, we identified 13,329 genes and 328,493 relations from KOG, 1,141 genes and 25,212 relations from AraCyc. In the relation integration process, we used KOG as
the scoring benchmark to score relations supported by different evidence for its larger coverage of genes and relations. The AraCyc benchmark was then used to evaluate the integrated network.

5.5.3 Selected Techniques for Relation Extraction

In this study, we selected the following techniques to extract gene/protein functional relations from the three data sources.

5.5.3.1 Gene Expression Data

To extract gene relations from gene expression data, we adopted two different analytical techniques, correlation coefficient and the mutual information, both indicating the strength of gene co-expression.

Pearson correlation coefficient (CC) has been commonly used in previous studies for gene co-expression analysis (Lee et al., 2004). A pair of genes with highly correlated expression values is believed to be co-expressed and therefore possibly functionally related. According to this measure, the correlation between a gene $x$ and a gene $y$ can be defined as follows:

$$
r(x, y) = \frac{\sum_k (x_k - \bar{x})(y_k - \bar{y})}{\sqrt{\sum_k (x_k - \bar{x})^2} \sqrt{\sum_k (y_k - \bar{y})^2}},
$$

where $x_k$ and $y_k$ represent the expression level of $x$ and $y$ in the $k$-th sample; $\bar{x}$ and $\bar{y}$
are the means of expression of $x$ and $y$ over all samples, respectively.

In BN learning for microarray data analysis, mutual information (MI) can be used to measure the inter-dependency between genes (Huang et al., 2007). The MI measure between gene $x$ and gene $y$ is defined as follows:

$$I(x, y) = \sum_{i,j} P(X_i, Y_j) \log \frac{P(X_i, Y_j)}{P(X_i)P(Y_j)}$$

where $X_i$ is the $i^{th}$ expression level of gene $x$, $Y_j$ is the $j^{th}$ expression level of gene $y$; $P(X_i)$ is the probability that the expression level of $x$ equals $X_i$; $P(Y_j)$ is the probability that the expression level of $y$ equals $Y_j$, and $P(X_i, Y_j)$ is the probability that the expression level of $x$ equals $X_i$ and the expression level of $y$ equals $Y_j$.

5.5.3.2 Biological Literature

In this study we used two well-known text mining techniques to extract gene relations from the body of biological literature. Co-occurrence analysis is commonly applied in mining gene relations from literature (Jenssen et al., 2001). It infers the existence of a certain functional relation between a pair of genes when they both appear within the same abstract. We extracted co-occurrence relations by searching genes that are cited in the same literature abstracts. In this study we only focus on binary co-occurrence relations from literature.

Previous studies show that fine-grained text analysis based on linguistic and
semantic sentence parsing produces more precise gene relations compared with raw co-occurrence analysis. In this study we adopted the Arizona Relation Parser (McDonald, Chen, Su, & Byron, 2004), a text mining tool developed by the University of Arizona, to fully parse meaningful gene relations from textual sentences in literature abstracts. This relation parser was reported to outperform other parsers in its use of a broad coverage syntax-semantic hybrid grammar (McDonald et al., 2004). In order to reduce the ambiguities for biological entities and functional processes in the extracted relations, we created an aggregate lexicon specifically for *Arabidopsis* and used an aggregation module, called BioAggregate Tagger, to aggregate relations and to capture important contextual information (Marshall, Su, McDonald, Eggers, & Chen, 2006).

5.5.3.3 Genome Sequence

In this study we included *Arabidopsis* protein functional relations extracted by four standard sequence analysis techniques (i.e., PP, RS, GN, and GC) available in the Prolinks database (Bowers et al., 2004). The PP method constructs profiles by analyzing the distribution of protein sequences across known genomes. Proteins with similar phylogenetic profiles are likely to participate in the same pathway. The RS method compares genomic sequence information across organisms, by searching for examples of protein coding sequences that represent the fusion of coding sequences that are separated in the genomes of other organisms. The fusion protein is termed the Rosetta Stone protein,
in that it allows us to infer the functional linkage between two proteins. The GN method identifies protein pairs encoded in close proximity across multiple genomes. The GC or operon method identifies closely spaced genes, and assigns a probability of observing a particular gap distance (or smaller), as judged by the collective set of inter-gene distances.

5.5.4 Relation Extraction and Integration

Genes are denoted by different symbols or identifiers in different data sources. Relation integration requires that all genes are denoted according to a common naming scheme. We mapped genes from different resources to their Entrez Gene identifiers. Relations involving genes that could not be mapped were eliminated. The numbers of extracted relations from different data sources are summarized in Table 5.1.

<table>
<thead>
<tr>
<th>Data Resources</th>
<th>Analytical Techniques</th>
<th># of relations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene expression</td>
<td>Correlation (CC)</td>
<td>250,600,078</td>
</tr>
<tr>
<td></td>
<td>Mutual information (MI)</td>
<td>250,600,078</td>
</tr>
<tr>
<td>Biological literature</td>
<td>Co-occurrence (CO)</td>
<td>1,291</td>
</tr>
<tr>
<td></td>
<td>Relation parsing (RP)</td>
<td>222</td>
</tr>
<tr>
<td>Genome sequence</td>
<td>Phylogenetic profiling (PP)</td>
<td>132,637</td>
</tr>
<tr>
<td></td>
<td>Rosetta Stone (RS)</td>
<td>989,795</td>
</tr>
<tr>
<td></td>
<td>Gene neighbor (GN)</td>
<td>18,823</td>
</tr>
<tr>
<td></td>
<td>Gene cluster (GC)</td>
<td>11,586</td>
</tr>
</tbody>
</table>
Based on our two-layered BN approach, relations supported by different evidence are scored by benchmarking against KOG and combined into an integrated network. The integrated network contains 25,660 genes and 250,762,268 relations.

5.5.5 Evaluation

Based on the proposed framework, we evaluated the integrated gene functional network.

5.5.5.1 Evaluation Metric

Because KOG was used as the scoring benchmark already, we used the other benchmark, AraCyc, to evaluate the integrated network. All the relations in the integrated network are sorted by the joint confidence score in a descending order. We use TP rate of top relations as the evaluation metric. This metric has also been used in previous studies (Bowers et al., 2004; Lee et al., 2004).

5.5.5.2 The Effect of Relation Integration

In order to demonstrate the power of relation integration, we compared the relations in the integrated network with those supported by evidence from each individual resource based on the AraCyc benchmark (Figure 5.3). CC, MI, CO, RP, PP, RS, GN, and GC represent different analytical techniques. For CC and MI, dev/abio corresponds to the dataset of development/abiotic stress.
Figure 5.3 Integrated Network and Individual Evidence from Different Data Sources

Relations extracted from gene expression are associated with a correlation or a mutual information value, corresponding to a different confidence score. We evaluated these relations sorted by the confidence score and drew a curve for each form of evidence from gene expression data. In contrast, relations extracted from literature and genome sequences are binary relations in this study. Each type of these relations is assigned with a single confidence score. Therefore, each of these evidence forms is shown as a dot in Figure 5.3. In this study the TP rates for these evidence forms, i.e., CO, RP, PP, RS, GN, and GC, are 0.0380, 0.0252, 0.0062, 0.0020, 0.0366, and 0.0044, respectively. We found
that among all different evidence, co-occurrence relations from literature had the highest TP rate, while relations extracted by MI analysis from the abiotic stress dataset had the lowest TP rate of all.

In general, the curve of integrated relations (JOINT) is above most of the other curves or dots, except for relations extracted by PP and GN from gene sequence. It demonstrates that the BN-based approach achieved higher TP rate by combining evidence from multiple resources. Relations supported by diverse forms of evidence are more likely to be correct. This highlights the merit of feature consolidation: weak evidence from multiple resources can be combined to provide strong evidence for a relation.

5.5.5.3 Reliability of the Integrated Network

In the integrated network, relations with a higher joint score are believed to be more reliable. Given different thresholds of joint confidence score, denoted by $P_{\text{cut}}$, we evaluated the TP rate of the relations above $P_{\text{cut}}$ based on the AraCyc benchmark. Figure 5.4 shows the TP rate of top relations in the network against $P_{\text{cut}}$. In general, the TP rate of top relations increases as $P_{\text{cut}}$ increases. Regression analysis on the TP rate against $P_{\text{cut}}$ shows that R-square between TP rate and $P_{\text{cut}}$ is 0.78 and the regression coefficient of the independent variable $P_{\text{cut}}$ is 0.28 (p < 0.0001). This result demonstrated a significant positive correlation between the TP rate of relations and the confidence score. Gene relations with a higher joint confidence score are more likely to be correct.
5.5.5.4 Contribution of Data Sources

In the integrated network of over 250 million relations, many of the relations represented noise. We used the AraCyc as a gold standard to eliminate low quality relations and thereby to identify a network with reliable relations. Among the 25,212 relations from AraCyc, 2,853 relations can be matched with those from KOG, i.e., the probability of matching is about equal to 0.1. Thus, we used the top relations with a joint confidence score greater than 0.1 to form the “confident network”. In this network, the number of genes was reduced to 11,082 and the number of relations was reduced to 1,007,883.

The confident network is believed to provide strong evidence about relations between various genes and proteins. We analyzed distribution of relations from different data sources to evaluate the contribution of each data source. The confident network
includes many genes of unknown function. Reliable relations involving such genes are interesting to biological researchers in raising new hypotheses about gene biological functions. Among the 11,082 genes in this network, there are 207 genes lacking an associated GO term and 3,027 genes associated with the GO term “biological process unknown,” which we collectively regard as genes of unknown function (GUF). Table 5.2 summarizes the distribution of gene relations from different data sources (E: gene expression data; L: biological literature; S: genome sequence) in the confident network. “√” and “–” indicate whether relations are from the source or not, respectively.

Table 5.2 Distribution of Gene Relations from Different Data Sources

<table>
<thead>
<tr>
<th>Data sources</th>
<th># of relations</th>
<th># of relations with GUF</th>
<th>% of relations with GUF</th>
</tr>
</thead>
<tbody>
<tr>
<td>E L S</td>
<td># of relations</td>
<td># of relations with GUF</td>
<td>% of relations with GUF</td>
</tr>
<tr>
<td>√ – –</td>
<td>50</td>
<td>30</td>
<td>60.00%</td>
</tr>
<tr>
<td>– √ –</td>
<td>87</td>
<td>65</td>
<td>74.71%</td>
</tr>
<tr>
<td>– – √</td>
<td>156,052</td>
<td>78,572</td>
<td>50.35%</td>
</tr>
<tr>
<td>√ √ –</td>
<td>840</td>
<td>292</td>
<td>34.76%</td>
</tr>
<tr>
<td>√ – √</td>
<td>850,689</td>
<td>266,618</td>
<td>31.34%</td>
</tr>
<tr>
<td>– √ √</td>
<td>14</td>
<td>2</td>
<td>14.29%</td>
</tr>
<tr>
<td>√ √ √</td>
<td>151</td>
<td>14</td>
<td>9.27%</td>
</tr>
<tr>
<td>Total</td>
<td>1,007,883</td>
<td>345,593</td>
<td>34.25%</td>
</tr>
</tbody>
</table>

Among the 1,007,883 relations in the confident network, most were extracted from genome sequence information and gene expression data. In addition, only 1,092 relations (87 + 840 + 14 + 151) were supported by evidence from biological literature. Among all
the relations, there are 345,593 relations (34.25%) involving GUF. Again, genome sequence and gene expression data were the major sources of these relations. In general, relations supported by fewer forms of evidence tend to involve genes of unknown functions, and vice versa. Particularly, relations only from literature had the highest percentage of relations involving GUF (74.71%), and relations from all the three data sources had the lowest percentage of relations with GUF (9.27%).

5.5.5.5 Analysis of Gene Functional Clusters

Relation integration provided a genome-wide network for *Arabidopsis*. In order to gain more insight about the network, we analyzed the network structure based on a stepwise creation process. Specifically, in descending order of the joint confidence score, we added one link at a time to construct the network. Thus, the sub-networks at each step are only composed of the functional relations with the highest joint score, i.e., with the strongest evidence to support. At the early stages, links with higher scores were mainly disconnected from each other because of the large coverage of the network. As more links were added, genes were gradually grouped into clusters, which may infer the existence of gene functional groups. After including the top 3,000 relations, three major clusters of gene functional relations emerged. Cluster A contains 492 nodes and 937 links, cluster B contains 221 nodes and 668 links, and cluster C contains 122 nodes and 145 links. We reviewed these three clusters and found a relatively clear distinction of function among
them. Particularly, genes in cluster A are mainly involved in regulation of transcription, genes in cluster B in protein phosphorylation/kinase activity, and those in cluster C in electron transport.

Analyzing the subnet could provide more insight into the gene functions. For example, in a subnet centered on \textit{PHYB} (At2g18790), we found that genes were connected by linkages supported by different evidence. Many genes in this subnet are transcription factors, kinases, or photoreceptors, which are typical players in signal transduction pathways. It is also observed that genes involved in the same pathway tend to form sub-clusters. For instance, GA1 (At4g02780), GA3 (At5g25900), and GA4 (At1g15550) are enzymes involved in gibberellic acid biosynthesis and signaling (Cowling et al., 1998; Helliwell, et al., 1999; Sun & Kamiya, 1994); FT (At1g65480), LFY (At5g61850), and CO (CONSTANS, At5g15840) are transcription factors that are responsible for flowering control or flower development (Ayre & Turgeon, 2004; Huang Böhlenius, Eriksson, Parcy, & Nilsson, 2005; Moon, Lee, Kim, & Lee, 2005). These results provide some evidence for the potential value of the proposed framework to integrate gene relations from multiple data sources.

5.6 Conclusions and Future Directions

In this study, we developed a framework for relation integration that combines evidence from different sources to construct a genome-wide functional network. We
employed different analytical techniques to extract gene relations from three data sources: gene expression data, biological literature, and genome sequence information, and we constructed a gene functional network of *Arabidopsis thaliana*. Evaluation on the integrated network confirmed the validity and potential value of the proposed framework for relation integration. To the best of our knowledge, this is the first study to construct such a genome-wide network of *Arabidopsis* by integrating relations from diverse data sources.

In the future, we plan to study prediction of new gene/protein relations based on topological analysis of the integrated network. The predicted relations can be evaluated and integrated into the network based on the Bayesian network. Furthermore, we plan to examine our proposed framework on a larger test-bed of related plant genes as well as on other biomedical domains such as cancers.
CHAPTER 6: FEATURE CONSOLIDATION: KERNEL-BASED LEARNING FOR BIOMEDICAL RELATION EXTRACTION

6.1 Introduction

 Previous chapters focused on feature-based methods that require explicitly extracting features from various data sources. However, in some applications data are represented in complex structure so that features cannot be easily extracted to capture the key information. Moreover, features defined on different data representations can be heterogeneous and difficult to consolidate. This chapter focuses on an alternative to feature methods, namely kernel-based learning, for knowledge discovery. In particular, I develop a framework of kernel-based learning methods for relation extraction from text.

 Information extraction is an important task in natural language processing (NLP). It is aimed at scanning text for information of interest, including entities and relations among them. Information extraction has many practical applications, primarily to economic, military, and scientific domains. So far, reliable extraction of relations between named entities is still a difficult and unsolved NLP problem. Moreover, the emergence of new application domains often brings new challenges to relation extraction.

 The recent advent of new experimental and analytical technologies in biomedicine has led to the rapid growth in biomedical research development and discovery. PubMed, the online portal of the U.S National Library of Medicine (NLM), is a valuable source of
biomedical research findings, including over 16 million articles from Medline and other life science journals. The size of PubMed abstracts grows at an average rate of 1,760 per day in 2005. Such a huge literature body makes manual inspection of biomedical findings very difficult and time-consuming. Automatically extracting biomedical information has been the subject of significant research efforts (Shatkay & Feldman 2003). Named entity taggers enable the efficient identification of biomedical entities such as genes and proteins (Bunescu, Ge et al., 2005; Hirschman, Yeh, Blaschke, & Valencia, 2005; Settles, 2005). Furthermore, a more challenging task is to identify inter-entity relations of these biomedical entities.

Substantial studies have focused on extracting biomedical relations such as identifying gene-disease relations, protein interactions, or subcellular localizations. However, most biomedical relation extractors still require manual development of lexicons and parsing rules based on domain knowledge. Recently, statistical learning methods have been introduced to information extraction and shown promising performance for general corpora. Due to the unique patterns of biomedical relations, techniques designed for extracting relations from general text may not be suitable for biomedical domain. This study is aimed at designing and examining kernel-based learning methods to extract biomedical relations from literature text.

The remainder of this paper is organized as follows. Section 6.2 reviews existing
relation extraction approaches. Section 6.3 raises our research questions. In Section 6.4 we develop a framework of kernel-based learning for biomedical relation extraction and introduce different kernels. Section 6.5 presents our experiment on a biomedical corpus. We conclude the study in Section 6.6 by summarizing key insights and future directions.

6.2 Literature Review

We categorize biomedical relation extraction approaches into three types: co-occurrence analysis, rule-based approaches, and statistical learning. Under each type, methods vary in how they utilize the lexical, syntactic, and semantic information in texts.

6.2.1 Co-occurrence Analysis

Co-occurrence analysis identifies relations between biomedical entities based on their probabilities of occurrence in articles (Stapley & Benoit, 2000; Jenssen et al., 2001). These approaches are based on the assumption that if two entities are co-mentioned in the same article there is an underlying biological relationship. In most cases, only lexical information (i.e., words) is needed for co-occurrence analysis. Due to their simplicity and flexibility, these approaches have been widely used for relation extraction and can achieve high recall. However, since it can capture little syntactic or semantic information, co-occurrence analysis cannot distinguish relation types or directionality and often achieves low precision.
6.2.2 Rule-based Approaches

Researchers have manually developed rules based on syntactic or semantic information to parse relations from text. Syntactic information such as part-of-speech (POS) and syntax structures can be represented via the data structures such as parse trees. Syntax parsing approaches extensively utilize syntactic information and reply on syntactic rules for relation extraction (Thomas, Milward, Ouzounis, Pulman, & Carroll, 2000; Park, Kim, & Kim, 2001; Yakushiji, Tateisi, Miyao, & Tsujii, 2001; Leroy, Chen, & Martinez, 2003). These approaches are general and flexible in terms of the applicable domain. However, existing syntax parsers are rarely able to cover all the variety of syntactic patterns for relations. Some syntax parsers with large coverage may over-generate irrelevant parses and lead to incorrect relations. Therefore, parsers that primarily rely on syntactic rules generally achieve poor precision.

Another type of rule-based approaches relies more on semantic information in sentences (Rindflesch, Tanabe, Weinstein, & Hunter, 2000; Friedman, Kra, Krauthammer, & Rzhetsky, 2001; Pustejovsky, Castaño, Zhang, Kotecki, & Cochran 2002). Semantic information often consists of certain slots of domain-specific trigger words (e.g., “interact with,” “inhibit,” or “bind to”) and is often represented as a template. A pair of biomedical entities whose contextual information satisfies a certain pre-defined semantic template is identified as a relation. Many semantic parsers have reported higher precision than syntax
parsers. However, semantic parsers are also subject to poor coverage of templates. Moreover, semantic templates are largely based on domain-specific lexicons and therefore have lower portability across domains.

To address the limitations of these two approaches, hybrid parsers have been developed to take advantage of both syntactic and semantic information. In most hybrid approaches (Gaizauskas, Demetriou, Artymiuk, & Willett, 2003; Novichkova, Egorov, & Daraselia, 2003), syntactic analysis takes place first to create possible parses from the original sentence. Next, they eliminate incorrect parses and identify domain words (such as genes) based on semantic information. Unfortunately, semantic analysis after syntactic processing still cannot effectively improve the poor coverage of syntax grammars. McDonald et al. (2004) combined access to syntactic and semantic information via a single grammar and reported higher precision and recall. Such hybrid parsers maintain both the flexibility of syntax parsing and high precision of semantic analysis. However, these rule-based approaches require manual encoding of syntactic and semantic rules, which is very labor intensive and time-consuming.

6.2.3 Statistical Learning

Unlike rule-based approaches, statistical learning requires little or no manual development of rules or templates. Instead, patterns are automatically learned from a corpus of documents in which human experts have tagged the desired relations. Thus, a
model consisting of these patterns can be used to extract relations from new documents. Although the annotation of relations in the corpus still requires manual inspection, the whole framework is highly portable. Statistical learning can be categorized into feature-based methods and kernel-based methods.

For feature-based methods, each data instance is represented as a feature vector \( X = \{x_1, x_2, \ldots, x_n\} \) in an \( n \)-dimensional space. Features are defined and selected to capture the data characteristics. For instance, “bag-of-words” methods represent a piece of text as a vector where each element indicates the occurrence of a specific word (Donaldson et al., 2003; Mitsumori, Murata, Fukuda, Doi, & Doi, 2006). Rosario and Hearst (2005) compared generative graphical and discriminative models for relation extraction using both word and role features. Benescu et al. (2005) developed three generalization methods that utilize word or POS sequences as features and induce rules of protein relations. These methods require features be explicitly defined and enumerated for a vector representation. Unfortunately, Natural Language Processing (NLP) tasks often involve large amounts of words, which can lead to high dimensionality but sparse feature vectors. If a sentence is represented as complex structures such as a parse tree, features cannot be easily defined to capture the structural information. Moreover, features defined on heterogeneous data representations (e.g., “bag-of-words” and parse trees) capture different information but may be incompatible to each other.
Kernel-based methods are an effective alternative to explicit feature extraction (Cristianini & Shawe-Taylor, 2000). They retain the original representation of objects and use the object only via computing a kernel function between a pair of objects. Formally, a kernel function is a mapping $K: X \times X \rightarrow [0, \infty)$ from input space $X$ to a similarity score $K(x,y) = \phi(x) \cdot \phi(y) = \sum_i \phi_i(x)\phi_i(y)$, where $\phi_i(x)$ is a function that maps $X$ to a higher dimensional space with no need to know its explicit representation. A kernel function is required to be symmetric and positive-semidefinite. Such a kernel function makes it possible to compute the similarity between objects without enumerating all the features.

Given a kernel matrix of pair-wise similarity values, a kernel machine, such as a support vector machine (SVM) (Cristianini & Shawe-Taylor, 2000), can train a model for future prediction. Kernel-based methods have been frequently used in the machine learning areas, such as pattern recognition (Chen, Yuen, Huang, & Dai, 2005; Zhao, Yuen, & Kwok, 2006), data mining (Zhou & Wang, 2005), text mining (Sun, Lim, Ng, & Srivastava, 2004), and Web mining (Yu, Han, & Chang, 2004). The performances of kernel methods are mainly determined by the selected kernel functions.

In NLP, various kernels have been applied to information extraction. For simple data representations (e.g., “bag-of-word”) in which features can be easily extracted, some basic kernel functions such as linear kernel, polynomial kernel, and Gaussian kernel are often used. For data in structured representation, convolution kernels are frequently used
Convolution kernels are a family of kernel functions, including string kernels (Lodhi, Saunders, Shawe-Taylor, Cristianini, & Watkins, 2002), tree kernels (Zelenko, Aone, & Richardella, 2003), and so on. They define the similarity between objects as the convolution of “sub-kernels,” i.e., the kernels for the decomposition of the objects. String kernels capture the sequence patterns in data instances. Lodhi et al. (2002) defined string kernels on letter or word sequence in sentences for text classification. Tree kernels capture the structure of syntactic parse tree and have been applied in relation extraction (Zelenko et al., 2003). Some recent studies have revised these tree kernels by incorporating richer semantic information (Culotta & Sorensen, 2004; Bunescu & Mooney, 2005). In most tree kernels, each relation instance is represented by the minimum subtree that covers the entity pair, which can often capture the major contextual information but may lose some useful information in the rest of the tree.

Another advantage of kernel methods is that they transform different data representations into kernel matrices of the same format, which enables the integration of heterogeneous information (Cristianini & Shawe-Taylor, 2000; Joachims et al., 2001; Lanckriet, de Bie, Cristianini, Jordan, & Noble, 2004). Studies have shown that composite kernels can reduce kernel sparsity and improve learning performance for NLP (Culotta & Sorensen 2004; Zhao & Grishman, 2005).
6.2.4 Summary and Research Gaps

Among various biomedical relation extractors, co-occurrence analysis utilizes little contextual information, whereas rule-based approaches based on syntactic and/or semantic information have shown good performance but require significant manual efforts. Statistical learning can automatically learn relation patterns from annotated corpora. In particular, kernel-based learning methods have shown promises in identifying various social relations such as action-role, part-of, or locational relations between named entities such as people, organizations, and locations from newspaper articles (Zelenko et al., 2003; Bunescu & Mooney, 2005). A major challenge in biomedical relation extraction is that current POS taggers and parsers that were usually trained on general text do not perform well on the biomedical literature (Bunescu et al. 2006). Moreover, kernel functions designed for general relation extraction may not be suitable for biomedical domain. To the best of my knowledge, there have been few studies that systematically evaluated kernel methods for biomedical relation extraction. Therefore, designing a framework of kernel-based methods for biomedical relation extraction is a necessity and a challenge.

6.3 Research Questions

To address these issues, we examine the performances of kernel-based methods for biomedical relation extraction. In this study we only deal with intra-sentence relations.
Specifically, given a sentence \( s \) consisting of entity \( \{e_1, \ldots, e_n\} \), we aim to identify and classify all binary relations \( r_{ij} = \langle e_i, e_j \rangle \). Our study focuses on four research questions.

Q1. How can we use kernel-based learning to extract biomedical relations from literature text?

Q2. Which data representation of instances is better for kernel-based relation extraction?

Q3. How can we modify kernels to improve the performance of relation extraction?

Q4. Can the composition of kernels improve the performance of relation extraction?

6.4 Kernel-based Learning for Biomedical Relation Extraction

We develop a framework of kernel-based learning for biomedical relation extraction as shown in Figure 6.1. This framework can be decomposed into four major modules: *entity recognition, relation annotation, kernel construction, and learning & evaluation.*
6.4.1 Entity Recognition

A prerequisite step for relation extraction is the identification of named entities from text. This entity recognition process can be performed by experts based on their domain knowledge. This manual approach can assure high correctness but is time-consuming for large corpora. Alternatively, biomedical entity taggers (Bunescu et al., 2005; Hirschman et al., 2005; Settles, 2005) can be employed to automate the entity recognition process.

6.4.2 Relation Annotation

Statistical learning requires a training corpus of annotated relations from which
patterns of true relations can be learned. To create such a training dataset, domain experts need to read the literature text in the corpora and manually annotate meaningful relations. For relation classification tasks, the type of biomedical relations, such as activation and inhibition, needs to be labeled as well. Some databases that summarize documented biomedical relations have been used as a proxy for training data (Rosario & Hearst 2005).

6.4.3 Kernel Construction

Statistical learning requires both positive and negative examples. Each entity pair that appears in the same sentence is regarded as a potential relation or a relation instance. Each instance can be represented by a “bag-of-words,” a word sequence, or a parse tree. Figure 6.2 shows an example of these three representations for the sentence: “Mutant p53 genes increase IL-6 expression,” in which “p53” and “IL-6” are identified as two biomedical entities (genes). For each representation that captures different contextual information, we can design a kernel function \( K(x, y) \) to compute the similarity between relation instances.
6.4.3.1 Word Kernel

The simplest and most used representation of relation instances is a “bag-of-words,” $W = \{w_1, \ldots, w_N\}$. Specifically, each relation instance is represented as a vector where each element indicates the occurrence of a particular word in the sentence. Here, $\phi_i(x) = 1$ if word $i$ occurs in sentence $x$. Without listing all the possible words, a word kernel (or a “bag-of-words” kernel) $K_W(x, y)$ takes the inner production of $\phi(x)$ and $\phi(y)$ and returns the number of words in common between two instances. The word kernel is simple and efficient. However, neither the word sequence nor the sentence structure is captured by word kernels.
6.4.3.2 Sequence Kernel

By considering the sequential order of words in a sentence, we can represent a relation instance as a sequence of words, \( S = [w_1 \ldots w_N] \). For each relation instance, all words except for the two named entities are listed according to their sequence in a sentence. Such a sequence can be further decomposed into subsequences of \( n \)-grams. The string kernel proposed by Lodhi et al. (2002) was first used for text classification by analyzing the subsequences of letters or words. Recently, this later has been extended and applied to relation extraction from text (Bunescu et al. 2006).

A subsequence is a finite sequence of words. For sequence \( s \) and \( t \), we denote by \( |s| \) the length of the sequence \( s = s_1 \ldots s_{|s|} \), and by \( st \) the sequence obtained by concatenating the sequence of \( s \) and \( t \). The sequence \( s[i:j] \) is the subsequence \( s_i \ldots s_j \) of \( s \). We say that \( u \) is a subsequence of \( s \), if there exist indices \( i = (i_1, \ldots, i_{|u|}) \), with \( 1 \leq i_1 < \ldots < i_{|u|} \leq |s| \), such that \( u_j = s_{i_j} \), for \( j = 1, \ldots, |u| \), or \( u = s[i] \) for short. The length \( l(i) \) of the subsequence in \( s \) is \( i_{|u|} - i_1 + 1 \).

The feature mapping \( \phi \) for a sequence \( s \) is given by defining the \( u \) coordinate \( \phi_u(s) \) for each subsequence \( u \). We define:

\[
\phi_u(s) = \sum_{i \in s[i]} \lambda^{l(i)}
\]

where \( 0 < \lambda \leq 1 \). These features measure the number of occurrences of subsequences in the \( s \) weighting them according to their lengths. \( \lambda \) is the decay factor to penalize subsequences with more interior gaps and therefore longer length.
Hence, the inner product of the feature vectors for two sequences \( s \) and \( t \) give a sum over all common subsequences weighted according to their frequency of occurrence and length.

\[
K_n(s, t) = \sum_{u \in \Sigma^n} \phi_u(s) \cdot \phi_u(t) = \sum_{u \in \Sigma^n} \sum_{|i|=a} \sum_{|j|=b} \lambda^{(i) + (j)}
\]

The sequence kernel \( K_S \) accumulates \( K_n \) of different sequence length to get the overall similarity between two sequences:

\[
K_S(s, t) = \sum_n K_n(s, t)
\]

6.4.3.3 Tree Kernels

Given a sentence, a syntax parser can create a parse tree that shows the syntactic structure (Lease & Charniak, 2005). In most studies, a relation instance is often represented as a minimum subtree that contains the two entities (Zelenko et al., 2003). A tree (or a subtree) \( T \) is represented as \( \{p, [T.c]\} \), where \( p \) is the \( T \)'s root node with a set of attributes \( V = \{v_1, v_2, \ldots\} \) and \([T.c]\) denotes \( p \)'s children (nodes or subtrees). The node attributes often consist of word, POS, and entity type (e.g., gene, protein, disease, or function). Some recent studies have incorporated attributes such as chunk-tag and Wordnet hypernyms to capture more semantic information (Culotta & Sorensen, 2004; Harabagiu, Bejan, & Morarescu, 2005). Given such a more structured representation, Zelenko et al. (2003) proposed tree kernels to extract relations (e.g., person-affiliation) from text. We briefly introduce the standard tree kernel and our proposed trace-tree
6.4.3.3.1 Standard Tree Kernel

In order to focus on the most relevant information to relations, a standard tree kernel is often defined on the minimum subtree that contains both entities in a parse tree. Node attributes \( V = \{ v_1, v_2, \ldots \} \) and the structural information are used in the tree kernel definition. First, we need to define two functions over tree nodes: a matching function \( m(p_i, p_j) \in \{0, 1\} \) and a similarity function \( s(p_i, p_j) \in [0, \infty) \). The matching function determines whether two nodes are matchable or not by comparing a subset of attributes, \( V^m \subseteq V \):

\[
m(p_i, p_j) = \begin{cases} 1, & \text{if } v'_k = v'_k, \forall v_k \in V^m \\ 0, & \text{otherwise} \end{cases}
\]

If two nodes are matchable, then the similarity function is computed by comparing the other attributes of nodes, \( V^s \subseteq V \):

\[
s(p_i, p_j) = \sum_{k \in \omega} \omega_k C(v'_k, v'_k)
\]

where \( 0 < \omega_k \leq 1 \) is the weight of attribute \( k \) and \( C(v'_k, v'_k) \) is a function that computes the compatibility between two values:

\[
C(v'_k, v'_k) = \begin{cases} 1, & \text{if } v'_k = v'_k \\ 0, & \text{otherwise} \end{cases}
\]

Then \( s(p_i, p_j) \) returns the weighted number of attributes values in common between \( p_i \) and \( p_j \).
For two relation instances $T_1$ and $T_2$, we define the tree kernel $K_T(T_1, T_2)$ that includes the similarity of the parent nodes and the similarity of the children.

$$K_T(T_1, T_2) = \begin{cases} 0, & \text{if } m(T_1.p, T_2.p) = 0 \\ s(T_1.p, T_2.p) + K_c(T_1.c, T_2.c), & \text{otherwise} \end{cases}$$

where the similarity function $K_c$ defined over children nodes $T.c$.

Let $i$ be a sequence of indices such that $i_1 \leq i_2 \leq \ldots \leq i_n$, and likewise for $j$. Let $d(i) = i_n - i_1 + 1$ and $l(i)$ be the length of $i$. For a relation instance $T$, let $T[i]$ denote a subsequence of children $T.c = \{T[i_1], \ldots, T[i_n]\}$. Then we have

$$K_c(T_1.c, T_2.c) = \sum_{i,j,l(i)=l(j)}^\lambda \lambda^{d(i)} \lambda^{d(j)} K(T_1[i], T_2[j])$$

where $0 < \lambda \leq 1$ is a decay factor that decreases the similarity between two sequences that are spread out within children sequences. For a pair of matching instances $T_1$ and $T_2$ such that $m(T_1.p, T_2.p) = 1$, the kernel function $K(T_1, T_2)$ needs to recursively compute the matching sequences of their children and accumulate the similarity scores. Tree kernels can be categorized into contiguous tree kernels and sparse tree kernels (Zelenko et al. 2003). A contiguous kernel only matches contiguous subsequences in children, $d(i) = l(i)$, whereas a sparse tree kernel allows non-contiguous sequences, $d(i) \geq l(i)$, which requires higher computational costs.

6.4.3.3.2 Trace-tree kernel

A relationship between two biomedical entities often follows one of the following
three patterns (Bunescu et al., 2006): (1) [B] *Between*: only words between the two entities express the relationship. Example: “X interacts with Y.” (2) [FB] *Fore-Between*: words before and between the two entities simultaneously express the relationship. Example: “…interaction of X and Y…” (3) [BA] *Between-After*: words between and after the two entities simultaneously express the relationship. Example: “X and Y interact…”

In the standard tree kernel, a relation instance is the minimum subtree that contains the two entities. If the relationship follows pattern [B], this minimum subtree often can capture the most relevant information in a parse tree. Unfortunately, in cases of [FB] or [BA], since some relation-indicating elements appear before or after the two entities, the minimum subtree may lose such contextual information in the remainder of the parse tree. For instance, Figure 6.3 shows the parse tree of a sentence: “There is a functional interaction between WT1 and p53.” The minimum subtree contains limited contextual information, whereas some important relation indicators such as “interaction” and “between” are not captured.
In order to catch such information without extending the size the subtree, we modified the tree kernel function by incorporating an auxiliary kernel. Specifically, we only focus on the trace from the root node of the minimum subtree to the root of the full parse tree. Such a trace is a sequence of head words of the branch nodes. For the relation instance shown in Figure 6.3, the trace is the word sequence: \[ \text{p53} - \text{between} - \text{interaction} - \text{is} - \text{is} \]. In a parse tree, the head word of a branch node indicates the key meaning of the corresponding sub-tree. Hence, such a trace captures more “global” context in the full parse tree in addition to the minimum subtree. Along this trace, branch nodes closer to the trace head (i.e., the root of minimum subtree) are assumed to be more relevant to the relation of interest. We imitate the sequence kernel and define a trace kernel function to measure the similarity between traces.

For trace \( s \), we denote by \( |s| \) the length of the trace \( s = s_1 \ldots s_{|s|} \). The sequence \( s[i:j] \) is
the subsequence \( s_i \ldots s_j \) of \( s \). Like the sequence kernel, the trace kernel compares the subsequences. By following the same notations, the feature mapping \( \phi \) for a trace \( s \) is given by defining \( \phi_u(s) \) for each subsequence \( u \) of length \( n \):

\[
\phi_u(s) = \sum_{i_u=s[i]} \lambda^{l(i)} \mu^{i_1}
\]

where \( 0 < \lambda, \mu \leq 1 \). These features measure the number of occurrences of subsequences in the trace \( s \) weighting them according to their lengths \( l(i) \) and distances from the trace end \( (i_1) \). \( \lambda \) is the decay factor to penalize subsequences with more interior gaps and therefore longer length; \( \mu \) is the decay factor to penalize subsequences starting farther from the trace head.

Hence, the inner product of the feature vectors for two traces \( s \) and \( t \) give a sum over all common subsequences weighted according to their frequency of occurrence, length, and position:

\[
K_n(s,t) = \sum_{u \in \Sigma^n} \phi_u(s) \cdot \phi_u(t) = \sum_{u \in \Sigma^n} \sum_{i_u=s[i]} \sum_{j_u=t[j]} \lambda^{l(i)+l(j)} \mu^{i_1+j_1}
\]

Like the sequence kernel, the trace kernel \( K_R \) accumulates \( K_n \) of different length to get the overall similarity between two traces:

\[
K_R(s,t) = \sum_n K_n(s,t)
\]

The trace-tree kernel \( K_{TR} \) combines the similarity between two sub-trees and the similarity between two traces, i.e., a standard tree kernel and a trace kernel:

\[
K_{TR}(x,y) = K_I(x,y) + K_R(x,y).
\]
6.4.4.3 Composite Kernels

The kernel functions above cast heterogeneous data representation into the common format of kernel matrices. This allows multiple kernel matrices to be combined into more complicated kernels using basic algebraic operations such as addition, multiplication, and exponentiation (Cristianini & Shawe-Taylor, 2000; Lanckriet et al., 2004). For example, given two kernels $K_1$ and $K_2$, the combined kernel $K(x, y) = \alpha_1 K_1(x, y) + \alpha_2 K_2(x, y)$ is also a valid kernel. Previous studies have shown that composite kernels could improve the performance of relation extraction (Culotta & Sorensen, 2004; Zhao & Grishman, 2005).

6.4.4 Learning & Evaluation

After kernel computation is performed on the training corpus, relation instances are denoted by a kernel matrix with a column of the instances’ labels (e.g., 1/true or 0/false). Next, a kernel machine such as SVM (Cristianini & Shawe-Taylor 2000) can be applied to learn a classification model that can identify relations in a testing set from future text. We can evaluate the performance of relation extraction model by comparing its prediction of relations on a testing set with their true labels.

6.5 Experimental Study

We conducted experiments to evaluate different kernel methods for relation extraction from biomedical literature.
6.5.1 Test-bed

We conducted experiments on a corpus of 200 randomly selected Medline abstracts, with similar scale of other studies (Marcotte et al., 2001; Bunescu et al., 2006). A biomedical scientist manually annotated biomedical entities such as genes, proteins, functions, and diseases. Each entity pair from the same sentence is regarded as a relation instance. In total, there are 8,071 relation instances, among which 2,156 were identified by the scientist as true relations. True relations were divided into four types: induction, suppression, non-directional association, and directional association, according to the taxonomy defined by (Marshall et al., 2006). Table 6.1 summarizes the biomedical relations from our corpus.

<table>
<thead>
<tr>
<th>Relation Types</th>
<th>Number</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>2,156</td>
<td>26.71%</td>
</tr>
<tr>
<td>1. induction</td>
<td>489</td>
<td>6.06%</td>
</tr>
<tr>
<td>2. suppression</td>
<td>244</td>
<td>3.02%</td>
</tr>
<tr>
<td>3. non-directional association</td>
<td>1,058</td>
<td>13.11%</td>
</tr>
<tr>
<td>4. directional association</td>
<td>365</td>
<td>4.52%</td>
</tr>
<tr>
<td>false</td>
<td>5,915</td>
<td>73.29%</td>
</tr>
<tr>
<td>Total</td>
<td>8,071</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

6.5.2 Evaluation Metrics

We use standard machine learning evaluation metrics, accuracy, precision, recall,
and F-measure, to evaluate the performances of relation extraction. Accuracy measures the overall correctness as defined as follows:

\[
\text{accuracy} = \frac{\text{# of all correctly identified instances}}{\text{total # of instances}}.
\]

Precision, recall, and F-measure evaluate the correctness for each class. Specifically, precision indicates the correctness of identified relations and recall indicates the completeness of identified relations. F-measure is the harmonic mean of precision and recall. These four measures are defined as follows

\[
\text{precision}(i) = \frac{\text{# of correctly identified instances for class } i}{\text{total # of instances identified as class } i},
\]

\[
\text{recall}(i) = \frac{\text{# of correctly identified instances for class } i}{\text{total # of instances in class } i},
\]

\[
\text{F-measure}(i) = \frac{2 \times \text{precision}(i) \times \text{recall}(i)}{\text{precision}(i) + \text{recall}(i)}.
\]

6.5.3 Hypotheses

Our study is aimed at testing the following three hypotheses:

H1. A tree kernel outperforms a sequence kernel and a word kernel for biomedical relation extraction.

H2. The trace-tree kernel outperforms the standard tree kernel.

H3. The composition of a word/sequence kernel and a tree kernel outperforms individual kernels alone.
6.5.4 Experimental Design

Following the experimental design in (Bunescu et al., 2006), we compare different kernels by performing two tasks: (1) relation detection: a binary classification of true and false relations, and (2) relation classification: a 4-class classification of the four relation types. All the 8,017 relation instances were used for relation detection task, while the 2,156 true relations were used to perform the 4-class classification.

In addition to a word kernel \((K_W)\), a sequence kernel \((K_S)\), and a standard tree kernel \((K_T)\), we also implemented the trace-tree kernel \(K_{TR}\) and four composite kernels: \(K_{TW} \triangleq (K_T + K_W)\), \(K_{TS} \triangleq (K_T + K_S)\), \(K_{TRW} \triangleq (K_{TR} + K_W)\), and \(K_{TRS} \triangleq (K_{TR} + K_S)\).

To parse the sentences in our corpus, we used the parser developed by (Lease & Charniak 2005), which was trained on biomedical corpora. Head words in branch nodes were assigned based on standard head word propagation rules (Collins, 1997), as shown in Appendix B. In a parse tree, we consider three node attributes: word, POS, and type. In the computation of tree kernels, \(I^m = \{\text{POS, type}\}\) are used in the matching function, while \(I^s = \{\text{word}\}\) is used in the similarity function and weight \(\omega_k\) is set to 1 (Zelenko et al., 2003). We chose a continuous tree kernel due to its less computational cost. The decay factors \(\lambda\) and \(\mu\) were set to 0.5 for all kernels.

We chose an SVM as the kernel machine due to its superior performance in many applications. Particularly, we used an SVM package, LibSVM.
(www.csie.ntu.edu.tw/~cjlin/libsvm), for learning because (1) it has been frequently used in previous studies (Culotta & Sorensen, 2004), (2) it supports multi-class classification, (3) it accepts customized kernels, and (4) it performs parameter selection for better performance.

We evaluated the eight kernels for relation detection and classification on our corpus. For each kernel, we used 90% of the relation instances as the training set to learn a classification model, and predicted the class labels of the remaining 10% instances as the testing set. We repeated this process 30 times by randomly sampling the datasets for statistical analysis.

6.5.5 Results and Discussion

The average performances of the eight kernels for relation detection and classification are summarized in Tables 6.2 and 6.3. Values in bold fonts are the best performances among the eight methods. For both classification tasks, $K_{TRS}$ achieved the best performance among all kernels. For relation detection, $K_{TRS}$ achieved 83.14% accuracy, 70.11% precision, 64.68% recall, and 67.23% F-measure. For relation classification, $K_{TRS}$ achieved 74.20% accuracy, 73.93% precision, 68.67% recall, and 70.65% F-measure.
Table 6.2 Kernel Methods for Biomedical Relation Detection

<table>
<thead>
<tr>
<th>Types</th>
<th>Kernels</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>$K_W$</td>
<td>73.10%</td>
<td>49.67%</td>
<td>32.33%</td>
<td>39.09%</td>
</tr>
<tr>
<td>Sequence</td>
<td>$K_S$</td>
<td>78.02%</td>
<td>60.72%</td>
<td>50.69%</td>
<td>55.21%</td>
</tr>
<tr>
<td>Tree</td>
<td>$K_T$</td>
<td>79.13%</td>
<td>62.19%</td>
<td>56.56%</td>
<td>59.16%</td>
</tr>
<tr>
<td></td>
<td>$K_{TR}$</td>
<td>80.52%</td>
<td>64.64%</td>
<td>60.22%</td>
<td>62.31%</td>
</tr>
<tr>
<td>Composite</td>
<td>$K_{TW}$</td>
<td>80.21%</td>
<td>63.91%</td>
<td>60.15%</td>
<td>61.92%</td>
</tr>
<tr>
<td></td>
<td>$K_{TS}$</td>
<td>81.97%</td>
<td>67.54%</td>
<td>63.06%</td>
<td>65.17%</td>
</tr>
<tr>
<td></td>
<td>$K_{TRW}$</td>
<td>81.61%</td>
<td>67.12%</td>
<td>61.62%</td>
<td>64.21%</td>
</tr>
<tr>
<td></td>
<td>$K_{TRS}$</td>
<td>83.14%</td>
<td>70.11%</td>
<td>64.68%</td>
<td>67.23%</td>
</tr>
</tbody>
</table>

Table 6.3 Kernel Methods for Biomedical Relation Classification

<table>
<thead>
<tr>
<th>Types</th>
<th>Kernels</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>$K_W$</td>
<td>66.42%</td>
<td>62.97%</td>
<td>59.91%</td>
<td>60.99%</td>
</tr>
<tr>
<td>Sequence</td>
<td>$K_S$</td>
<td>72.67%</td>
<td>70.26%</td>
<td>65.62%</td>
<td>67.30%</td>
</tr>
<tr>
<td>Tree</td>
<td>$K_T$</td>
<td>69.09%</td>
<td>71.13%</td>
<td>60.44%</td>
<td>63.83%</td>
</tr>
<tr>
<td></td>
<td>$K_{TR}$</td>
<td>69.18%</td>
<td>69.62%</td>
<td>61.76%</td>
<td>64.46%</td>
</tr>
<tr>
<td>Composite</td>
<td>$K_{TW}$</td>
<td>71.16%</td>
<td>71.39%</td>
<td>64.13%</td>
<td>66.67%</td>
</tr>
<tr>
<td></td>
<td>$K_{TS}$</td>
<td>73.60%</td>
<td>73.91%</td>
<td>66.74%</td>
<td>69.34%</td>
</tr>
<tr>
<td></td>
<td>$K_{TRW}$</td>
<td>70.76%</td>
<td>69.48%</td>
<td>65.32%</td>
<td>66.86%</td>
</tr>
<tr>
<td></td>
<td>$K_{TRS}$</td>
<td>74.20%</td>
<td>73.93%</td>
<td>68.67%</td>
<td>70.65%</td>
</tr>
</tbody>
</table>

Furthermore, we conducted pair-wise $t$-tests on the four evaluation metrics to test our hypotheses. The p-values of hypotheses testing for relation detection and classification are presented in Tables 6.4 and 6.5, where p-values with * and ** indicates significant difference at the level of $\alpha = 0.05$ and 0.01, respectively. The underlined p-values mean that the results contradict the hypotheses. Overall, most hypotheses were supported by our experiments.
Table 6.4 Hypotheses Testing for Biomedical Relation Detection

<table>
<thead>
<tr>
<th>No.</th>
<th>Hypotheses</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>$K_S &gt; K_W$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_T &gt; K_W$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_T &gt; K_S$</td>
<td>0.0005^{**}</td>
<td>0.0304*</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td>H2</td>
<td>$K_{TR} &gt; K_T$</td>
<td>$&lt;0.0001^{**}$</td>
<td>0.0002^{**}</td>
<td>0.0001^{**}</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td>H3</td>
<td>$K_{TW} &gt; K_W$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TW} &gt; K_T$</td>
<td>0.0003^{**}</td>
<td>0.0080^{**}</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TS} &gt; K_S$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TS} &gt; K_T$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TRW} &gt; K_W$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TRW} &gt; K_{TR}$</td>
<td>0.0006^{**}</td>
<td>0.0005^{**}</td>
<td>0.0417*</td>
<td>0.0018^{**}</td>
</tr>
<tr>
<td></td>
<td>$K_{TRS} &gt; K_S$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TRS} &gt; K_{TR}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
</tbody>
</table>

Table 6.5 Hypotheses Testing for Biomedical Relation Classification

<table>
<thead>
<tr>
<th>No.</th>
<th>Hypotheses</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>$K_S &gt; K_W$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_T &gt; K_W$</td>
<td>0.0008^{**}</td>
<td>$&lt;0.0001^{**}$</td>
<td>0.2899</td>
<td>0.0018^{**}</td>
</tr>
<tr>
<td></td>
<td>$K_T &gt; K_S$</td>
<td>$&lt;0.0001^{**}$</td>
<td>0.2109</td>
<td>$&lt;0.0001^{**}$</td>
<td>0.0006^{**}</td>
</tr>
<tr>
<td>H2</td>
<td>$K_{TR} &gt; K_T$</td>
<td>0.4556</td>
<td>0.0969</td>
<td>0.1053</td>
<td>0.2768</td>
</tr>
<tr>
<td>H3</td>
<td>$K_{TW} &gt; K_W$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TW} &gt; K_T$</td>
<td>0.0008^{**}</td>
<td>0.3816</td>
<td>0.0011</td>
<td>0.0003^{**}</td>
</tr>
<tr>
<td></td>
<td>$K_{TS} &gt; K_S$</td>
<td>0.1269</td>
<td>0.0007^{**}</td>
<td>0.1518</td>
<td>0.0303*</td>
</tr>
<tr>
<td></td>
<td>$K_{TS} &gt; K_T$</td>
<td>$&lt;0.0001^{**}$</td>
<td>0.0048^{**}</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TRW} &gt; K_W$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
<tr>
<td></td>
<td>$K_{TRW} &gt; K_{TR}$</td>
<td>0.0340*</td>
<td>0.4526</td>
<td>0.0010^{**}</td>
<td>0.0158*</td>
</tr>
<tr>
<td></td>
<td>$K_{TRS} &gt; K_S$</td>
<td>0.0390*</td>
<td>0.0005^{**}</td>
<td>0.0040^{**}</td>
<td>0.0013^{**}</td>
</tr>
<tr>
<td></td>
<td>$K_{TRS} &gt; K_{TR}$</td>
<td>$&lt;0.0001^{**}$</td>
<td>0.0002^{**}</td>
<td>$&lt;0.0001^{**}$</td>
<td>$&lt;0.0001^{**}$</td>
</tr>
</tbody>
</table>

H1. $K_T > K_S > K_W$

For both relation detection and classification, the sequence kernel $K_S$ significantly
outperformed the word kernel $K_W$ for all the four metrics with p-values less than 0.0001. This is because a sequence kernel captures not only the common words in two relation instances but also the linear sequences of words in sentences. The linear sequences are shown to provide richer contextual information than a “bag-of-words.” For relation detection, the standard tree kernel $K_T$ was shown to significantly outperform both $K_W$ and $K_S$ with most p-values less than 0.01 except for precision of $K_S < K_T$ ($p = 0.0304 < 0.05$).

For relation classification, $K_T$ also outperformed $K_W$ significantly except for recall ($p = 0.2899$). However, the hypothesis that $K_T$ outperforms $K_S$ for relation classification was not supported. By contrast, our experiments showed that $K_S$ significantly outperformed $K_T$ on accuracy, recall, and F-measure at the level of $\alpha=0.01$. The tree kernel captures rich structural information in a subtree. Such information is effective to differentiate true relations from false ones. However, given a true relation, the tree structure seems not as effective as lexical patterns (word sequences) to distinguish different types of relations.

H2. $K_{TR} > K_R$.

For relation detection, our trace-tree kernel $K_{TR}$ achieved 80.52% accuracy, 64.64% precision, 60.22% recall, and 62.31% F-measure, which significantly outperformed $K_T$ ($p < 0.01$). This confirmed hypothesis H2. The trace kernel captures main contextual information missing in a minimum subtree therefore it improves the relation detection performance. As discussed before, structural information is not as effective for relation
classification as for relation detection. This explained why, as compared to $K_T$, $K_{TR}$ did not improve the performance of relation classification ($p > 0.05$ for all four measures).

H3. A composite kernel > individual kernels

Kernel functions convert different structural representations into the same format of kernel matrix, which enables the composition of multiple kernels. In our experiments, for relation detection, except that the recall of $K_{TRW}$ was significantly higher than that of $K_{TR}$ at $\alpha=0.05$, all the other comparisons were consistent with our hypotheses and showed significant differences at the level with $p$-values less than 0.01. For relation classification, we observed that in most cases, composite kernels ($K_{TW}$, $K_{TS}$, $K_{TRW}$, and $K_{TRS}$) outperformed their sub-kernels at the significance level of $\alpha=0.05$ or $\alpha=0.01$. We only found four comparisons that did not show significant difference: $K_{TW} > K_T$ in precision ($p = 0.3816$), $K_{TS} > K_S$ in accuracy ($p = 0.1269$) and recall ($p = 0.1518$), and $K_{TRW} > K_{TR}$ in precision (contradictory to the hypothesis but $p = 0.4526$). In these four cases, the composite kernels achieved performances comparable to the sub-kernels in certain evaluation measures. Notably, their average F-measures of composite kernels were all significantly higher than that of the sub-kernels.

We can draw the following conclusions from the experiments. (1) The parse tree kernel captures richer structural information, which helped identify relations but did not show significant benefits for relation classification as compared to the sequence kernel.
(2) Our modified tree kernel with a trace kernel can capture more information and help identify relations from text. (3) By integrating multiple types of information, composite kernels can improve the performance of relation detection and classification.

6.6 Conclusions and Future Directions

In summary, we developed a framework of kernel-based learning for biomedical relation extraction. We conducted an experimental study to compare different kernel methods on a corpus of biomedical literature abstracts. The results showed excellent performance of tree kernel for relation extraction. Particularly, we modified the standard tree kernel by incorporating a trace kernel, which can capture richer contextual information and improves the relation extraction performance. In addition, the good performance of composite kernels demonstrated the effectiveness of integrating contextual information represented in different data types.

We will extend our study in the following directions. (1) Kernel functions in most studies compare words based on exact match. We plan to redesign kernels that can capture semantic similarity between words to improve the performance of relation extraction. (2) In our current study named entities are manually tagged by domain experts before relation extraction. We will incorporate an automated named entity recognition module into our relation extraction framework and re-examine the overall performance.
CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS

The advance of information technologies has brought massive amounts of digitized data to various application areas. Knowledge discovery techniques have been applied to automatically search for patterns so as to support decision making processes. In order to improve the representation, modeling, and analysis of various data, my dissertation presents a framework of feature engineering for knowledge discovery. In particular, I explored techniques for constructing, selecting, and consolidating features and examined them in fields such as identity matching, cybercrime investigation, cancer diagnosis, plant sciences, and information extraction.

This chapter concludes my dissertation by summarizing the major contributions, discussing the relevance to research in management information systems, and proposing future research directions.

7.1 Contributions

This dissertation involves five studies of feature engineering and makes several contributions to the knowledge discovery discipline. This section summarizes their theoretical, technical, and empirical contributions, respectively.

7.1.1 Theoretical Contributions

This dissertation presents a framework of feature engineering for knowledge
discovery. This framework addresses three major challenges in feature engineering: feature construction, feature selection, and feature consolidation. By identifying the significant features, we can improve the effectiveness, efficiency, and interpretability of knowledge discovery.

Chapter 2 explores the personal and social aspects of identity. Features are constructed to describe personal information and social behavior of individuals for identity matching in large databases. Social features are shown to provide complementary information to distinguish individuals.

Chapter 3 extends the traditional stylometric analysis and derives four types of writing style features for online messages: lexical, syntactic, structural, and content-specific features. These four types of features provide a theoretical foundation for describing the writeprints for authorship identification of online messages.

Chapter 4 presents a taxonomy of feature selection techniques of two dimensions: evaluation criterion and generation process. This taxonomy can provide a theoretical basis for choosing feature selection methods in data mining.

7.1.2 Technical Contributions

In Chapter 2 we propose to use a probabilistic relational model (PRM)-based approach to construct features from relational database structure. Both personal and social identity features can be derived for identity matching in databases.
Chapter 3 presents a framework for authorship identification of online messages. Four types of features are extracted to distinguish online authorship. Moreover, a GA-based feature subset selection model is developed to identify key writeprint features.

Chapter 4 presents a framework of optimal search-based feature subset selection. Specifically, we propose to use tabu search for feature subset selection from a high dimensional feature space and show promising performances.

Chapter 5 presents a framework of integrating relations from heterogeneous data sources. A two-layered Bayesian network approach is used to evaluate and combine relations extracted from different sources.

In Chapter 6 we develop and compare different kernel methods for relation extraction from text. In particular, we modified the standard tree kernel by incorporating a trace kernel to capture the global information in the parse tree. This study also shows the power of kernel composition by combining multiple sub-kernels for improved relation extraction.

7.1.3 Empirical Contributions

In Chapter 2 we examined the PRM-based approach to construct personal and social features for identity matching in a criminal dataset. In Chapter 3 we propose authorship identification approach based on analyzing writeprint features of online messages in cybercrime investigation. These approaches have shown promising performances and can
contribute significantly to security applications.

Chapter 4 focuses on identifying marker genes from microarray data for cancer classification. The experiment findings can potentially help biologists uncover marker genes for cancer diagnosis and cure development. In Chapter 5 we combined gene relations from diverse data sources into a genome-wide gene functional network. This integrated network can bring more insights into unknown plant gene functions.

In Chapter 6 we examined different kernel methods for relation extraction on a biomedical literature corpus and our experiments showed excellent performance. These techniques can also be applied to other domains such as e-commerce and e-learning to facilitate information seeking and access.

7.2 Relevance to MIS Research

Hevner, March, Park, & Ram (2004) categorized the research in the Information Systems discipline into two paradigms: behavioral science and design science. This dissertation follows the design science paradigm because it is aimed at creating new and innovative IT artifacts so as to reduce of the gap between growing amounts of data and human beings’ capabilities to understand it. The proposed feature engineering techniques largely rely on machine learning algorithms that automatically generate decision models with minimal human intervention. Constructing, selecting and consolidating significant features can not only help improve the effectiveness and efficiency of knowledge
discovery, but also bring fresh insights into application domains such as social networks, online communities, biological systems, and so on.

Although the test-beds in these studies are from various application domains such as crime investigation, cybercrime, cancer diagnosis and plant science, our proposed framework for feature engineering is generic and widely applicable. One direct application domain of this research is business intelligence (BI) which provides technologies for data collection, processing, and analysis. BI systems help companies have a more comprehensive knowledge of the factors or features affecting their business, such as metrics on sales, production, and internal operations, and they can help companies make better business decisions (Wikipedia, April 20th, 2007). Information for BI analysis can be extracted from various types of resources: either internal or external, either structured or unstructured, either simple or complex, etc. Successful BI analysis requires identifying significant features about employees, customers, suppliers, products, market demands, business processes, transactions, etc. Feature engineering techniques discussed in this dissertation can also be applied to construct, select, and consolidate various features for BI.

In particular, the proposed PRM-based approach can be applied for matching identities in organizational databases in cases of mergers or acquisition. Both personal and social features can provide evidence for identity matching. In e-commerce systems,
we can also follow the same methodology and derive features about customers, products, and transactions so as to provide better recommendations which are potential linkages between a customer and a product. In addition, a successful BI system needs to analyze information from various data sources including the Internet. Online articles such as business news articles, product reviews, forums, and weblogs are becoming popular and important sources of business intelligence. The authorship analysis approaches can help us identify the online users and further understand their behavior so as to support business decision making. Moreover, we can apply feature selection techniques to identify critical factors that affect various business objectives, such as sales and stock prices, and help managers make better business decisions. Our proposed kernel-based learning methods can be used to automatically extract relations between business entities (e.g., people, products, and companies) from online news articles for business intelligence analysis. In addition, our proposed Bayesian framework of relation integration is potentially useful for constructing social networks among business entities by consolidating information from various data sources.

7.3 Future Directions

My dissertation addresses various challenges faced in feature engineering for knowledge discovery. The proposed tools have shown promising performance in these application domains. This work can be extended in the following directions to further
improve the effectiveness, efficiency, and interpretability of knowledge discovery.

(1) Features may have interactions among each other and these interactions can significantly affect the knowledge discovery performances. How to capture the feature interactions and incorporate them into the feature engineering process is an interesting topic. For example, if we can extract gene interactions from an existing knowledge base or literature text and incorporate them into gene selection, such information can help guide the search in the feature space and reveal the biological relevance of selected marker genes. (2) As shown in Chapter 6, kernel methods can capture rich information from various complex data representations and cast them into the same format of kernel matrices. Kernel methods are regarded as an effective alternative to feature methods for knowledge discovery, especially for domains where features are difficult to define or extract. For example, graph kernels can be applied to identity matching based on social networks, gene function classification based on gene networks, and so on. (3) Approaches presented in my dissertation, such as optimal search-based feature subset selection and kernel-based learning, can achieve satisfactory performances but often require high computational cost. Reducing the complexity and improving the efficiency is still a challenging task. (4) As discussed in the previous section, I am interested in expanding and examining the proposed feature engineering approaches in other domains, especially business applications.
APPENDIX A FUNCTION WORD FEATURES FOR AUTHORSHIP ANALYSIS

English function words in our feature set:

a  between  in  nor  some  upon
about  both  including  nothing  somebody  us
above  but  inside  of  someone  used
after  by  into  off  something  via
all  can  is  on  such  we
although  coz  it  once  than  what
am  do  its  one  that  whatever
among  down  latter  onto  the  when
an  each  less  opposite  their  where
and  either  like  or  them  whether
another  enough  little  our  these  which
any  every  lots  outside  they  while
anybody  everybody  many  over  this  who
anyone  everyone  me  own  those  whoever
anything  everything  more  past  though  whom
are  few  most  per  through  whose
around  following  much  plenty  till  will
as  for  must  plus  to  with
at  from  my  regarding  toward  within
be  have  near  same  towards  without
because  he  need  several  under  worth
before  her  neither  she  unless  would
behind  him  no  should  unlike  yes
below  i  nobody  since  until  you
beside  if  none  so  up  your

Chinese function words in our feature set:

我 偶 你 他 我们 偶们 俺们 咱们 你们 他们 的 地 得 着 了 过 啊 呀 哎 呢
吧 哦 喔 呼 呼呼 也 也许 都 又 是 就 一个 以后 然后 然而 虽然 但是 到
底 随着 不然 后来 之后 总之 直到 往往 其实 反正 觉得 我想 认为 为什么 什
么 怎么 怎样 难道 特别 却是 确实 的确 要不
## APPENDIX B TREE HEAD TABLE

<table>
<thead>
<tr>
<th>Non-terminal</th>
<th>Category</th>
<th>List of Head</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJP</td>
<td>right</td>
<td>% QP JJ VBN VBG ADJP $ JJR JJS DT FW **** RBR RBS RB **</td>
</tr>
<tr>
<td>ADVP</td>
<td>left</td>
<td>RBR RB RBS FW ADVP CD **** JJR JJS JJ **</td>
</tr>
<tr>
<td>CONJP</td>
<td>left</td>
<td>CC RB IN **</td>
</tr>
<tr>
<td>FRAG</td>
<td>left</td>
<td>**</td>
</tr>
<tr>
<td>INTJ</td>
<td>right</td>
<td>**</td>
</tr>
<tr>
<td>LST</td>
<td>left</td>
<td>LS : **</td>
</tr>
<tr>
<td>NAC</td>
<td>right</td>
<td>NN NNS NNP NNPS NP NAC EX $ CD QP PRP VBG JJ JJS JJR ADJP FW **</td>
</tr>
<tr>
<td>NP</td>
<td>right</td>
<td>N* EX $ CD QP PRP VBG JJ JJS JJR ADJP DT FW RB SYM PRPS **</td>
</tr>
<tr>
<td>NPS</td>
<td>right</td>
<td>NN NNS NNP NNPS NP NAC EX $ CD QP PRP VBG JJ JJS JJR ADJP FW SYM **</td>
</tr>
<tr>
<td>PNP</td>
<td>right</td>
<td>**</td>
</tr>
<tr>
<td>PP</td>
<td>left</td>
<td>IN TO FW **</td>
</tr>
<tr>
<td>PRN</td>
<td>left</td>
<td>**</td>
</tr>
<tr>
<td>PRT</td>
<td>left</td>
<td>RP **</td>
</tr>
<tr>
<td>QP</td>
<td>right</td>
<td>CD NCD % QP JJ JJR JJS DT **</td>
</tr>
<tr>
<td>RRC</td>
<td>left</td>
<td>VP NP ADVP ADJP PP **</td>
</tr>
<tr>
<td>S</td>
<td>right</td>
<td>VP VP/S S SBAR ADJP UCP NP **</td>
</tr>
<tr>
<td>SBAR</td>
<td>right</td>
<td>S SQ SINV SBAR FRAG X **</td>
</tr>
<tr>
<td>SBARQ</td>
<td>right</td>
<td>SQ S SINV SBARQ FRAG X **</td>
</tr>
<tr>
<td>SINV</td>
<td>right</td>
<td>S VP VBZ VBD VBP VB SINV ADJP NP **</td>
</tr>
<tr>
<td>SQ</td>
<td>right</td>
<td>VP VBZ VBD VBP VB MD SQ **</td>
</tr>
<tr>
<td>UCP</td>
<td>left</td>
<td>**</td>
</tr>
<tr>
<td>VP</td>
<td>left</td>
<td>VBD VBN MD VBZ TO VB VP VBG VBP ADJP NP **</td>
</tr>
<tr>
<td>VP/S</td>
<td>left</td>
<td>VBD VBN MD VBZ TO VB VP VBG VBP ADJP NP **</td>
</tr>
<tr>
<td>WHADJP</td>
<td>right</td>
<td>JJ ADJP **</td>
</tr>
<tr>
<td>WHADVP</td>
<td>left</td>
<td>WRB **</td>
</tr>
<tr>
<td>WHNP</td>
<td>right</td>
<td>WDT WP WPS WHADJP WHPP WHNP **</td>
</tr>
<tr>
<td>WHPP</td>
<td>left</td>
<td>IN TO FW **</td>
</tr>
<tr>
<td>X</td>
<td>left</td>
<td>**</td>
</tr>
</tbody>
</table>


Note: “Category” indicates where to start to look for a head (left is for head-initial categories; and right is for head-final categories).
REFERENCES


Resolution. Paper presented at the SIAM International Conference on Data Mining, June.


Intelligence}, 151, 155-176.


Sun, A., Lim, E.-P., Ng, W.-K., & Srivastava, J. (2004). Blocking Reduction Strategies in Hierarchical Text Classification. *IEEE Transactions on Knowledge and Data Engineering*, 16(10), 1305-1308.


Association, Alexandria, Virginia.


Journal of Biological Chemistry, 278(9), 7580-7590.


