ENTITY MATCHING FOR INTELLIGENT INFORMATION INTEGRATION

By

Gang Wang

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

2 0 0 6

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Gang Wang

entitled Entity Matching for Intelligent Information Integration,

and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

_____ Date: 07/14/2006
Hsinchun Chen

_____ Date: 07/14/2006
Jay F. Nunamaker

_____ Date: 07/14/2006
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: 07/14/2006
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.

SIGNED: Gang Wang

ACKNOWLEDGEMENTS

I gratefully thank my advisor, Dr. Hsinchun Chen, for his advice and guidance throughout my doctoral study. I believe what I have learned in the Artificial Intelligence Lab under his direction would benefit the rest of my academic career. I also thank my major committee members, Dr. Jay F. Nunamaker, Jr., Dr. J. Leon Zhao, and Dr. Zhu Zhang, and my minor committee member in the Department of Electrical and Computer Engineering, Dr. Jerzy W. Rozenblit, for their guidance and encouragement. A special thank note goes to my academic advisor at Louisiana State University, Dr. T. Warren Liao, for his farseeing advice on my future career.

My dissertation has been partly supported by National Science Foundation and Central Intelligence Agency (#0429364, "COPLINK Center: Social Network Analysis and Identity Deception Detection for Law Enforcement and Homeland Security," 2004 – 2006, and #9983304 Supplement, "COPLINK Center: Detecting Identity Concealment," 2004 – 2006). My work at the Artificial Intelligence Lab has been supported by many colleagues. I would like to thank the following personnel, Det. Tim Petersen, Lt. Jenny Schroeder, and Dan Casey in the Tucson Police Department, and Dr. Daniel Zeng, Cathy Larson, Homa Atabakhsh, Shailesh Joshi, Mark Patton, and Kira Joslin in the Artificial Intelligence Lab.

It is my pleasure to work with an excellent group of colleagues in the department. I would like to thank Michael Chau, Bin Zhu, Gondy Leroy, Chienting Lin, Thian-Huat Ong, Zan Huang, Jennifer Xu, Jinwei Cao, Limin Zhang, Byron Marshall, Daniel McDonald, Wingyan Chung, Yiwen Zhang, Ming Lin, Harry Wang, Jialun Qin, Yilu Zhou, Jason Li, Rong Zheng, Wei Chang, Wei Wei, Jun Li, Sherry Sun, Ming Yuan, Ling Zhu, Xin Li, Siddharth Kaza, Rob Schumaker, Ahmed Abbasi, Danning Hu, Hsin-Min Lu, Nichalin Suakkaphong, Manlu Liu, Aaron Sun, Ping Yan, Tianjun Fu, and Yuan Wang.

Lastly but most importantly, I dedicate this dissertation to my family. I would like to thank my wife, Ping, for her companion, assistance, understanding, patience, and trust throughout my doctoral study. With all my heart I am also grateful to my parents and my sister, for their everlasting love and care. Without their constant support and encouragement this work would be impossible.

TABLE OF CONTENTS

TABLE OF CONTENTS - *Continued*

TABLE OF CONTENTS - *Continued*

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

Due to the rapid development of information technologies, especially the network technologies, business activities have never been as integrated as they are now. Business decision making often requires gathering information from different sources. This dissertation focuses on the problem of entity matching, associating corresponding information elements within or across information systems. It is devoted to providing complete and accurate information for business decision making.

Three challenges have been identified that may affect entity matching performance: feature selection for entity representative, matching techniques, and searching strategy. This dissertation first provides a theoretical foundation for entity matching by connecting entity matching to the similarity and categorization theories developed in the field of cognitive science. The theories provide guidance for tackling the three challenges identified. First, based on the feature contrast similarity model, we propose a case-study-based methodology that identifies key features that uniquely identify an entity. Second, we propose a record comparison technique and a multi-layer naïve Bayes model that correspond respectively to the deterministic and the probability response selection models defined in the categorization theory. Experiments show that both techniques are effective in linking deceptive criminal identities. However, the probabilistic matching technique is preferable because it uses a semi-supervised learning method, which requires less human intervention during training. Third, based on the prototype access assumption proposed in the categorization theory, we apply an adaptive detection algorithm to entity matching so

that efficiency can be greatly improved by the reduced search space. Experiments show that this technique significantly improves matching efficiency without significant accuracy loss.

Based on the above findings we developed the Arizona IDMatcher, an identity matching system based on the multi-layer naïve Bayes model and the adaptive detection method. We compare the proposed system against the IBM Identity Resolution tool, a leading commercial product developed using heuristic decision rules. Experiments do not suggest a clear winner, but provide the pros and cons of each system. The Arizona IDMatcher is able to capture more true matches than IBM Identity Resolution (i.e., high recall). On the other hand, the matches identified by IBM Identity Resolution are mostly true matches (i.e., high precision).

CHAPTER 1: INTRODUCTION

Due to the rapid development of information technologies, especially the network technologies, business activities have never been as integrated as they are now. Business decision making often requires gathering information from different sources. However, many information systems were developed without the consideration of being compatible to one another (i.e., information islands). Other problems such as poor data quality and information overload may also arise during information retrieval and integration across heterogeneous information systems. Redman (1998) reported that data error rates in typical enterprises could be as high as 30%. Trout (1997) provided statistics that more information has been produced in the last thirty years than in the previous five thousand years. These problems raise new challenges for information extraction and knowledge management in various organizational practices such as health care management, business analysis and management, and crime investigation in law enforcement (Khoumbati, Themistocleous, & Irani, 2006; Kim, Choi, Hong, Kim, & Lee, 2003; Seifert, 2004).

Research efforts for providing interoperability across heterogeneous information systems have been focusing on two fundamental areas: schema heterogeneity and data heterogeneity (Kim & Seo, 1991; Levitin & Redman, 1995). As the name implies, schema heterogeneity refers to the problem of information systems that are designed differently and thus have incompatible system schemas. Schema heterogeneity often contains two types of conflicts. First, independent information system design defines

different structures (e.g., tables and attributes) for the same information. For example, an address is stored as one attribute value in one system while it can be separated into as many as five attribute values in another system. Second, schema elements that are semantically equivalent across different information systems may have different specification such as attribute name, data type, and constraints. Data heterogeneity refers to the problem of disagreeing values between corresponding data elements in different information systems. There are two main reasons for this. First, data maintained in an information system may be incorrectly entered, obsolete, or acquired from inaccurate data sources. Second, different representations for the same data element may exist. For example, different words such as “Texas” and “TX” can be used to express the same state information. When those problems occur in record identifiers, multiple representations for the same entity may exist in a system. When integrating information from different systems, data heterogeneity causes difficulty in determining correspondence between data elements.

1.1 Research on Schema Integration

Schema integration, also known as schema matching, takes two system schemas as input and produces a mapping between their elements that correspond semantically to each other (Rahm & Bernstein, 2001). There is abundant research focusing on schema integration. Spaccapietra et al. (1992) classified schema integration techniques into two categories. A manual integration process is popular in traditional database design methodology to find commonalities and discrepancies between input schemas. Database administrators (DBA) identify modifications or restructuring transformations

to input schemas so that they can be translated into a “superview,” which is a supertype schema generalized from both input schemas. Examples of manual approaches include SUPERVIEWS (Motro, 1987), MULTIBASE (Landers & Rosenberg, 1982), PRECI distributed database system (Deen, Amin, & Taylor, 1987), and MERMAID (Templeton et al., 1987). The manual approaches are time-consuming and costly. Various techniques have been proposed to investigate the feasibility of automating the schema integration process. They use automated data analysis techniques, such as rule-based systems and machine learning algorithms, to generate recommendations on schema element correspondences. DBAs’ responsibility is greatly reduced by merely reviewing the recommendations. The proposed machine learning algorithms include clustering (Cliffton, Housman, & Rosenthal, 1997), correlation and regression analysis (Fan, Lu, Madnick, & Cheung, 2001, , 2002), and mutual information (Zhao & Soofi, 2006). Automatic integration techniques have better portability than manual approaches because they can be repeatedly applied to different schema integration tasks with the least modification and customization.

1.2 Research on Entity Matching

Research focusing on data heterogeneity has attracted less attention than schema integration. Techniques proposed for solving the data heterogeneity problem are sporadic and unsystematic. For example, this problem has been called by different names such as entity matching (Dey, Sarkar, & De, 1998, , 2002), instance identification (Wang & Madnick, 1989), merge/purge (Hernandez & Stolfo, 1995, , 1998), object isomerism (Chen, Tsai, & Koh, 1996), and record linkage (Fellegi &

Sunter, 1969). Throughout this dissertation we use the term entity matching to refer to the process of determining if an entity in one data source is the same as another entity in the same or another data source. However, our definition is different from the one given by Dey et al. (1998) in two ways: our definition removes the assumption that one entity in a data source can map to one and only one entity in another data source and allows finding duplicate entity representations within a data source.

Entity matching techniques face challenges in the following three areas: feature selection for entity representation, searching strategy, and matching techniques (Verykios, Elmagarmid, & Houstis, 2000).

Feature selection is the process of choosing a set of common features, the combined values of which are expected to uniquely identify an entity. Existing entity matching techniques are inefficient because they lack a feature selection process and basically use all the common features available in two data sources. If the number of features is large the computation load could be quite high. In most cases entity features are not equally informative and reliable in determining matching entities. For example, name is often considered more informative than height in determining a personal identity. Therefore, there is a need to identify an appropriate feature set such that entity matching can proceed in an efficient and accurate way.

Entity matching search strategies reduce search space by identifying a subset of entities so that matching only needs to be performed within the subset. Due to the development of automated transaction processes, organizations have accumulated a

large volume of data. To provide business decision support in real time, entity matching has to be done in a timely manner.

Matching techniques compare the corresponding feature values of two entities and determine whether or not they match. The matching process involves uncertainty due to the lack of reliable unique identifiers and poor data quality. Various techniques have been proposed to automatically predict matching decisions among entities (Brown & Hagen, 2003; Dey et al., 1998, , 2002; Fellegi & Sunter, 1969; Jaro, 1989). However, they are often application driven and lack theoretical foundations.

In the next section we link similarity and categorization theories developed in the field of cognitive science to entity matching and then propose a theoretical research framework for entity matching techniques in the following section.

1.3 Theoretical Foundations for Entity Matching

Before the invention of automatic entity matching techniques, human experts often made entity matching decisions based upon the perceived similarities between two entities. Studies on perceived similarities were developed in the field of cognitive science. Geometric models were first used to explain human similarity judgment. Geometric models represent each object (i.e., an entity in our case) as a point in a multidimensional space. Assuming an Euclidean space, the perceived similarity between two objects is considered inversely related to the metric distance between objects in the multidimensional metric space (Shepard, 1962a, , 1962b, , 1963). Because of their reliance on distance, perceived similarities estimated by geometric models must satisfy certain distance axioms including symmetry (the similarity

between entity *A* and entity *B* is equivalent to that between *B* and *A*) and triangle inequality (the sum of the dissimilarity between entities *A* and *B* and that between *B* and *C* is greater than the dissimilarity between *A* and *C*) among others. However, those axioms might not be true in certain scenarios. Ashby and Perrin (1988) provided an example where triangle inequality was clearly inappropriate: "a flame (*A*) is similar to the moon (*B*) because they both appear luminous, and the moon is similar to a ball (*C*) because they are both round. However, a flame and a ball are very dissimilar."

Tversky (1977) proposed an alternative theory, the feature contrast model, in one of his seminal papers. Given two objects being compared, the model assumes that the perceived similarity increases as a function of the common feature values and decreases as a function of the distinctive feature values. The similarity is defined as

$$Similarity(S_A, S_B) = \vartheta f(S_A \cap S_B) - \alpha f(S_A - S_B) - \beta F(S_B - S_A),$$

where θ, α, and β are non-negative parameters. This model is considered superior to geometric models because it is similar to the human reasoning process.

Entity matching is considered a categorization task in cognitive science. Matching entities are categorized into the same category. Ashby and Alfonso-Reese (1995) proposed that all categorization models made assumptions about three things. First, the objects need to have a numeric representation. In geometric similarity models, an object is represented as a point in a multidimensional space (MDS). We can use a vector to denote the coordinates of an object in the MDS. In feature contrast models, an object is represented as a set of features. Although features can be non-numeric, we may represent feature value comparisons as numerical values. For

example, given two string feature values, a string comparison algorithm such as Edit Distance (Levenshtein, 1966) can compute a distance/similarity score. The second assumption is made on how information is accessed and evaluated for categorization. Using different access models may affect the efficiency of categorization. For example, the exemplar model (Nosofsky, 1986) compares a subject entity to every member in each category. The prototype model (Rosch, 1973, , 1977) only compares a subject entity to the prototype of each category. The prototype is defined as the most typical or representative category member. Therefore, the prototype model is more efficient than the exemplar model. The third assumption is about categorization response selection models. A deterministic model assumes that the same perceptual information on different trials always prompts the same categorization response. For example, if the perceived similarity between two entities is greater than a matching threshold, the deterministic model always makes a matching decision with certainty (i.e., with the probability of one). A probabilistic model assumes that one always guesses a categorization response in a sophisticated manner. Given the same perceived similarity, a probabilistic model chooses the category associated with a higher probability. Therefore, each response is associated with a probability between zero and one.

1.4 Research Framework

In this section we identify research opportunities and propose a research framework in the area of entity matching (Figure 1.1). This dissertation focuses on the three entity matching challenges identified in Section 1.2.

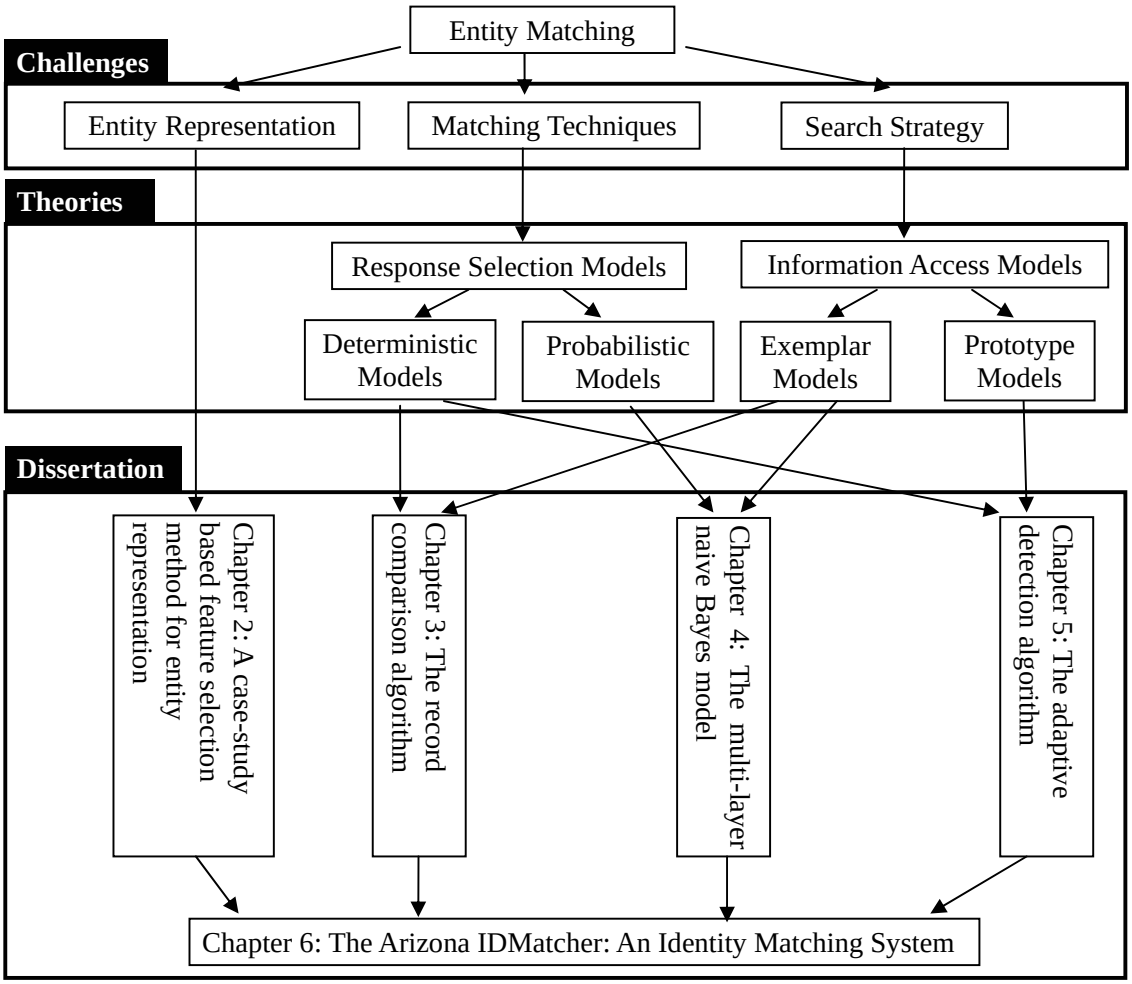


Figure 1.1 Research framework

Entity representation is a feature selection process that identifies a minimal set of common features from different data sources in order to uniquely identify an entity. Current entity matching techniques are inefficient due to using all available common features. The essay in Chapter 2 proposes a case-study-based methodology for identifying key entity features. It first examines the feature patterns of matching entities in a case study methodology. Only features showing distinguishing power are

selected for entity representation. The study is carried out in the domain of law enforcement, focusing on the problem of criminal identity deception.

Entity matching techniques capture the hints of connections between matching entities. The connection is often represented by a similarity score (i.e., a deterministic response selection model) or a matching probability (i.e., a probabilistic response selection model). Due to data quality issues, matching connections need to be captured in an approximate manner rather than an exact manner. The essay in Chapter 3 proposes a deterministic response selection model that computes a weighted-sum similarity score for an entity pair. A weighted-sum score combines the similarity measures on individual features into an overall matching score between two entities. Two entities are considered matches if their similarity score is greater than a matching threshold, which can be learning by a training process. Experiments show that this matching technique is effective in linking deceptive criminal identities. The essay in Chapter 4 proposes a probabilistic response selection model that uses formal probability theory to capture the uncertainty in entity matching. This matching technique is similar to human reasoning process. Thus, the matching results produced by this technique are easy to interpret. It can also use more advanced training methods, such as semi-supervised learning, which minimize the efforts of human experts in the training process.

Searching strategy reduces search space by identifying a small set of prospective entities for matching. It can be very time-consuming to compare a subject entity to every entity in a data source especially when the number of entities is huge. The essay

in Chapter 5 discusses techniques that improve matching efficiency and scalability. An adaptive detection algorithm is proposed and achieves a linear time complexity (O($w'N$)). Experiments show that this algorithm can significantly improve matching efficiency without significant accuracy loss.

The essay in Chapter 6 introduces an identity matching system, the Arizona IDMatcher, that we built based on the above findings. The system achieves the following goals: First, it relies on minimal human effort to develop its matching decision model. Second, the system can be applied to various domains with minimal modification and customization. Third, it is efficient in matching a large number of identity records within or across different information systems.

CHAPTER 2: IDENTITY DECEPTION AND DECEPTION DETECTION IN LAW ENFORCEMENT: A CASE STUDY

2.1 Introduction

In this essay we propose a case-study-based methodology that identifies key identity features that may uniquely identify an identity. The selection of the features is expected to help detect deceptive criminal identities.

Criminal identity deception is a problem critical to law enforcement and intelligence activities. Criminals often intentionally falsify their identities in order to deter police investigation. Identity can be represented by a vector of key and value pairs that identify a specific person. It usually includes information such as name, gender, date-of-birth, social security number, and address. Criminal records in a local law enforcement agency show that many criminals have used deceptive identities. The identity deception problem is also encountered in the field of national security. It received increasing attention following the terrorist attacks in the United States on September 11, 2001. The FBI (2001) reported that most of the nineteen hijackers in the incident used false identities, including impersonating stolen identities and using aliases. This made it difficult for the FBI to dig out their real identities. As a result, the validity of published investigation results is still being questioned.

The 9/11 tragedy might have been avoided if investigative agents were able to recognize false identities. One reason why it is difficult for police officers to realize someone is using a stolen identity is that law enforcement agencies do not easily share information amongst themselves. One solution would be to have law enforcement

agencies work together collaboratively and share information across agencies and even across countries. However, substantial efforts are required to build such a collaborative infrastructure since the existing obstacles are not only technical but also political and social. Another solution would be to address the problem of detecting criminal identity deception. For example, one 9/11 hijacker reported by the FBI used aliases such as: “Majed M.GH Moqed,” “Majed Moqed,” and “Majed Mashaan Moqed.” Another hijacker used two dates-of-birth: “01-01-1976” and “03-03-1976.” Each of these examples has some similarities that could be used to identify variations of deceptive criminal identities. For example, knowing only one date-of-birth used by a suspect, a police officer can look at variations of that date and find out whether this suspect has reported a false date-of-birth in the past.

To the best of our knowledge, criminal identity deception has not yet been addressed in any literature. In this essay we try to address this issue and propose a key identity features that can be used to detect deceptive criminal identities. In section 2.2 we discuss the concept of general deception and identity deception. Criminal identity deception is defined as a subset of identity deception in the law enforcement domain. We report the characteristics of criminal identity deception based on an interview with a police detective. In section 2.3 we describe a case study in which we investigated patterns of how criminals or suspects lie about their identities. We introduce a taxonomy of criminal identity deception built upon the case study and identify key identity features for detecting deceptive criminal identities in section 2.4. In section 2.5 we discuss the need of an automated technique for detecting deceptive criminal

identities. We identify four key identity features for detecting criminal identity deception and propose an machine learning approach in section 2.6. Conclusions and future research are discussed in the last section.

2.2 Defining Criminal Identity Deception

Deception has been studied in social science for many years. However, as a subcategory of deception, identity deception has not been well defined. In this section we try to define this problem based on the understanding of current deception theories. We also address the relationship between identity deception and identity theft/fraud. Criminal identity deception is the type of identity deception occurring in the law enforcement domain. To better understand this issue, we discuss its underlying causes and ways in which criminals lie about their identities.

2.2.1 What Is Deception

As a multidisciplinary concept, deception has been defined in many ways. Knapp and Comadena (Knapp & Comadena, 1979) defined it as “the conscious alteration of information a person believes to be true in order to significantly change another’s perceptions from what the deceiver thought they would be without alteration.” This definition specified acts of lying as information alteration. However, the deceiver could act differently to mislead the receiver, for example, by concealing information. Mitchell (Mitchell, 1966) defined both human and non-human deception in a general way as “a false communication that tends to benefit the communicator.” One flaw in Mitchell’s definition was that it implied including unconscious and mistaken deception (Vrij, 2000). Ekman (1985) defined deception as “one person

intends to mislead another, doing so deliberately, without prior notification of this purpose, and without having been explicitly asked to do so by the target." This definition explicitly stated that deception is an intentional action. However, it is not necessary to exclude the prior notification of intent to deceive. Buller and Burgoon (1998) defined deception more precisely and concisely as "a sender's knowingly transmitting messages intended to foster a false belief or conclusion in the receiver." Rather than focusing on the act itself, they judged deception on the basis of the deceiver's motivations in an interpersonal communication context. This definition is also suitable for describing criminal identity deception, since criminals usually lie about their identities in an interactive environment (for example, during interrogation).

2.2.2 Identity and Identity Deception

Clarke (1994) defined human identity as "the condition of being a specified person." A specified person does not mean a specific person, but an entity described by a distinct set of characteristics. A specific person may adopt different identities at various times or maintain several at once. Identity information is a vector of distinguishing key-value pairs that include names, codes, tokens (e.g., birth certificate), knowledge (e.g., what is the person supposed to know), and biometric information (e.g., appearance, voice characteristics) (Clarke, 1994).

By adopting Buller and Burgoon's definition of deception, identity deception can be defined as a sender's knowingly transmitting identity information intended to foster a false belief in the receiver. This phenomenon exists in different domains. For

example, it is common to observe identity deception in a virtual community (Donath, 1998), specifically the Internet.

Identity deception includes the issue of identity theft or identity fraud. As defined in the Identity Theft and Assumption Deterrence Act, identity theft refers to the action of “knowingly transferring or using, without lawful authority, a means of identification of another person with the intent to commit, or to aid or abet, any unlawful activity” ("Identity Theft and Assumption Deterrence Act," 1998). This definition focuses on the use of another person’s identity and is often called impersonation. Identity deception is a broader concept than identity theft because impersonation is just one of many ways to alter an identity. For example in the hijacker case discussed previously, those hijackers deceived by using variations of their identities rather than impersonation.

2.2.3 Criminal Identity Deception in Law Enforcement

In this section we examine identity deception in law enforcement. Following the discussion on identity deception, we define criminal identity deception as a criminal intentionally altering his/her identity in order to foster a false belief in officers. This definition excludes the use of new identities approved by the Witness Protection Program. This program involves the legitimate acquisition of new identities through the issuance of new legitimate documentation such as birth certificates, driver’s licenses, marriage licenses, homes, and so on. Subjects in this program are trying to protect themselves from being hurt rather than to deter police investigations. Based on

our definition of criminal identity deception, this is considered entirely out of the scope of this research.

Because our topic is domain specific and requires domain knowledge, we interviewed an experienced police detective who has served in law enforcement for more than 30 years. According to him, criminals or suspects usually lie about the particulars of their identity, such as name, date of birth (DOB), address, or identification numbers, in order to deceive a police officer. Why do criminals lie? Hample (1980) concluded from an experiment that most lies were "defensive reactions to minimize trouble in situations in which lying was virtually automatic." This helps to explain the underlying causes of criminal identity deception. According to our detective expert, both suspects who committed the crime and those suspects who were not actually involved in the investigated crime will give false identities. This can be explained by Hample's statement: suspects give defensive reactions to minimize possible future troubles. However, detailed reasons need to be further studied from a psychological aspect.

In current law-enforcement computer systems, police officers run exact-match queries to locate any historical data about a suspect. When a criminal uses a false identity, even if a very similar identity is recorded in the database, an exact-match query will not bring up that record. This results in a discrepancy between available information and the need for information retrieval. According to our detective expert, criminals have found it easy and effective to escape justice by using a false identity. A felon at large may be able to escape arrest by using a falsified identity and continue to

endanger society. On the other hand, it is possible that police officers may find themselves fruitlessly engaged in pursuing a most-wanted criminal who is already in a jail somewhere under a falsified identity. Both cases diminish the efficiency of police investigative activities.

Criminals usually conduct identity deception in two manners: ad-hoc and pre-planned deception. According to the detective expert, ad-hoc deception is the most common. Criminals generally attempt ad-hoc deception while they are in direct contact with police (i.e., they are being interviewed in reference to their possible involvement in a criminal activity). In certain types of crimes such as fraud or forgery, criminals may plan identity deception in advance by assuming someone else's identity or creating a false identity. In some other crimes or criminal careers (e.g., gang members), where criminals perceive that there is a strong likelihood that they will be contacted by police, they may pre-plan deception as well. Within the scope of this research, we only address the ad-hoc manner of deception.

We have identified and defined the problem of criminal identity deception. There is still the question of how criminals lie about their identities when they commit identity deception, specifically in the ad-hoc manner. In the next section, we present findings of a case study that we conducted in a local law enforcement agency. Patterns of criminal identity deception were identified and built into a taxonomy.

2.3 A Case Study on Criminal Identity Deception

To answer the question "in what ways do criminals lie about their identities?" we conducted a case study with criminal records from the Tucson Police Department

(TPD) in order to acquire actual patterns on our subjects. TPD has about 2.4 million criminal records kept on file. All of those records are managed by a computerized database system, which made it very convenient for us to draw samples. Records contained criminal identity information such as name, data of birth (DOB), address, identification numbers (e.g., social security numbers), race, weight, height, hair color, eye color, etc.

After examining each attribute in the records, we discarded those physical description fields that had little consequence in finding deceptive patterns. The discarded physical description fields include height, weight, hair color, and eye color. Some of these fields, such as height and weight, may change over time. According to the detective expert, those fields are considered inexact. In other fields criminals can intentionally make changes with ease. For example, hair can be dyed different colors and colored contact lenses change eye color. Therefore, these physical descriptions are too unreliable to be of any real importance. After dropping the physical description fields, we examined the remaining fields: last name, first name, address, DOB, and Social Security Number (SSN).

We then invited the police detective expert to validate the deceptive criminal identities. First, we manually located criminals having reported deceptive identities. The detective expert was asked to validate criminals' deceptive identities using his investigative methods, for example, comparing mug shots or fingerprints if available, background checking, etc. He could only confirm those deceptive identities that had strong evidence to prove that the person represented by a deceptive identity and the

corresponding target criminal, were the same. Strong evidence included, but was not limited to, a combination of the following: multiple matches in their previous addresses, matches in mug shots or fingerprints, matches in other types of identification number (e.g., driver's license number). We finally identified a sample of 24 criminals containing 372 criminal identities reported from incidents in which they were involved. To make clear comparisons we grouped identities by criminal. Each group had a criminal's true identity information and his/her identity variations, which were also grouped by fields like alias, DOB, SSN, and address. Gender, age, or ethnic group may or may not play a role in how people deceive. Therefore we chose an equal number of male and female criminals, ranging from 18 to 70 in age. Also, there was an equal number of Caucasians, Hispanics, and African Americans in the selected sample (Table 2.1). These were the three largest ethnic groups in the City of Tucson in 2001.

Table 2.1 Breakdown of sample records into the categories of age and ethnic groups

Ethnic group	**The number of criminals in different age groups**				**Total**
	19 and younger	20~29	30~39	40 and older	
Caucasian	2	2	2	2	8
Hispanics	2	2	2	2	8
African American	2	2	2	2	8
				Sum:	24

2.4 A Taxonomy of Criminal Identity Deception

Patterns of criminal identity deception were noticed when we compared an individual's deceptive records to his/her real identity record. We categorized criminal

identity deception into four types: name deception, residency deception, DOB deception, and ID deception. A taxonomy of criminal identity deception was built based upon our observations and analyses.

2.4.1 Name Deception

As noted by our police detective expert, it is common for criminals to lie about their names. In our case study, all twenty-four criminals identified had used a false name. The name field contains three parts: last name, first name, and middle initial. Our sample showed that criminals were more likely to make variations on name spelling and/or name sequence than to use a completely different name. Each type of name deception is described below.

2.4.1.1 Spelling Variation

Deception by spelling variation was the most frequent type in the category of name deception. It can be further categorized based on the way criminals make spelling changes. A criminal could use several types of spelling variations simultaneously.

(1) Name with Similar Pronunciation:

This type of spelling variation means that either the first name or the last name is replaced by a name phonetically the same or similar but different in spelling. To illustrate, one subject named “Cecirio” used the false name “Cicero” instead. In our sample, this type of name deception was found in ten out of twenty-four criminals (41.7%).

(2) Abbreviations and Add-ons:

In this type of spelling deception, criminals may use abbreviated names or add additional letters to their real ones. A good example of an abbreviation is using “Ed” instead of “Edward.” Similarly, using “Edwardo” instead of “Edward” is a good example of Add-on. Seven criminals in our sample data (29.2%) were found to have used this type of deception.

(3) Changing Middle Initial:

Unlike first and last names, spelling is not an issue for middle initials. Criminals sometimes just simply change them. In our sample data, ten criminals (62.5%) either left out or changed their middle initials, or fabricated a middle initial when there was none. The middle initial is not as important as a first name or a last name because when a police detective is conducting a name search on a specific suspect in the database the middle initial is always ignored. According to the detective expert, police officers only use the middle initial to differentiate between suspects when there are several records for a common name.

2.4.1.2 Completeness and Sequence

Our sample showed two types of deception under this category: name swap and partly missing name.

(1) Name Swap:

Name swap is defined as the action of transposing one’s first and last names. For example, “Edward Alexander” can be altered as “Alexander Edward.” This type of deception can only happen where transposing the first and last names does

not raise immediate doubt. People consider some names to be used as first names or last names only. For instance, “Smith” is usually a last name and could be suspected immediately if a criminal reports it as his first name. In our sample of criminal records, two criminals (8.3%) were found to have used name swap.

(2) Partly Missing Name:

Partly missing name is defined as the situation in which a criminal record lacks either the first name or the last name. Seven criminals in our sample data (29.2%) had used this type of deception. Criminals might not report part of their names intentionally in order to interfere with the investigation against them. On the other hand, it is also possible that police officers lose part of the information during the data entry or they are unable to acquire complete information during the investigation.

2.4.1.3 Completely Deceptive Name

A completely deceptive name means that a criminal uses a name, either a first name or a last name or a full name, which is totally different from and irrelevant to its real representation. In that case we would not be able to see any of the patterns described previously. Seven criminals (29.2%) were deceptive in this way. For example, a subject named “Joy Baker” falsified her first name as “Rebecca Baker.” According to our police detective expert, criminals in this case usually choose the name of a brother, sister, or partner.

2.4.2 Residency Deception

Residency deception is related to address information. Generally, an address is composed of a street number, a street direction, a street name, and a street type, e.g., “1201 W Highland Ave.” Suspects usually make changes to only one portion of the full address. Eight criminals in our study were found to have used a deceptive address. Based on our observation, street number was the most commonly altered part.

2.4.2.1 Deception on Street Number

Among the eight criminals we examined, seven (87.5%) deceived on the street number. Deception was made by changing, removing, or inserting some digits into the real street number. For example, the address “1201 W. Highland Ave.” can be altered as “1211 W. Highland Ave.,” “120 W. Highland Ave.,” or “11201 W. Highland Ave.” In most cases, there were no more than two digits altered.

2.4.2.2 Deception on Street Direction

A street direction can be longitudinal (e.g., North, South) or latitudinal (e.g., West, East). Most streets have only one type of direction. Some streets may have both types when swerves exist. For streets having only one type of direction, criminals who alter the direction are very likely to change other portions of the address such as street type and street name as well. That will make the deception more reasonable. However, one criminal in our study only altered the direction without making other changes, which simply created a non-existing address. Another criminal altered both street direction and street type, which made it a valid address and hard to detect. For streets having both types of direction, criminals may simply alter the direction to make the

false address valid. One subject in our case study changed the street direction from “East” to “South” with other portions of his address intact, which exemplified that kind of deception.

Three criminals (37.5%) were found to have altered street directions. Two criminals created a valid false address while the other created an invalid one.

2.4.2.3 Deception on Street Name

A street name can be numeric (e.g., 2nd St.) or textual (e.g., Hatfield St.). In our case study, three subjects (37.5%) falsified their street names. One criminal altered his numeric street name to another numeric name. He reported “73 E. 34th St” as “73 E. 35th St.” The other two criminals used deception on their textual street names by making spelling variations. One reported “Calle Arroyito” as the street name instead of “Calle Del Arroyito,” while the other reported “Desert” instead of “Desert Mesa.” This type of deception is similar to name deception and most types of variation discussed in name deception are expected to occur in the deception on street names as well.

2.4.2.4 Deception on Street Type

Values for street type can be entities such as Street, Road, Avenue, Drive, Boulevard, and Way. Two criminals (25%) were found to deceive on street types. In both cases criminals also altered other parts of the address. It seems that criminals alter street types in order to make the false address look valid. For example, the real address for a criminal was “144 E. 9th St.” In one incident report, he altered his address to “144 S. 9th Ave.” If he had not altered the street type, the address would

have been an invalid one because there is no address such as “S. 9th St.” in the Tucson area. This is still an ad-hoc manner of deception.

2.4.3 DOB Deception

DOB deception is the most common type of criminal identity deception. Our case study showed that sixteen criminals (66.7%) in the sample had deceived on DOB. The DOB field in the TPD database has an eight-digit number representing the year, the month, and the day respectively. For example, “19700215” represents a DOB of February 15th, 1970. By studying the deceptive cases, we found that suspects usually made only slight changes to their deceptive DOB. For example, “19700215” might have been falsified as “19700205” by changing the day. Changes can also be made to month and year. In all DOB deception cases in our sample, 65% only falsified one portion of their DOB, 25% made changes on two portions of their DOB, and 10% made changes to all three portions.

The way criminals altered their DOB was similar to name deception. Criminals made “spelling variations” to their real DOB, such as replacing a couple of digits with false digits and transposing digits. Thus, if two records in the database show the same or similar names and two different but very similar DOB’s (for example, 19560608 and 19560806), a police officer can deduce that both records are in fact the same person.

2.4.4 ID Deception

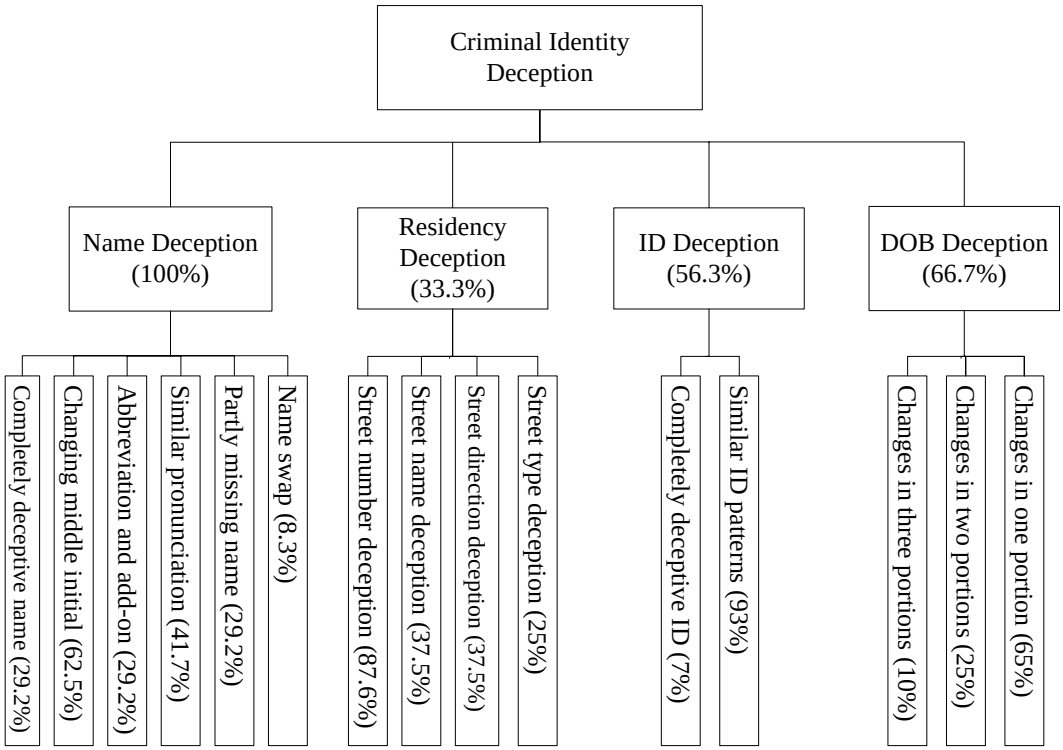
ID is a unique sequence of numbers and letters that is associated with an identification document In the US an identification document can be a passport, a

driver's license, birth certificate, or a social security card. The law enforcement database system we studied stores Social Security Numbers (SSN) along with other types of ID (e.g., driver's license number). There were many more records having SSNs than other types of ID. Therefore we chose the SSN to represent all ID patterns. We assume the deception patterns occurring in SSNs are the same as those occurring in any other types of ID.

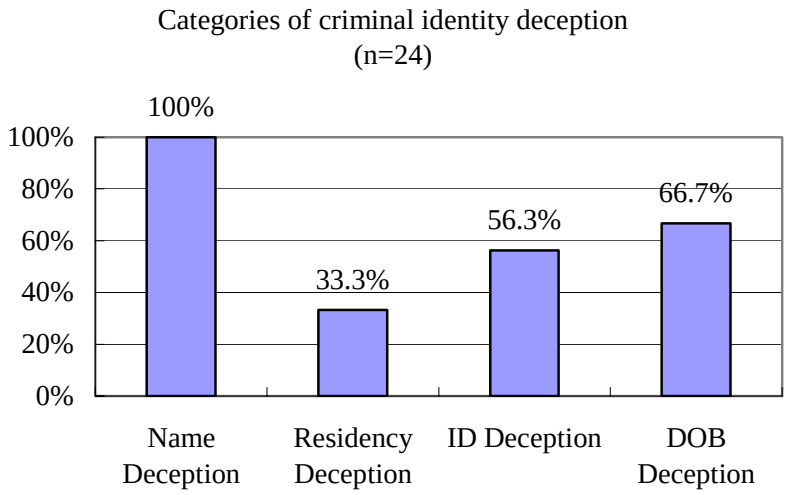
SSN is a nine-digit number, which is a unique identification number for each person. One may not have a SSN under some circumstances (e.g., a non-citizen staying for nonwork purposes). In this study, we only used those records having SSNs. Within our sample, 56.3% of the suspects used a falsified SSN. Among those cases of SSN deception, most of them (96%) varied no more than two digits from the corresponding correct ones. One example of SSN deception is the ID "123-45-6789" may be changed to "123-46-6789" or "123-35-9789." Still, criminals may make variations similar to those described for name deception, such as number swap and completely deceptive numbers. In our sample, we found one case where a criminal gave a totally different SSN.

Figure 2.1 summarizes the different types of criminal identity deception described above. During the case study we noticed it was sometimes difficult to distinguish between deceptive records and records having data entry errors. In this case, we examined all fields in the suspected identity. When there were more altered fields the record was more likely to be deceptive. For example, partly missing names may result from intentional criminal deception or from operational information loss.

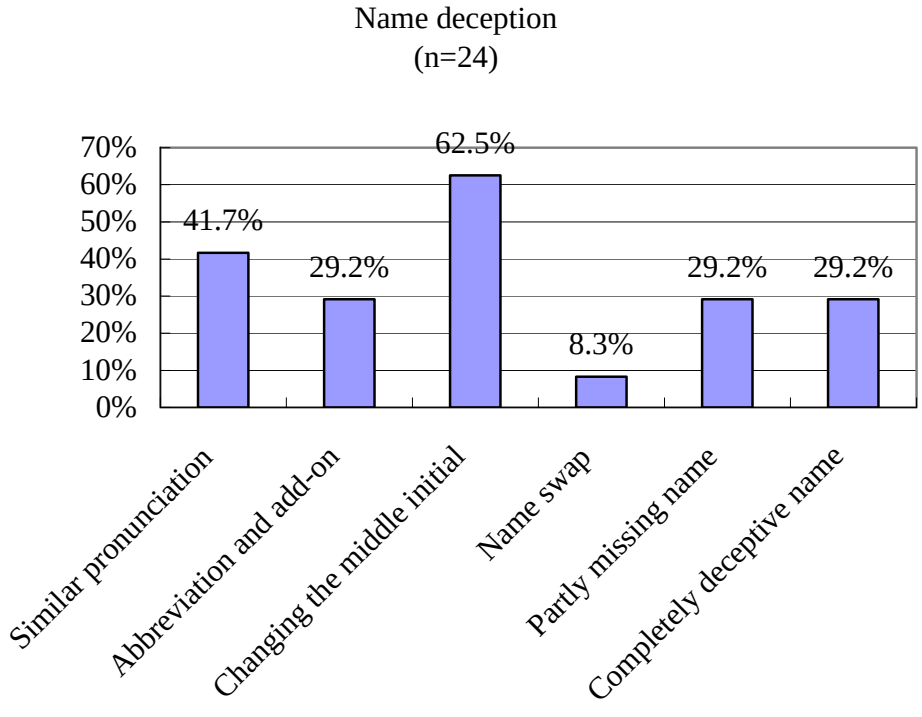
We identified this type of name deception by comparing records in the database and checking for the completeness of names. If two different records showed similar names with one being more complete than the other (e.g., the first name missing in one record), then we looked at other fields in those two records (e.g., DOB, SSN, physical characteristics) and if other fields were very similar we deduced that those two people were in fact the same person and we considered this to be a case of deception. In the same manner if two names were similar with the exception of the first and last names being swapped, we compared other fields in those two records in order to see whether we could deduce that this was a deceptive case.



(a) A taxonomy of criminal identity deception

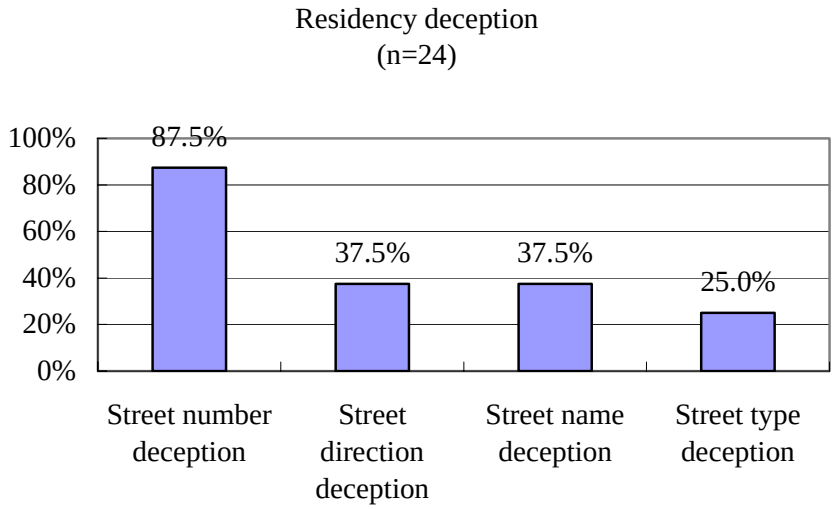


(b) Bar-chart of the major categories of criminal identity deception



*: *Each criminal may have more than one type of name deception.*

(c) Bar-chart of the types of name deception



*: *Each criminal may have more than one type of residency deception.*
(d) Bar-chart of the types of residency deception
Figure 2.1 Taxonomy of criminal identity deception

2.5 The Need of an Automated Method for Detecting Criminal Identity Deception

Police detectives usually do not specifically aim to detect criminal identity deception. A false criminal identity is often revealed as a byproduct of other investigation activities, unless there are serious doubts about a criminal's identity. There are techniques, such as observing a combination of physical, emotional and mental symptoms of deception, developed for deception detection in law enforcement (Aubry & Caputo, 1980). However, those techniques are typically used to verify statements made by criminals. None of them is specifically designed for revealing lies about identities.

Vrij (2000) summarized three ways for detecting lies in law enforcement. The first method is to observe liars' non-verbal behavior, such as their body movements

(e.g., scratching the head), their emotional expressions, their facial expression (e.g., blinking of the eyes), and vocal characteristics (e.g., pitch of voice). It has been shown that there are automatic links between emotions and non-verbal behaviors (Ekman, 1992). Non-verbal cues to deception are more likely to occur if the lie is difficult to fabricate (Vrij, 2000). The emotional fluctuation caused by the action of lying will influence one's behavior, which could expose deception. The second method is to analyze verbal characteristics of what a subject said. Vrij defined several types of verbal characteristics including negative statements, plausible answers, irrelevant information, over-generalized statement, self-references, direct answers, and response length. Verbal cues can help to discriminate between deceptive and truthful statements in the sense that some verbal criteria are more likely to occur in false rather than in truthful statements. Statement Validity Assessment (SVA) and Reality Monitoring are two popular techniques for detecting verbal cues. The third way is to examine physiological responses such as blood pressure, heart rate, palmar sweating, respiration, and so on. The device that can detect physiological activities is called a polygraph. These three techniques are used by people specifically trained for that purpose (for example, polygraph examiners) (Vrij, 2000). The practical accuracy of those three methods, as used by experts in detecting lies, is reported in Table 2.2 (Vrij, 2000).

Table 2.2 Accuracy report of the three methods for detecting lies

	Non-verbal	**Verbal**	**Physiological**
How good are experts at detecting lies?	51%-82% accuracy in truth detection 30%-66% accuracy in lie detection 31%-64% total accuracy	76% accuracy in truth detection* 68% accuracy in lie detection*	72% accuracy in truth detection** 87% accuracy in lie detection** 96% accuracy in truth detection*** 59% accuracy in lie detection***

*: Results in average with the Criteria-Based Content Analysis (CBCA)
**: Results in average with the Control Question Technique
***: Results in average with the Guilty Knowledge Test

In practice, police officers and detectives generally perform worse than the trained experts (DePaulo & Pfeifei, 1986; Ekman & O'Sullivan, 1991; Kohnken, 1987; Kraut & Poe, 1980), so that many cases of deception are not discovered. Deception can also be revealed by investigation (e.g., checking a criminal's history information), which is time-consuming and involves great amounts of manual information processing. For example, a crime analyst may find a suspect using a false identity when he/she is investigating crime patterns and trends for a particular case by conducting link analysis. This method is often used to construct criminal networks from database records or textual documents. Sometimes police detectives compare the suspect's identity information to the current criminal records in the database to find discrepancies that indicate deception, which is simple but also time-consuming. It is also unrealistic for a human detective to examine all records in the database one by one, because of the huge amount of criminal records in a law enforcement database system (for example, the TPD has about 2.4 million records currently).

2.6 Key Identity Features and the Proposed Detection Approach

The experimental results listed in Table 2.2 show that the three methods introduced by Vrij are not reliable enough to detect lies. Although widely used, polygraph results are not even admissible as evidence in the Supreme Court of the United States. These methods can indicate lies to some extent, but they fail to extract the true information linked to the lies. According to our police expert, those general approaches can hardly apply to the previously defined problem of criminal identity deception. . Identity information contains no verbal cues that can be used in verbal characteristic analysis. Usually, a criminal or a suspect's identity is reported at the time they are interviewed in the field by a police officer. Non-verbal cues usually are too subtle to be noticed by the officer.

By examining the characteristics of criminal identity deception, we found clues that might help provide a solution to this problem. First, as the taxonomy indicates, similar values of name, DOB, ID numbers, and address prompt possible identity matches. In most cases observed in the study, criminals "wisely" changed only a small portion of their original identity information. The alteration showed similar patterns to the corresponding true identity. For example, spelling variations and approximate pronunciation are two commonly used tricks to alter names. Second, different fields in identity information need to work together in order to reveal an identity deception. In most circumstances, finding one deceptive field cannot adequately indicate a deception case. For example, there is one record for "Tom Smith" and another one for "Tommy Smith," and both records are completely different in other fields such as

address, birth date, and identification number. “Tommy Smith” could be a deceptive name for “Tom Smith,” but it is most likely that the two records represent two different people when the other identity fields are considered. On the other hand, if all other fields exactly match, a difference in name is probably caused by data entry error.

Approximate string matching techniques have been studied in computer science for years (Navarro, 2001) and may detect spelling differences between strings or search for a given string in a text allowing spelling variations. Based on what we have discovered in the case study, such a technique may help us to locate falsified information (e.g., name deception) if given the original. For example, edit distance is one of the well-known approximate string matching algorithms (Levenshtein, 1966). It calculates the smallest number of character insertions, deletions, and substitutions required to change one string into another. As we built the taxonomy of criminal identity deception, we observed that most of the altered identity information involved character insertions, deletions, and substitutions (e.g., name add-ons and abbreviations). We expect the edit distance between the original information and the altered one can indicate their association to some extent.

Based on the above analysis, we propose a similarity-based approach in Chapter 3 to automatically find a link between true identity information and corresponding deceptive identity information. By comparing an identity input with each record in police databases using our proposed approach, we will consider it a deception if there is a similar but different record in databases. We have to be careful in defining the “similar but different” relationship between two identity records. “Different” means

that two identities are not exactly the same, while “similar” indicates that two identities actually represent the same person. However, we cannot assume all current records are true simply because they exist in the police databases. It is possible that a criminal has given deceptive identities beginning with his/her first record. In this case, we cannot simply tell whether a new input is deceptive or not by finding its corresponding similar and different records in police databases. There will be three possibilities for this case: the new input is true and the existing record is deceptive, the existing record is true and the new input is deceptive, both identities are deceptive. In practice, according to our detective expert, it is more important to find relevant information than to ascertain the truthfulness of an identity. Therefore we should broaden the meaning of deception detection for criminal identity deception in law enforcement. Rather than determining whether a specific identity is deceptive or not, we aim to find at least one deceptive identity in an identity pair and provide more relevant information to assist police investigation.

2.7 Conclusions

In this essay we have defined criminal identity deception based on an understanding of the various theories of deception. Based on an interview with a police detective, we discussed the aspects of criminal identity deception in a practical context. In a case study conducted in a local law enforcement department (TPD) we found different types of criminal identity deception. A taxonomy was built based on deception patterns revealed through the case study. We explored some generic methods that are currently employed to detect deception in law enforcement. However,

these methods are neither effective nor efficient in detecting criminal identity deceptions. Based upon the deception patterns found in the taxonomy, we identified four key identity features that may help find associations between deceptive identities and their corresponding true identities.

Law enforcement agencies require an effective and efficient method for detecting criminal identity deceptions. Exact queries based on deceptive information and general deception detection techniques are not effective for solving this problem. Police officers would effectively reveal deceptive identities if they could compare a suspect's identity to each criminal history record in police databases. However, the huge number of criminal history records prevents them from doing so. Having studied the various types of criminal identity deception, we have identified four key identity features that often indicates associations between deceptive criminal identities and the corresponding true identities. In Chapter 3 we aim to develop an automated deception detection technique that uses the vector of the four features identified to uniquely identify an identity.

CHAPTER 3: AUTOMATICALLY DETECTING DECEPTIVE CRIMINAL IDENTITIES: A RECORD COMPARISON ALGORITHM

3.1 Introduction

In this essay we develop an automated technique for revealing deceptive identities using the four key identity features identified in the case study discussed in Chapter 2.

3.2 Literature Review

It is a common practice for criminals to lie about the particulars of their identity, such as name, date of birth, address, and social security number, in order to deceive a police investigator. For a criminal using a falsified identity, even if it is one quite similar to the real identity recorded in a law enforcement computer system, an exact-match query can do very little to bring up that record. According to our expert detective from the Tucson Police Department (TPD), criminals find it is easy and effective to escape justice by using a false identity.

A criminal might either give a deceptive identity or falsely use an innocent person’s identity. Currently, there are two ways law enforcement officers can determine false identities. First, police officers can sometimes detect a deceptive identity during interrogation and investigation. Techniques used by police officers include repeated and detailed questioning. Repeated questioning is when a police officer asks a suspect the same question (e.g., their social security number) several times. The suspect might forget his/her false answer and eventually reply differently.

Detailed questioning can be used to detect lies in the case of a suspect impersonating another person's identity. The liar might forget detailed information about the person whose identity is being impersonated. Sometimes clues such as the mother's maiden name are also useful. If information provided by the suspect does not match the existing law enforcement records, the deception would be discovered. However, lies are difficult to detect if the suspect is a good liar. Consequently, there are still many deceptive records existing in law enforcement data. Sometimes a police officer has to inquire about an innocent person whose identity was stolen, until their innocence is proven.

Second, crime analysts can detect some deceptive identities through crime analysis techniques, of which link analysis is one that is often used to construct criminal networks from database records or textual documents. Besides focusing on criminal identity information, link analysis also examines associations among criminals, organizations, and vehicles, etc. However, in real life, crime analysis usually is a time-consuming investigative activity involving great amounts of manual information processing.

3.3 Record Linkage Algorithm

A literature survey was conducted to identify research that could contribute to our understanding of criminal profile analysis. In his review of this field, Winkler (Winkler, 1999) defined record linkage as: a methodology for bringing together corresponding records from two or more files or for finding duplicates within a file. Record linkage originated from statistics and survey research. Pioneering work was

done by Newcombe et al. (1959) in a study designed to associate a birth record in a birth profile system with a marriage record in a marriage profile system if information in both records pointed to the same couple. His work enabled the first computerized approach to record linkage. In recent years, record linkage techniques have incorporated sophisticated theories from computer science, statistics, and operations research (Winkler, 1999). Work on library holdings duplication is also a related field.

Two basic components in record linkage are the string comparator and the weight determination method. The string comparator is used to determine the degree of agreement between corresponding fields, such as names, in two records. The weight determination is a mechanism to combine agreement values of all fields and of results in an overall degree of agreement between two records. The performance of a string comparator is very important because it is the key component in computing agreement values. Although current string comparator methods employed in record linkage have different limitations, they can be improved significantly for various applications.

3.3.1 String Comparator in Record Linkage

3.3.1.1 Phonetic String Comparator.

To compute agreement values between surnames, Newcombe et al. (1959) encoded surnames using the Russell Soundex Code, which represented the phonetic pattern in each surname. According to the rules of Soundex coding, surnames were encoded into a uniform format having a prefix letter followed by a three-digit number. Surnames having the same pronunciation in spite of spelling variations should produce identical Soundex codes. For example, “PEARSE” and “PIERCE” are both coded as

"P620." However, Soundex does not work perfectly. In some cases, names that sound alike may not always have the same Soundex code. For example, "Cathy" (C300) and "Kathy" (K300) are pronounced identically. Also, names that do not sound alike might have the same Soundex code; for example, "Pierce" (P620) and "Price" (P620).

3.3.1.2 Spelling String Comparator

A spelling string comparator compares spelling variations between two strings instead of phonetic codes. In another pioneering record linkage study, Jaro (1976) presented a string comparator dealing with typographical errors such as character insertions, deletions, and transpositions. This method has a restriction in that common characters in both strings must be within half of the length of the shorter string.

String comparison, whether string matching or approximate string matching, also has attracted the interest of computer scientists. A common measure of similarity between two strings is defined by Levenshtein as "edit distance" (Levenshtein, 1966). The edit distance is the minimum number of single character insertions, deletions, and substitutions required to transform one string into the other. Agrep, developed by Wu and Manber (1992), is one of the fast approximate string matching algorithms in the field and performs edit distance scoring. Agrep outperforms Jaro's method because it can deal with all kinds of string patterns. Since agrep is designed to detect spelling differences between two strings, it does not detect phonetic errors.

Porter and Winkler (1997) showed the effect of Jaro's method and its several enhanced methods on last names, first names, and street names. In order to compare the Soundex coding method, Jaro's method and agrep, we calculated several string

examples (which were used in (Porter & Winkler, 1997)) using Soundex and agrep respectively. Table 3.1 summarizes a comparison of the results from Soundex, Jaro's method, and agrep. Each number shown in the table represents a similarity measure (a scale between 0 and 1) between the corresponding strings. Soundex measures were good for phonetically similar strings, such as "MASSEY" and "MASSIE," "JON" and "JAN." However, they gave improper ratings when two strings happened to be encoded similarly by Soundex, such as "JONES" (J520) and "JOHNSONS" (J525), "HARDIN" (H635) and "MARTINEZ" (M635). Agrep measures were capable of reflecting the spelling differences in the cases where Soundex measures were improper. Jaro's method could also detect spelling variations between strings. However, it was unable to compare certain string patterns (with scores of zero). In order to capture both phonetic and spelling similarity of strings, a combination of agrep and Soundex was selected for our research.

Table 3.1 Comparison between Soundex, Jaro's method, and agrep

A pair of strings		**Soundex**	**Jaro's**	**Agrep**
JONES	JOHNSONS	0.75	0.79	0.50
MASSEY	MASSIE	1.00	0.889	0.66
SEAN	SUSAN	0.50	0.783	0.60
HARDIN	MARTINEZ	0.75	0.00	0.50
JON	JAN	1.00	0.00	0.66

3.4 Deception Detection Algorithm Design and Experimental Results

3.4.1 Record Comparison Algorithm

To detect the deceptions identified in the taxonomy, we chose the four most significant fields (name, DOB, SSN, and address) for our analysis. The idea was to compare each corresponding field of every pair of records. Agreement/disagreement

values for each field were summed up to represent an overall agreement/disagreement value between two records.

As previously discussed, we used a combination of agrep and Soundex string comparators. Because agrep was originally designed to search for an approximate pattern in a given context, it was revised to behave as a comparator that calculates the number of errors between two strings. The number of errors was normalized to a disagreement value between 0 and 1. To detect both spelling and phonetic variations between two name strings, agrep and Soundex agreement values were computed separately. In order to capture name exchange deception, agreement values were also computed, based on different sequences of first name and last name. We took the agreement value from the sequence that had the least difference (the maximum agreement value) between two names. Agrep itself was used to compare nonphonetic fields of DOB, SSN, and address. The agreement value over all four fields was calculated by a normalized Euclidean distance function. According to our expert police detective, each field may have equal importance for identifying a suspect. Therefore, we started by assigning equal weights to each field.

3.4.2 Experiment Data Collection

In order to test the feasibility of our algorithm, a sample set of data records with identified deceptions was chosen from the police database. At the time, we were not considering records with missing fields. Therefore, we drew from police profiles another set of 120 deceptive criminal identity records with complete information in the four fields. The same veteran police detective verified that all the records had

deception information. The 120 records involved 44 criminals, each of whom had an average of 3 records in the sample set. Some data was used to train and test our algorithm so that records pointing to the same suspect could be associated with each other.

Training and testing were validated by a standard hold-out sampling method. Eighty of the 120 records (2/3) in the test bed were used for training the algorithm, while the remaining one third (40) were used for testing purposes.

3.4.3 Training Results

A disagreement matrix was built based upon the disagreement value between each pair of records. Using the disagreement values in the matrix, threshold values were tested to distinguish between the in-agreement pairs of records and the disagreement pairs. Accuracy rates for correctly recognizing agreeing pairs of records using different threshold values are shown in Table 3.2. When the threshold value was set to 0.48, our algorithm achieved its highest accuracy of 97.4%, with relatively small false negative and false positive rates, both of which were 2.6% (Figure 3.1).

3.4.4 Testing Results

Similarly, a disagreement matrix was built for the 40 testing records by comparing every pair of records. By applying the optimal threshold value 0.48 to the testing disagreement matrix, records having a disagreement value of less than 0.48 were considered to be pointing to the same suspect and were associated together. The accuracy of linkage in the testing data set is shown in Table 3.3. The result shows that

the algorithm is effective (with an accuracy level of 94%) in linking deceptive records pointing to the same suspect.

3.5 Conclusions and Future Work

We present a record linkage method based on string comparators to associate different deceptive criminal identity records. It is a deterministic response selection model that produces matching decisions associated with the probability of one. The experimental results have shown the method to be promising. The testing results also show that no false positive errors (recognizing related records as unrelated suspects) occurred, which means the algorithm has captured all deceptive cases.

Table 3.2 Accuracy comparison based on different threshold values

Threshold	**Accuracy**	**False Negative***	**False Positive****
0.4	76.60%	23.40%	0.00%
0.45	92.20%	7.80%	0.00%
0.46	93.50%	6.50%	2.60%
0.47	96.10%	3.90%	2.60%
0.48	**97.40%**	**2.60%**	**2.60%**
0.49	97.40%	2.60%	6.50%
0.5	97.40%	2.60%	11.70%

* False negative: consider dissimilar records as similar ones
** False positive: consider similar records as dissimilar ones

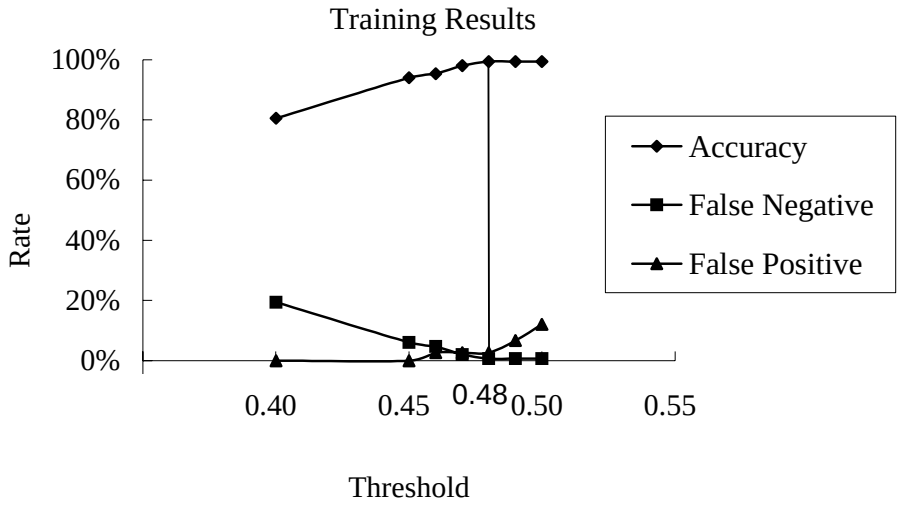


Figure 3.1 Training accuracy comparison based on different threshold values

Table 3.3 The accuracy of linkage in the testing data set

Threshold	**Accuracy**	**False Negative**	**False Positive**
0.48	94.0%	6.0%	0.0%

On the other hand, all the errors occurred in the false negative category, in which unrelated suspects were recognized as being related. In that case, different people could mistakenly be considered the same suspect. This might be caused by the overall threshold value gained from the training process. The threshold value was set to capture as many true similar records as possible, nonetheless, a few marginal dissimilar pairs of records were counted as being similar. Currently, an investigator-guided verification process is needed to alleviate such a problem. An adaptive threshold might be more desirable for making an automated process in future research.

CHAPTER 4: A MULTI-LAYER NAIVE BAYES MODEL FOR APPROXIMATE IDENTITY MATCHING

The record comparison technique proposed in chapter 3 is a deterministic matching technique. Its matching results are associated with certainty (a probability of one). It also uses a supervised training method that is time-consuming in generating a training dataset. In this essay we propose a probabilistic matching technique with the following advantages. First, a probabilistic matching technique is similar to human reasoning process that associates matching decisions with probability ratings between zero and one. Second, it can use advanced training methods such as unsupervised or semi-supervised training that requires no or the minimal efforts of human experts.

4.1 Introduction

Current identity matching techniques either rely on exact value matching (Marshall et al., 2004) or require human intervention (Brown & Hagen, 2003; Wang, Chen, & Atabakhsh, 2004a). Exact value matching is also called all-or-none matching (Gill, 2001). Even if an existing identity record is very similar to the information of a suspect, if it is not actually the same, an exact-match query is unlikely to bring up that record. Approximate matching techniques, such as the record comparison algorithm introduced in Chapter 3, are available. However, they often need to be trained using a manually generated dataset. This manual process, however, can be very time-consuming and labor-intensive.

In this essay we examine what causes problems in identity matching and propose an advanced identity matching technique that requires less or no human intervention.

In section 4.2 we discuss possible identity problems and review existing identity matching techniques. In section 4.3 we state the objectives of this research. We report a case study on identity problems using real law enforcement data in section 4.4. In section 4.5 we propose a Naïve-Bayes model for approximate identity matching. We describe our experimental design and report the results and discussions in section 4.6. We summarize our findings, discuss research contributions, and provide future directions in the last section.

4.2 Literature Review

In this section, we review issues related to identity problems and matching techniques proposed in previous research.

4.2.1 Identity Problems

An identity is a set of characteristic features that distinguish a person from others (Donath, 1998; Jones, 2001). For identification purposes, Clarke (Clarke, 1994) proposed an identity model that categorized identity features into five categories (Table 4.1). An identity can be represented by a vector of features selected from the five categories.

Table 4.1 Identity features included in Clarke's identity model

Feature Category	Description	Example
Names	What the person is called by other people	First name, surname
Codes	What the person is called by an organization	Social Security Number
Knowledge	Information that the person is expected to know	Address
Tokens	What the person has	Driver's License
Biometric features	Physical and difficult-to-change characteristics	Fingerprint

Identity information, however, is unreliable due to various reasons. First, 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). Second, identity information sometimes is subject to intentional deception, especially the identities of criminals or terrorists who are known to use false identities to mislead police investigations (Wang et al., 2004a). Identity deception also exists in online auction. A customer may use false identities to register multiple user accounts in order to drive up the bidding prices (Snyder, 2000). Third, the lack of a reliable unique identifier causes problems when integrating identity information from different sources. For example, one information system may use a social security number (SSN) to uniquely identify a person whereas another system uses employee ID to uniquely represent each identity. There are no easy means to match identities in two systems where unique identifiers do not agree.

These problems may result in multiple identity representations for an individual in an information system or across different systems. To efficiently manage identities, we need a mechanism to associate identities that belong to an individual. It will also be useful in searching for information about a particular person, which is a critical task for law enforcement and intelligence investigations.

4.2.2 Identity Matching

According to a psychological theory, the perceptive matching process is closely related to pair-wise similarity judgment (Ashby & Perrin, 1988). Assuming each

stimulus (e.g., an identity) is represented by a vector of perceived attributes in a multidimensional space, the psychological similarity of two stimuli can be represented by the perceived similarities on individual attributes. Two types of matching models, deterministic and probabilistic models, are often considered to convert the perceived similarities into an overall matching decision (Ashby & Alfonso-Reese, 1995). Deterministic models always produce the same decision (matching or non-matching) on different trials if the perceived similarities are the same. Probabilistic models can capture the uncertainty of matching by associating each matching decision with a probability rating.

4.2.3 Existing Identity Matching Techniques

To the best of our knowledge, there are few solutions designed specifically for the problem of identity matching. Marshall et al. (Marshall et al., 2004) provided an exact matching technique for law enforcement applications. Two identities are considered matching only when their first names, last names, and date-of-birth (DOB) values are identical. However, as identity information is unreliable, it is possible that identities referring to the same person have disagreeing values. Wang et al. (Wang et al., 2004a) proposed a record comparison algorithm to detect deceptive identities. Given two identities, the algorithm first computes a similarity rating for the value-pair of each individual identity feature. Assuming features are equally important in making a matching decision, all the similarity ratings are combined into an overall similarity rating using a Euclidean distance function. The two identities being compared are considered matching when the overall similarity rating is greater than a threshold

value. A supervised learning process is required to determine the threshold value. This technique has the following disadvantages. First, supervised learning needs experts to manually generate a training dataset, which can be time-consuming and inefficient. Second, the assumption of features' equal importance is ad-hoc. Moreover, this technique is specifically designed to match deceptive identities. It is unknown whether this algorithm could match identities having other problems such as unintentional errors.

4.2.4 Entity Matching Techniques

The problem of identity matching can be considered as a special case of entity matching, where it is determined whether an entity in one database is the same as the entity in another one (Dey et al., 2002). Mostly studied in the area of databases, entity matching techniques consist of two components: similarity measures and a matching decision model. Like an identity, an entity is also represented by a feature vector. A similarity measure is defined for each entity feature and calculates a score for the perceived similarities between two feature values. Table 4.2 shows common similarity measures defined for different feature types.

A matching decision model maps a set of feature similarity scores onto a binary decision variable. We classify existing entity matching models into two categories: deterministic and probabilistic models.

Table 4.2 Common similarity measures for different feature types

Feature Type	**Similarity Measure**			
Numeric	Similarity(s1, s2) = $1 - \frac{\lvert s_1 - s_2 \rvert}{\max(s) - \min(s)}$ (Johnson & Wichern, 1989)			
Binary	Similarity(s1, s2) = $\begin{cases} 1, \text{when } s_1 \text{ and } s_2 \text{ agree} \\ 0, \text{otherwise} \end{cases}$ (Anderberg, 1973)			
Nominal	Look up a similarity table that is elicited from users (Anderberg, 1973)			
Textual strings	Phonetic	Character-based	Token-based	Hybrid
	Soundex (Russell, 1918)	Edit distance (Levenshtein, 1966)	Jaccard similarity (Bilenko, Mooney, Cohen, Ravikumar, & Fienberg, 2003)	Token-based plus character-based (Bilenko et al., 2003)

4.2.4.1 Deterministic Decision Models

A deterministic decision model associates matching decisions with the probability of one. Given a pair of entities, it often combines similarity scores of individual features into an overall similarity rating and makes matching decisions based on the overall similarity rating. Weights may be used to represent relative importance of individual features when combining individual similarity scores. Brown and Hagen (Brown & Hagen, 2003) proposed a deterministic technique for associating suspects or incidents having similar modus operandi. It first compares corresponding feature values of two records and calculates a weighted total similarity measure (TSM). It relies on experts to estimate weights by minimizing the difference between the algorithm's similarity ratings and those given by experts. For each suspect or incident, this technique orders candidate matching records by their TSM values.

Assuming a one-to-one mapping between two databases, Dey et al. modeled entity matching as an integer programming problem (Dey et al., 2002). In this model, individual feature similarities are first combined into a total similarity measure as a weighted-sum. Weights are elicited from users. Entities in two databases are mapped in a way that the total cost of type-I and type-II errors is minimized. The major problem of both approaches is their reliance on human input to determine feature weights. They become inefficient since human interventions are time-consuming. Some deterministic models developed unsupervised clustering algorithms to find approximate matches without human interventions (Li & Biswas, 2002; Li, Chang, Garcia-Molina, & Wiederhold, 2002). However, the clustering-based algorithms tend to achieve low recall ratings or high recall with very low precision.

4.2.4.2 Probabilistic Decision Models

A probabilistic decision model applies statistical theories to capturing the uncertainty in entity matching. Unlike a deterministic model, a probabilistic model associates a matching decision with a probability between zero and one in most of the cases. A matching decision is made directly based on individual similarity ratings instead of an overall similarity score. Originated in the area of statistics, record linkage (RL) is a probabilistic model that identifies records corresponding to the same entity in one or more data sources. A formal RL definition was given by Fellegi and Sunter (Fellegi & Sunter, 1969). Given γ, a comparison vector that consists of feature similarity ratings for a record pair, RL calculates an odds ratio $R = m(\gamma) / u(\gamma)$, where $m(\gamma)$ is the probability that γ belongs to the matching set (M) and $u(\gamma)$ is the

probability that γ belongs to the non-matching set (U). The higher the ratio *R*, the more likely it is for the two records being compared to match. Two cut-off threshold values are determined according to the expected type-I and type-II error rates. When *R* is greater than the upper threshold, a matching decision will be made. When *R* is smaller than the lower threshold, a non-matching decision will be made. When *R* falls between the two thresholds, the comparison needs to hold for clerical review. However, a supervised learning is required to estimate the parameters such as $m(\gamma)$, $u(\gamma)$ and threshold values. It needs a training dataset generated by domain experts. This process could be time-consuming and labor-intensive. Winkler (Winkler, 1998) and Jaro (Jaro, 1989) improved the classic RL by providing an unsupervised learning mechanism. Their approaches assume that all comparison vectors are distributed according to a finite mixture with unknown parameters such as $m(\gamma)$ and $u(\gamma)$. An EM algorithm is used to estimate these parameters. The unsupervised learning avoids the time-consuming process of manually generating a training dataset. However, it only performed well in a few situations that were extremely favorable (Winkler, 2002).

Ravikumar and Cohen proposed a three-layer hierarchical graphical model for record linkage problems (Ravikumar & Cohen, 2004). In this model a layer of latent variables were added between the binary matching decision variable and feature similarity ratings. This latent layer captures the intuition that a match decision made for a record pair is often dependent on matching decisions based on features rather than the similarity ratings of features. To estimate the parameters of the graphical

model shown in Figure 4.1, the structural EM algorithm is used for unsupervised learning. Experiments showed that this approach achieved performance comparable to that of fully supervised RL methods. Although effective, this approach has the following problems. First, similar to the classic RL, unsupervised learning is not always preferable because unlabeled data alone often are not sufficient for training (Nigam, McCallum, Thrun, & Mitchell, 2000) and the model is subject to overfitting the noisy data (Ravikumar & Cohen, 2004). Second, the computational complexity of this approach is high because it allows dependencies between latent variables. These dependencies can increase computation in the orders between 100 and 10,000 (Winkler, 2002). Moreover, the three-layer architecture may limit the ability to capture more complex matching heuristics. For example, a matching decision made for a pair of names can depend on the matching decisions of first name and surname. Therefore, an extra layer of latent variable is required to capture the matching decisions of name components.

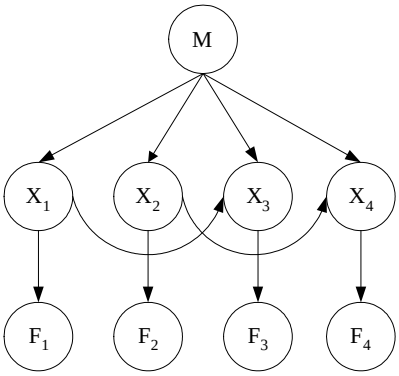


Figure 4.1 A three-layer hierarchical gaphical model

In summary, deterministic entity matching models rely on human input to determine feature weights. This manual process can be inefficient and time-consuming. Probabilistic models use statistical theories to capture feature weights in the form of probabilities. With unsupervised learning, they require no human intervention and can be fully automated. Compared to record linkage, the three-layer hierarchical graphical model is preferable. Not only does it build upon a formal statistical model, but also the learning and inference algorithms of the graphical model have a build-in mechanism for handling missing values, which is a common problem in typical database systems.

4.3 Research Objectives

In this research our first objective is to investigate carefully the identity problems that hinder effectively matching identities. We hope such a study could provide a good basis for proposing a more advanced identity matching technique. Our second objective is to develop a probabilistic decision model for identity matching. We hope the proposed approach could significantly improve the performance of identity matching. A systematic evaluation was conducted to evaluate our proposed approach.

4.4 A Case Study on Identity Problems

The case study presented in Chapter 2 focused specifically on the problem of identity deception. In this section we use the same case study methodology to study all types of identity problems.

Law enforcement agencies have accumulated a great amount of personal identities from investigational activities. Local police departments can provide a rich

data source for studying identity problems. We chose the Tucson Police Department (TPD) as our test bed. TPD serves a relatively large population of 487,000 that ranks 30th among U.S. cities with populations of over 100,000. We hope that the results of the case study conducted at the TPD can be generalized to other agencies.

TPD's COPLINK database maintains 2.4 million identity records. Each identity record consists of many features such as name, DOB, ID numbers, gender, race, weight, height, address, phone number, eye color, and hair color. Name is a mandatory feature and always has a value. Other features are allowed to be empty (i.e., missing) or are assigned a default value when not available (e.g., the default value for height is one inch in the TPD).

4.4.1 Data Collection

We randomly drew 200 unique identity records out of the 2.4 million records stored in the TPD database. We considered them to be "suspects" for whom we were trying to find matching identities in the TPD database. Given the huge amount of identity records in the TPD, it is nearly impossible to manually find matching identity records. We used the identity matching approach that we had previously developed to compute a similarity score when comparing each "suspect" against every identity record in the database (Wang et al., 2004a). For each suspect, we chose the top ten identity records having the highest similarity scores. With the help of a veteran TPD detective who has served law enforcement for 30 years, we manually verified the 10 possible matches for each of the 200 suspects. Each possible match was classified into one of the four categories defined in Table 4.3. The first two categories, deception (D)

and error (E), imply a true match. A possible match was considered an unintentional error when identical values were found in key attributes such as name and ID numbers. A possible match was considered intentional deception when values of key attributes were not identical but showed similar patterns that can be visually identified by the police expert. If a possible match had missing values in many attributes and we were unable to judge, we classified it as U (uncertain). All other cases were classified as N (nonmatch). We found that more than half (55.5%) of the 200 “suspects” had at least one true match, caused by deception (29.5%) and/or by errors (42%) in the database (Table 4.4).

Table 4.3 Classification categories

Category	Description
D	Intentional Deception
E	Unintentional Error
N	Non-matching
U	Uncertain (too little information to make a call)

Table 4.4 Statistics of matched identities

Category	Number of Suspects	Percentage
Matching	111	55.5%
-- Intentional deception (D)	(59)	(29.5%)
-- Unintentional errors (E)	(84)	(42.0%)
Non-matching (N)	56	28.0%
Uncertain (U)	64	32.0%

4.4.2 Taxonomy of Identity Problems

After carefully studying the patterns of the features in matching identities, we created a taxonomy of identity problems (Figure 4.2) based on our findings. Among

others features, matching values in name, DOB, ID numbers, and address indicate deception or errors in most cases. Errors were found to occur in only one feature of the records with errors. There were 50% of identities with errors in name, 11.9% in DOB, and 3.6% in ID numbers. It was interesting that 34.5% identities were duplicate, which is probably caused by a bad database design. Deceptive identities usually involve false values in more than one feature. The name feature was found to be the most deceptive feature (91.5%). Less than half of the deceptive identities (44.1%) had altered DOB values, 22% of them had altered ID numbers, and 6.8% of the deceptive identities had altered residential addresses.

Value changes resulting from errors were minor in most cases. For example, all false DOB values and false ID numbers resulting from errors were found to have only a one-digit difference with their corresponding true values. Deception was found in values for name, DOB, ID numbers and address. Value changes that resulted from deception are more drastic than those that resulted from errors. We found that people preferred telling a half-truth lie rather than completely making up things. In most of the cases, deceptive values looked very similar to their corresponding true values. Although in some cases a deceptive value could be very different from its true value in one feature (for example, using someone else's name), other feature values such as DOB and ID numbers might still remain similar and help to identify the connections between the two identity records.

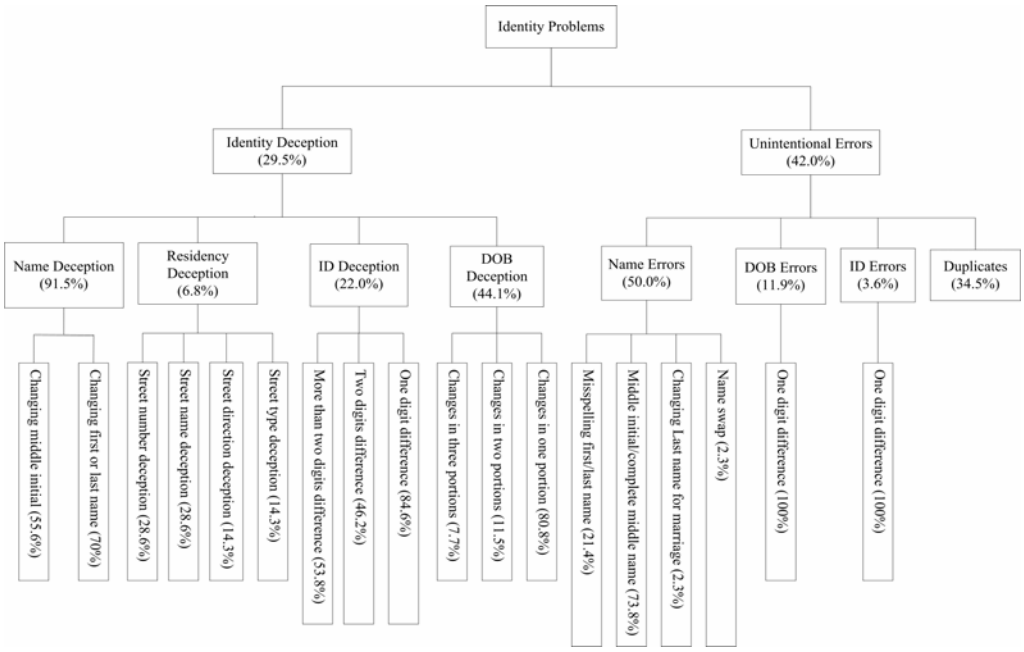


Figure 4.2 Taxonomy of identity problems

4.5 Research Design

4.5.1 A Multi-Layer Naïve Bayes Model

Based on the three-layer hierarchical graphical model, we propose a multi-layer naïve Bayes model (Figure 4.3) by making the following modifications. First, we assume middle-layer latent variables are independent of each other. After removing dependencies between the latent variables, the hierarchical graphical model becomes a tree-augmented naïve Bayes model. In the rest of this dissertation we use the term naïve Bayes to refer to a tree-augmented naïve Bayes model for simplicity reasons. In spite of the strict assumption, naïve Bayes classifiers reportedly often perform better than many complex Bayesian network models (Hand & Yu, 2001; Zhang, 2005). In addition, removing the dependencies may reduce computation complexity in the orders between 100 and 10,000 (Winkler, 2002). Second, we extend the three-layer

network structure into a generic multi-layer structure in order to capture complex matching heuristics. Moreover, we propose an unsupervised learning method that is expected to achieve a balance between model training effectiveness and efficiency.

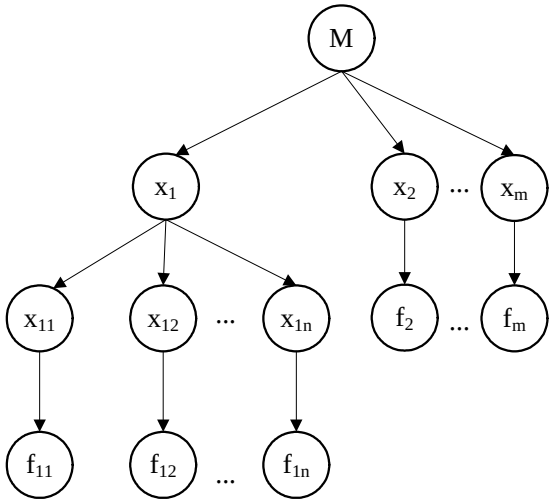


Figure 4.3 A multi-layer naïve Bayes model

Let I be an identity record represented as a vector of m feature values, $I=\{I_1, I_2, \ldots, I_m\}$. Given a pair of identities, a comparison vector F consists of the similarity ratings computed for the value-pairs of m features: $F=\{f_1, f_2, \ldots, f_m\}$. A binary-valued node x_i is defined as the latent matching variable for the i^{th} feature similarity rating f_i $(1 \le i \le m)$. The value of x_i is conditionally dependent on the value of f_i. The value of the match-class variable M is conditionally dependent on those intermediate latent match variables. The removal of the dependencies between latent variables makes this model a Naïve Bayes model, the computational complexity of which is greatly reduced. In addition, our proposed model allows more than three layers in order to determine matching decisions for complex features (e.g., name). In these cases, a

latent match variable x_i may have its own latent matching variables x_{i1} , x_{i2} , …, x_{in} for the n sub components (e.g., first name, surname) of the i^{th} feature. The value of each sub latent match variable depends on the corresponding sub feature similarity rating.

The proposed a multi-layer Naïve Bayes model for identity matching is shown in Figure 4.4 According to our case study on identity problems, matching values in name, DOB, ID numbers, and address indicate matching decisions in most cases. We have chosen those four features to represent each identity in our research.

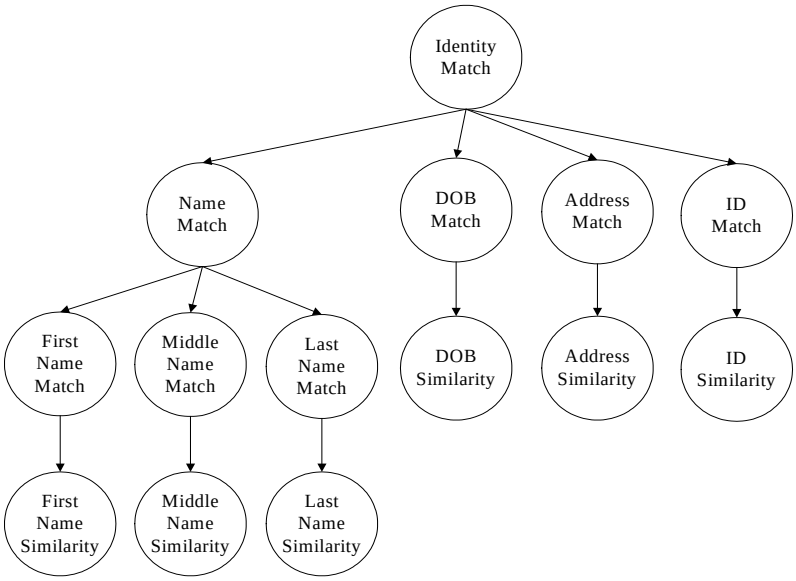


Figure 4.4 A multi-layer naïve Bayes model for identity matching

4.5.2 A Naïve Bayes Framework for Identity Matching

The overall framework we propose consists of a training process and a testing process. Given a training dataset, the training process estimates the parameters of the proposed model. Given a pair of identities with an unknown matching decision, the testing process infers the probability that the identity pair is matched using the trained model. Figure 4.5 illustrates both training and testing processes. In the rest of this section we discuss each component of the framework.

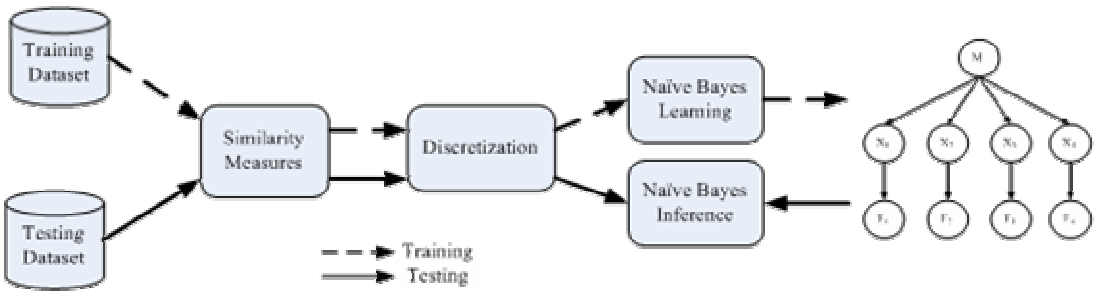


Figure 4.5 A naïve Bayes framework for identity matching

4.5.3 Similarity Measures

All the four features included in our proposed model are textual features. Different similarity measures exist for computing a similarity score between two strings (shown in Table 4.2). Bilenko et al. (Bilenko et al., 2003) experimented with these string comparators and found that Levenshtein edit distance (Levenshtein, 1966) performed the best with name matching using Census data. Because identity records are quite similar to Census records, we choose Levenshtein edit distance to compute a similarity rating between two strings. Edit distance is computed as the minimum number of character insertions, deletions, and substitutions required to transform one string into the other. A normalized edit distance is computed as the following:

$$Sim(S_1, S_2) = 1 - \frac{ED(S_1, S_2)}{MAX(|S_1|, |S_2|)}$$

where ED() is the Levenshtein edit distance function and $|S|$ is a function that returns the length of the string S.

4.5.4 Discretization

Naïve Bayes (NB) is a multinomial probability model. Continuous similarity scores cannot be directly used by NB learning and inference algorithms. There are two common approaches, including Gaussian mixture models and discretization, to deal with continuous variables in multinomial probability models. Gaussian mixture models assume values are Gaussian distributed and estimate a probability density function using the training data. Parameters of the multinomial probability model are generated using the estimated probability density function. However, this technique

may affect the learning accuracy if the density function is not a proper estimate of the true density (Yang & Webb, 2003). Discretization is a process that separates a continuous value domain into a number of intervals (Ravikumar & Cohen, 2004; Winkler, 1990). Feature values falling into the same interval have the same nominal value. The number of intervals determines the approximation of the continuous distribution. If the number is too small, it leads to a poor approximation as well as a poor learning performance. If the number is too large, it significantly increases the computational complexity of NB learning and inference algorithms (Dechter, 1996).

Yang and Webb suggested a Weighted Proportional *k*-Interval Discretization method (WPKID) that performed better than the alternatives at reducing NB learning errors (Yang & Webb, 2002). This method maintains a good balance between bias reduction and variance reduction. It also keeps the interval size above a minimum number so that each interval can have enough instances to draw statistical inferences. Let t be the number of intervals and s the interval size. WPKID uses the follow formula to calculate s and t:

$$s \times t = n$$
$$s - 30 = t$$

where n is the number of instances in the training dataset. Each interval contains at least 30 instances.

4.5.5 Naïve Bayes Learning

We propose a semi-supervised learning method to estimate the parameters of the proposed NB model. We believe semi-supervised learning is more preferable than fully supervised learning or unsupervised learning for the identity matching problem.

Supervised learning requires manually generating a training dataset of identity matches, which could be time-consuming and labor-intensive. Unsupervised learning is subject to overfitting towards noisy data and may reduce learning accuracy. With a network data structure, our no-dependency NB assumption greatly reduces the computational complexity in learning. The Expectation-Maximization (EM) algorithm is commonly used for semi-supervised NB learning (Nigam et al., 2000). It consists of three steps:

(1) **Prime M step:** Estimate the NB parameters θ_i using the Maximum Likelihood Estimation (MLE) with a set of labeled data.

$$\theta_i = p(x_i \mid x_{pai}) = \frac{N_{x_i x_{pai}}}{\sum_{x_i} N_{x_i x_{pai}}}$$

where x_i is the value of node i, x_{pai} is the value of i's parent node, and $N_{x_i, x_{pai}}$ is the number of instances that contain both x_i and x_{pai}.

(2) **E step:** Given the parameters θ_i, estimated in the prime M step, the E step computes the expected values of unknown class labels in the training dataset using an inference algorithm, which is discussed in the next section.

(3) **M step:** This step treats the expected values estimated in the E step as though they were observed and estimates a new set of NB parameters using MLE.

Steps (2) and (3) iterate until the following likelihood function converges:

$$p(D \mid \theta) = \prod_{i=1}^{N} p(x_i \mid x_{pai})$$

where D is the training dataset and N is the number of instances in D.

4.5.6 Naïve Bayes Inference

Given a parameterized NB model and a new comparison vector, an inference algorithm infers the probability that the two identities being compared match. Most Bayesian network inference algorithms can apply. However, most of them have exponential computational complexity. We choose Pearl's evidence propagation algorithm for polytrees (Pearl, 1988) because it can achieve a linear computational complexity. A polytree is a network where there is at most one path from one node to another.

4.6 Experiments

In this section we report a systematic evaluation of our proposed NB model for identity matching using real law enforcement data.

4.6.1 A Law Enforcement Dataset

We randomly drew 200 unique identity records out of the 2.4 million records stored in the Tucson Police Department (TPD) database. We considered them to be "suspects" for whom we were trying to find matching identities in the database. Given the huge amount of identity records in the TPD database, it is nearly impossible to manually find matching identity records. We used the identity matching approach developed by Wang et al. (2004a) to compute a similarity score when comparing each "suspect" against every identity record in the database. For each suspect, we chose the top ten possible matches that had the highest similarity scores. With the help of a police veteran we manually verified the 10 possible matches for each suspect and labeled each possible match with either "true match" or "false match." Each identity

was represented by a vector of four features: name, DOB, address, and SSN. The training dataset was generated by comparing each suspect's primary identity against each of its possible matches. The training dataset contained 2,000 comparisons, each of which was represented by a comparison vector and a matching decision label. A matching decision label was set to one if the corresponding comparison was considered to be a true match by the police expert, and zero otherwise.

4.6.2 Performance Metrics

We consider identity matching as a classification problem. When we compared a matching decision label predicted by a matching algorithm to the actual label, the result fell into one of the four categories defined in Table 4.5.

Table 4.5 Categories of classification results

	Actual Label=1	**Actual Label=0**
Predicted Label =1	True Positive (TP)	False Positive (FP)
Predicted Label =0	False Negative (FN)	True Negative (TN)

We measured the performance of our proposed model using the following three measures: recall, precision, and F-measure. Those measures are widely used in information retrieval and text mining (Salton, 1988). Precision, in this scenario, is defined as the percentage of correctly detected matching identities in all matching identities suggested by the algorithm. Recall is the percentage of matching identities that are correctly identified. F-measure is a well-accepted single measure that combines both recall and precision.

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$F\text{-}measure = \frac{2 * precision * recall}{precision + recall}$$

4.6.3 Experimental Design

4.6.3.1 Hypotheses

In our experiments we compared the performance of our proposed model to that of other existing identity matching techniques, namely the exact-match based technique (Marshall et al., 2004) and the record comparison algorithm (Wang et al., 2004a). We also evaluated the performance differences of our model in three different learning modes: supervised, semi-supervised, or unsupervised learning. The following hypotheses were tested in our experiments.

Hypothesis 1. The multi-layer Naïve Bayes model with semi-supervised learning can achieve an F-measure comparable to that with supervised learning.

Hypothesis 2. The Naïve Bayes model with semi-supervised learning can achieve a higher F-measure than that with unsupervised learning.

Hypothesis 3. The multi-layer Naïve Bayes model can achieve a higher F-measure than the exact-match based technique.

Hypothesis 4. The multi-layer Naïve Bayes model can achieve a higher F-measure than the record comparison algorithm.

4.6.3.2 Testing Procedure

The exact-match based technique: A 10-fold validation procedure was adopted. We randomly divided the training dataset into 10 subsets. For each subset, we

compared each identity to every other identity in the same subset using the exact-based matching heuristic. The predicted class label of a comparison was assigned the value one (i.e., a match) only when the first name, last name, and DOB values of one identity were identical to those of the other identity being compared. Precision, recall, and F-measure were calculated for each subset.

The record comparison algorithm: We also used a 10-fold validation method here. After randomly dividing the dataset into 10 subsets, each time we used 9 subsets for training in which a threshold value was determined. With the other subset for testing, we computed a similarity score for every pair of identities in that subset. Only when the similarity score is greater than the threshold value determined in the training, we consider the two identities as matching.

The multi-layer Naïve Bayes model: During the training process, the EM algorithm estimated the parameters of the NB model using 9 subsets. During the testing, we compared every pair of identity in the other subset and used Pearl's inference algorithm (Pearl, 1988) to calculate the probability that the two identities match. The predicted class label was assigned the value one when the probability was greater than 0.5 (i.e., the probability of matching is greater than that of not-matching), or zero otherwise. To implement semi-supervised learning, we controlled the ratio of unlabeled instances by randomly removing class labels from a percentage of training instances. When the percentage was set to zero (i.e., no class labels were removed), the learning became supervised. We increased the percentage by 10% each time until all class labels of training instances were removed, and it became unsupervised

learning. In unsupervised learning the prime M step of the EM algorithm cannot apply due to the absence of labeled instances. Instead, we started with a prime E step that randomly assigned class labels to all training instances. The M step therefore can follow the prime E step and is followed by the iteration of the EM algorithm.

4.6.4 Experimental Results

In this section we discuss the experimental results summarized in Table 4.6 and report findings related to each of the four research hypotheses.

Table 4.6 Experimental results

			Precision	**Recall**	**F-Measure**
Exact-Match Based Technique			0.9667	0.0291	0.0560
Record Comparison Algorithm			0.6401	0.9073	0.7307
Multi-Layer Naïve Bayes Model	Supervised Learning		0.9563	0.7015	0.8063
	Semi-Supervised Learning	R=0.1	0.9493	0.6918	0.7983
		R=0.2	0.9513	0.6761	0.7868
		R=0.3	0.9342	0.6702	0.7792
		R=0.4	0.9598	0.6662	0.7814
		R=0.5	0.9543	0.6715	0.7867
		R=0.6	0.9734	0.6592	0.7807
		R=0.7	**0.9613**	**0.6394**	**0.7632**
		R=0.8	0.9673	0.6002	0.7326
		R=0.9	0.9839	0.5560	0.7017
	Unsupervised Learning		0.1635	0.5567	0.2396
R is the ratio of unlabeled instances to all instances in a training dataset; Values are the average over 10 subsets					

(1) Hypothesis 1: Semi-Supervised Learning vs. Supervised Learning

With varied percentages of unlabeled training instances, the multi-layer Naïve Bayes model with semi-supervised learning achieved F-measures ranging from 0.7017 to 0.7983 (Figure 4.6). When the percentage was less than or equal to 70%, the F-measure performance of the semi-supervised learning was not statistically different

from that of the supervised learning (p-value>0.05). Therefore, we fixed the percentage of unlabeled training instances at 70% in subsequent hypotheses testing. At this level the semi-supervised learning can perform comparably to the supervised learning, but with the least human inputs.

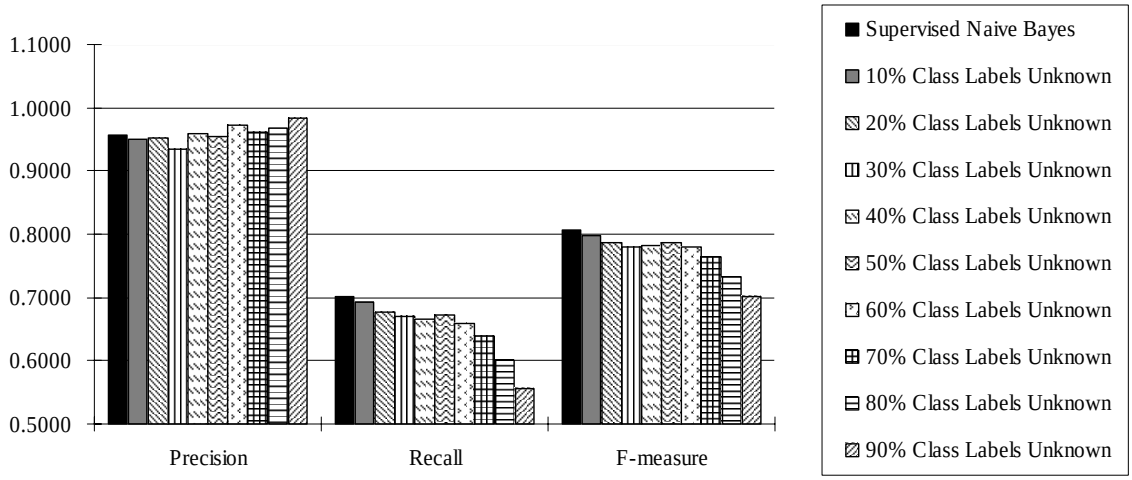


Figure 4.6 Semi-supervised naïve Bayes vs. supervised naïve Bayes

As the percentage of unlabeled training instances increased, one may notice that the precision of the semi-supervised learning improved slightly while its recall slightly degraded. This may be caused by the uneven class distribution. In our dataset the proportion of minority class instances (i.e., matching instances with class labels equal to one) was less than 11%. Learning with unlabeled instances favored the majority class more than the minority class (i.e., considering actual matching instances as non-matching). In general we found that adding unlabeled instances reduced false positive rates while increasing false negative rates.

(2) Hypothesis 2: Semi-Supervised Learning vs. Unsupervised Learning

The Naïve Bayes model with unsupervised learning achieved low F-measures (with an average of 0.2396) for identity matching (Figure 4.7). Statistical *t*-tests showed that semi-supervised learning significantly outperformed unsupervised learning at all levels (*p*-values < 0.001). Hypothesis 2 was supported. Due to the lack of prior knowledge (i.e., a set of labeled instances), unsupervised learning favored the majority class even more. This led to less true positive and more false negative predictions.

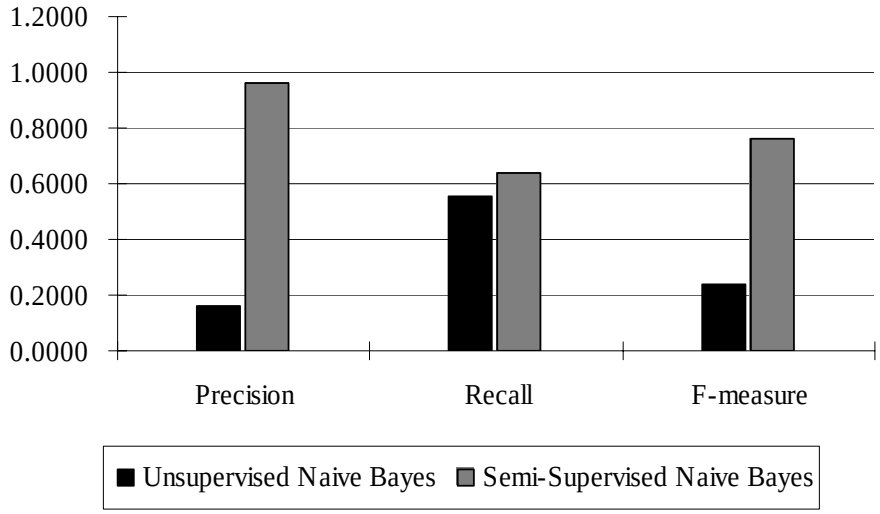


Figure 4.7 Semi-supervised naïve Bayes vs. unsupervised naïve Bayes

(3) Hypothesis 3: Naïve Bayes Model vs. Exact-Match

Although the exact-match based technique achieved high precision (0.9667), it received a very low average recall (0.0291) as well as a low average F-measure (0.0560) (Figure 4.8). The low recall suffered greatly from those matching identities with disagreed name and/or DOB values. As our case study on identity problems indicates, many matching identities have varied feature values. The proposed Naïve

Bayes model with semi-supervised learning performed significantly better than the exact-match technique (p-values<0.001). Hypothesis 3 was supported.

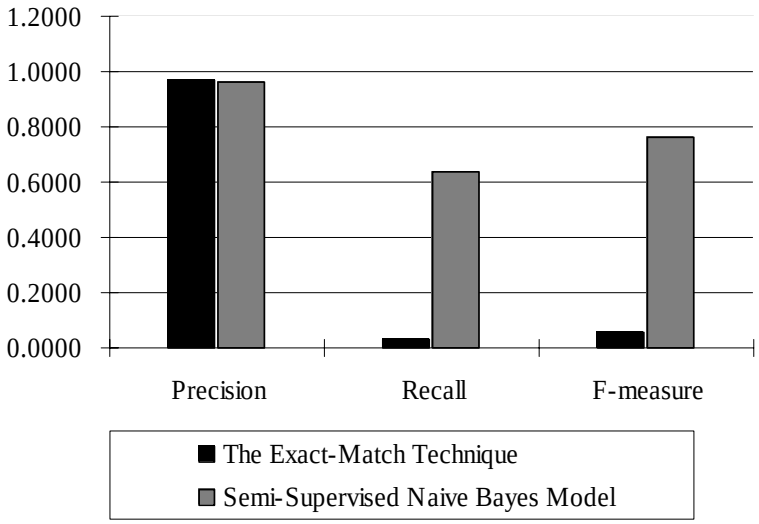


Figure 4.8 Semi-supervised naïve Bayes vs. exact-match technique

(4) Hypothesis 4: Naïve Bayes Model vs. Record Comparison

The record comparison algorithm achieved an average F-measure of 0.7307 (Figure 4.9). Its precision (0.6401) was low due to a large number of false positive matching decisions. Non-matching identities were considered as matching ones when their similarity ratings were greater than the decision threshold. The semi-supervised Naïve Bayes model achieved statistically higher precision and lower recall than the weighted-sum technique. The low recall was also likely caused by the skewed class distribution of the dataset. The overall F-measure of the semi-supervised Naïve Bayes model did not statistically outperform the weighted-sum technique (p-value=0.328). Hypothesis 4 was not supported. However, the semi-supervised Naïve Bayes model is

still favorable because it achieved comparable performance with only 30% training instances labeled.

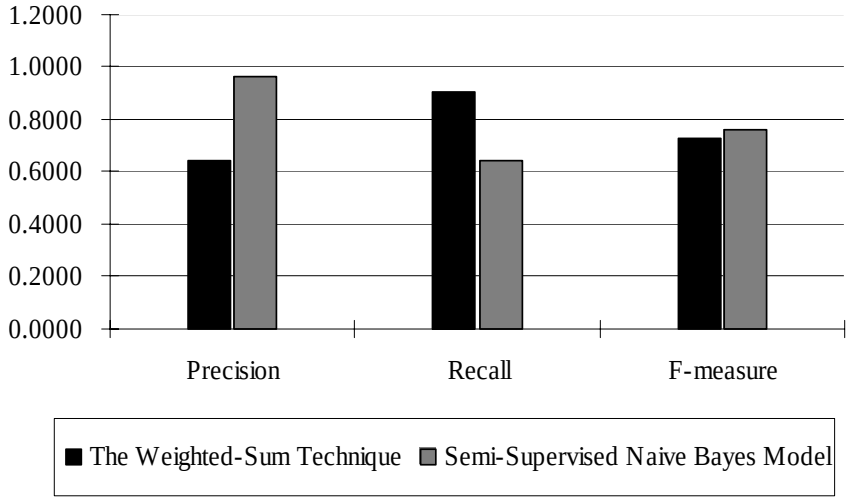


Figure.4.9 Semi-supervised naïve Bayes vs. weighted-sum technique

4.7 Conclusions

Identity information is critical to various organizational practices ranging from customer relationship management to crime investigation. The task of searching for a specific identity is difficult because multiple identity representations may exist due to issues such as errors and deception. In this chapter we proposed a probabilistic Naïve Bayes model that improved existing identity matching techniques. Experiments showed that the proposed model achieved F-measure ratings significantly better than those of the exact-search based technique. With 30% training instances, the semi-supervised Naïve Bayes model achieved a performance comparable to the fully supervised weighted-sum technique. This training method outperformed both fully supervised and unsupervised learning. Although unsupervised learning required no

human intervention, it did not train the model well and performed poorly in the task of identity matching. With only 10% training instances labeled, our proposed technique still achieved a high F-measure of 0.71.

We also have several caveats for this research. First, the identity problems that we studied were drawn from a law enforcement agency dataset. Problems in other domains such as customer relationship management might display characteristics different from those of criminal identities. Second, we assumed that for a pair of matching identities, the value-pairs of individual features would also be matches. This assumption made the learning easier but contradicted the fact that a pair of matching identities may have non-matching feature value-pairs.

CHAPTER 5: TACKLING MISSING VALUES AND SCALABILITY: AN ADAPTIVE DETECTION ALGORITHM

The last two chapters focus on matching techniques that improve the effectiveness of entity matching. In this essay we propose an adaptive detection algorithm that improves the efficiency and scalability of entity matching. The problem of missing values is also addressed in this chapter.

5.1 Introduction

Identity deception occurs when someone intentionally conceals his/her original identity, impersonates another individual's identity, or uses forged identity documents. One of the problems that identity deception may cause is financial loss. For example, England reports financial losses of at least £1.3 billion each year due to identity deception (HomeOffice, 2002). More importantly, criminals or terrorists using false identities may cause casualties and property damages too large to be quantifiable. Thus, the identity deception problem has become a central issue in law enforcement and intelligence agencies.

A fabricated identity is difficult for law enforcement or intelligence agents to uncover. Police officers often rely on computer systems to search a suspect's identity against history records in police databases. Generally, computer systems search using exact match queries. Even if the fabricated identity is similar to the original identity recorded in the law enforcement computer system, an exact-match query is unlikely to bring up that record. Techniques to perform inexact searches have been developed. They can be used to detect deceptive identities by finding records that are similar but

not exactly the same. However, most of these techniques are ad hoc and cannot be easily applied to real deception detection applications because of problems such as missing values and large volumes of data. Because a police database usually contains millions of criminal identity records, the detection techniques need to be efficient and scalable enough to examine all deceptive identities. In addition, for any large dataset, it is "unlikely that complete information will be present in all cases" (Kim & Curry, 1977). Missing values contained in past criminal records may greatly affect the accuracy of the detection techniques in finding deceptive identities because of the reduced information.

In this chapter, we aim to develop an automated approach that looks for inexact matches for fabricated identities. Such a technique is expected to search through past criminal identity records that may contain missing values and to be efficient enough to handle large volumes of data. In Section 2 we briefly discuss the identity deception problem and review some existing deception detection techniques. We also review techniques that handle the missing value problem and those that improve algorithm efficiency and scalability. We present our research questions in Section 3. In Section 4 we propose an adaptive detection algorithm for identity deception problems. This algorithm is able to utilize records containing missing values and is scalable to large volumes of identity data. We describe our experimental design in Section 5 and report the results and discussions in Section 6. We conclude our findings and future directions in the last section.

5.2 Related Work

5.2.1 Identity Deception

Identity is a set of characteristic elements that distinguish a person from others (Donath, 1998; Jones, 2001). There are three types of basic identity components (Clarke, 1999; HomeOffice, 2002): attributed identity, biometric identity, and biographical identity. Attributed identity is the information given to a person at birth, such as name, date and place of birth. Biometric identity contains biometric features that are unique to a person, such as fingerprints. Information that builds up over a life span comprises a person's biographical identity, examples of which are credit history and crime history, etc. Among these three types of identity components, attributed and biographical identities are often subject to deception while biometric features of a person are the most difficult to falsify.

Deception is "a sender's knowingly transmitting messages intended to foster a false belief or conclusion in the receiver" (Burgoon, Buller, Guerrero, Afifi, & Feldman, 1996). This definition originates from the interpersonal communication perspective and also applies to identity deception which usually occurs in an interactive environment (e.g., during an interrogation). We categorize three types of identity deception based on the method of deception: identity concealment, identity theft, and identity forgery.

Identity concealment is deceiving by omitting or changing details of the true identity (DePaulo & Pfeifei, 1986). For example, a person may report his birth date with an altered month or day, or provide a false first name along with his true last

name. This type of deception is popular when a subject unexpectedly encounters a law enforcement officer (GAO, 2004). Concealment could be more advantageous than using a completely fictitious identity to those who lie about their identities. Subjects may recall partially true information more easily than a completely fictitious identity when questioned repeatedly, because the true part of the concealed information serves as recall cues and cued recall may reconstruct memory better than recall without cues (i.e., free recall) (Cohen, 2001). Hence, the difficulty of recognizing such a deception (e.g., by law enforcement agents) is substantially increased. Identity theft, also called impersonation, is the action of one person illegally using another person's identity information for fraudulent purposes. Credit card fraud is a good example of identity theft. Identity forgery is committed through the use of forged or faked identity documents such as birth certificates, social security cards, and passports. This is common for illegal aliens who need forged documents in order to stay unnoticed and, yet, make a living (Toth, 2003).

In this research we mainly focus on the problem of identity concealment. We believe a solution to this problem can greatly improve crime investigation by law enforcement and intelligence agencies. We also hope that the solution proposed will be of value in detecting identity theft as well as forgery.

We provided evidence for the existence of identity concealment in previous study, in which a taxonomy of identity deception was built upon a case study of real criminal identity deception. We found that deception mostly occurs in specific attributes, namely, name, address, Date-Of-Birth (DOB), and ID number (e.g., the

Social Security Number). Name concealment, occurring in most deceptive cases, includes giving a false first name and a true last name or vice versa, changing the middle initial, giving a name pronounced similarly but spelled differently, etc. Concealment made on DOB can consist of, for example, switching places between the month of birth and the day of birth. Similarly, ID deception is often made by changing a few digits of a social security number or by switching their places. In residency deception, criminals usually change only one portion of the address. For example, the case study found that in about 87% of cases subjects provided a false street number along with the true street direction, name, and type.

Based on this case study, we observed that a concealed identity often partially matched with its original identity. We studied whether a certain technique could utilize such a characteristic and automatically detect this type of identity deception. In the next section we review techniques that can be used to detect identity deception.

5.2.2 Deception Detection Techniques

Detection techniques for general deception have been developed in the behavioral research fields, such as psychology, physiology, and communication. Techniques include the analysis of verbal cues (symptoms of verbal content that are used to determine truth and deception), observing non-verbal cues (indications conveyed through non-verbal communication channels such as facial expression), and measuring physiological reactions (e.g., polygraph lie detector) (Aubry & Caputo, 1980; Ekman, 1992; Vrij, 2000). However, detection results from these techniques are quite unreliable (DePaulo & Pfeifei, 1986; Ekman & O'Sullivan, 1991; Kohnken,

1987; Kraut & Poe, 1980). Moreover, these techniques are not automated processes and require human operators.

Practical detection techniques for identity deception are developed in law enforcement and intelligence communities. Firstly, police officers often use techniques such as repeated questioning and detailed questioning to validate the truthfulness of a suspect's identity. During the questioning process, inconsistent answers may disclose a false identity. However, those questioning methods are not reliable techniques, especially when dealing with good liars. Consequently, many deceptive records still exist in law enforcement databases. Secondly, after talking to the crime analysts of Tucson Police Department, we find that professional crime analysts can sometimes detect deceptive identities using link analysis techniques. By examining associations among criminals, organizations, and vehicles, a crime analyst is able to build criminal networks. When information about a suspect's identity is incompatible with known relationships represented in the criminal networks, the identity will be flagged as a possible deception. This technique, however, requires great amounts of manual information processing and is very time-consuming. In fact, it often serves as a post-investigative tool rather than a proactive investigation technique.

Some techniques that were initially designed for crime analysis can be used to detect identity deception. These techniques basically perform data association that links suspects to the crime being investigated, ordered from the most possible to the least possible. Brown and Hagen (2003) proposed a similarity-based data association

method for associating records of the same suspect or incidents having similar modus operandi. It compares corresponding description attributes of two records and calculates a total similarity measure between the two records. Experiments showed that associations suggested by the algorithm agreed with those made by experts. Both techniques introduced above are automated processes and can be used to detect identity deception by associating a suspect's identity with past criminal records. However, these methods only define similarity measures for categorical (e.g., hair color) and quantitative (e.g., height) attributes, but not for textual non-categorical attributes such as name and address.

A record comparison algorithm specifically targeting the detection of identity deception was proposed in our previous research (Wang et al., 2004a). This automated detection method makes use of string comparison techniques and searches for inexact matches of suspects' identities in police databases. This technique examines the attributes of name, address, DOB (date of birth), and SSN (Social Security Number) for each identity. It computes a disagreement measure between values in each corresponding attribute of two identities and calculates an overall disagreement value between the two identities as an equally weighted sum of the attribute disagreement measures. The formula for the overall disagreement value is as follows:

$$d = \sqrt{\frac{d_{Name}^{\ 2} + d_{Addr}^{\ 2} + d_{SSN}^{\ 2} + d_{DOB}^{\ 2}}{4}}$$

where d_{Name}, d_{Addr}, d_{SSN}, and d_{DOB} represent the disagreement measures in the fields of name, address, SSN, and DOB respectively. Each field value is considered a string of characters. Disagreement between two field values is computed by a string comparator,

namely the Levenshtein edit distance (Levenshtein, 1966), which calculates the minimum number of single character insertions, deletions, and substitutions required to transform one string to the other. Dividing the edit distance by the length of the longer string, each disagreement value is normalized between 0 and 1. If an overall disagreement value d between a suspect's identity and a past identity record is less than a threshold, which can be pre-determined by a training process, the algorithm suggests one identity is a deceptive form of the other. Experiments showed that this algorithm achieved high detection accuracy (94%). However, this method is quite inefficient for large-scale datasets. The computational time complexity of the algorithm is $O(N^2)$, because it compares each pair of records in a dataset. The computational time will increase exponentially as the size of the dataset increases. Furthermore, this method is unable to deal with identities that do not have values in all four fields (i.e., containing missing values).

The record comparison algorithm works better than data association algorithms for detecting identity deception, because it specifically captures the concealment deception patterns defined in the taxonomy introduced in the previous section. However, the problems with the record comparison algorithm, namely the inability to handle missing values and the inefficiency in processing large data volumes, prevent it from being used in any real world applications. In the next two sections we review techniques that handle the missing value problem and methods that improve the algorithm efficiency.

5.2.3 Missing Value Problem

Missing value is defined as values excluded from arithmetic calculations because they are missing (Clarke, 1999). In statistical analysis and data mining fields there are three major types of strategies that deal with the missing value problem: deletion, imputation, and adaptive data analysis.

Deletion (listwise or pair-wise deletion) (Buck, 1960; Glasser, 1964; Gyimah, 2001; Haitovsky, 1968; Kim & Curry, 1977) is the simplest technique to overcome the missing value problem and is easy to implement. Listwise deletion deletes or ignores those data records where missing values occur. Pair-wise deletion only excludes records missing information on the variables under examination (Gyimah, 2001; White, Liu, Hallissey, & Fielding, 1996). Both approaches may result in a great amount of information loss if the fraction of missing values is high (Gyimah, 2001; White et al., 1996). Also, deletion methods may lead to serious statistical biases if the missing values are not randomly distributed (Schafer, 1997).

Another alternative is imputation, which fills in missing values with plausible estimates (Allison, 2001; Schafer, 1997). Such a technique makes use of patterns or statistical associations found in complete records. These patterns are then applied to records with missing values, making estimates of the missing values in each record based on known attribute values. For example, mean imputation (Rubin, 1987) replaces a missing value with the mean of non-missing values of the same attribute. Some imputation methods can be complex due to the process of finding statistical patterns (Quinlan, 1986). However, imputation techniques can only make estimates on

numeric or categorical attributes, upon which statistical patterns can be built. Textual attributes, such as names or addresses, can hardly be estimated. Another disadvantage of imputation methods is, potentially biasing datasets by treating artificially imputed values as real ones in the subsequent data analysis (Myrtveit, Stensrud, & Olsson, 2001).

In cases where imputation methods cannot reasonably estimate, adaptive data analysis methods are usually developed in order to minimize the impact of missing values. Timm and Klawonn (1999) gave an example with the fuzzy c-means clustering algorithm, in which missing values were omitted and known ones were taken into account in calculating the center of each cluster. Quinlan (1986) developed an adaptive approach for missing values in decision tree problems. He reduced the information gain from testing an attribute *A* by the proportion of cases with missing values of *A*. Experiments showed that this approach performed better than that of dropping all incomplete cases (i.e., listwise deletion).

In conclusion, listwise or pair-wise deletion is not always desirable because they lead to great information loss when there are many missing values. For the problem of identity deception, imputation methods are not appropriate because identity attributes such as names and addresses are textual attributes to which imputation techniques simply do not apply. Therefore, an adaptive data analysis method that is suitable for our scenario needs to be developed in order to fully utilize the known attribute values and minimize the impact of those that are unknown.

5.2.4 Algorithm Efficiency and Scalability

The efficiency and scalability problem impacts many algorithms that process large amounts of data, such as algorithms for finding duplicate records from large databases involving millions of records. In order to find all duplicate records in a database, the most reliable way is to compare every record with every other record (Low, Lee, & Ling, 2001). Such a method apparently is the most inefficient, especially when it is applied to large databases, because of its time complexity ($O(N^2)$).

Much database research has focused on data comparison efficiency. Hernandez and Stolfo (1995) presented a sorted neighborhood method (SNM) for the so-called merge/purge problems, in which data were merged from multiple sources. The SNM has three steps: creating sorting keys, sorting data, and merging duplicates. A key is made by extracting a relevant attribute or a combination of relevant attributes. The selection of a key, determined mainly by domain dependent knowledge, is critical for final merging results (Hernandez & Stolfo, 1998). The dataset is then sorted by the selected key in the sorting phase. During the merging phase, a window of a fixed size sequentially moves through the sorted data set from the top. Every new record entering the window compares with the previous records in the window and looks for matching records. To maintain the fixed window size, the first record in the window is dropped when a new record enters a full window. The time complexity of the SNM is $O(wN)$ (the time complexity of the merging phase) if $w < log\ N$, or else $O(NlogN)$ (the time complexity of the sorting phase), where w is the window size and N is the total

number of records in the dataset. Experiments showed that the SNM could achieve high detection accuracy and greatly reduce running time. The SNM methods assume that duplicate records sorted by an appropriate key are located close to each other, which is not always the case. One may increase the window size to find potential duplicates, however this may increase the running time as well.

Monge (1997; Monge & Elkan, 1997) proposed an adaptive duplicate detection algorithm that further improved the detection efficiency over the SNM. Like the SNM, this method also starts by creating a sorting key and sorts the dataset with the key. While a window sequentially scans the sorted dataset, it does not compare each newly entering record with all existing records in the window. If there are duplicate records existing in the window, the newly entering record only compares with one of them and others are ignored. Therefore, the actual number of comparisons w' that a newly entering record makes within the window varies. The time complexity of this algorithm is $O(w'N)$, where w' is usually less than the window size w. Consequently, this adaptive detection method is much more efficient than the SNM. Experiments showed that the detection accuracies of both methods were similar (Monge, 1997).

5.3 Research Questions

In this research we aim to develop a technique that can automatically detect deceptive criminal identities in law enforcement and intelligence databases in an effective and efficient way. Such a technique is applicable to the following law enforcement scenarios:

1) Given a suspect's possibly false identity, the algorithm is able to locate relevant

identity records of the same individual in police databases. Therefore, the true identity of the suspect may be recovered and more information becomes available to assist the police investigation.

2) The algorithm detects deceptive identities by examining records currently existing in police databases. This requires an efficient algorithm that deals with large data volumes, especially when data are integrated from different sources.

We have identified a record comparison algorithm that is most appropriate for detecting identity deception. We aim to improve this algorithm using techniques that allow it to deal with missing values and make it efficient and scalable with large data volumes. Our research questions are:

1) Can the improved technique effectively detect deceptive identities with records having missing values?
2) Is the improved technique efficient and scalable enough to handle the large amount of identities in police databases while the detection accuracy is maintained?

5.4 Adaptive Detection Algorithm

We aim to develop a detection algorithm that can adapt to real world applications where missing values are prevalent and data volume is often on the order of millions. In this section we propose an adaptive detection algorithm for detecting identity deception. We use an improved version of the record comparison algorithm's process so that identities containing missing values can be compared based on known attributes. The new algorithm also incorporates the heuristics of Monge's adaptive

duplicate detection method. We expect the efficiency of the detection process to be highly improved.

We choose to use an adaptive analysis method to handle the problem of missing values. Our intention is to make use of as many known attribute values as possible and to ignore missing values. Deletion methods discard not only attributes that have missing values, but also some attribute values that are not missing. Statistics-based imputation methods try to impute missing values based on the statistical relationship between attribute values that are missing and those that are not. However, they require attributes to be either quantitative or categorical so that statistical relationship can be established. In our case, most of the attributes (e.g., names and addresses) are textual. Statistical relationships between these attributes do not make sense (for example, it would be strange to conclude that people named “George” usually live on “Broadway Blvd.”).

In the pair-wise record comparison algorithm identity records containing missing values are simply discarded (i.e., listwise deletion). In the proposed adaptive detection algorithm, only the missing attributes are ignored while other available attributes are used in comparing a pair of identities. Here we assume that every two identities being compared have at least one non-missing attribute. We also assume that two matching identities have similar values on all attributes. We modify the original formula given in the previous section:

$$d' = \sqrt{\frac{d_{Name}{}^2 + d_{Addr}{}^2 + d_{SSN}{}^2 + d_{DOB}{}^2}{a}},$$

where a is the number of attributes that are available in both identity records being compared. The disagreement measures on missing attributes are set to zero. The heuristic is similar to what police officers would do when they manually compare two identities. It is obvious that the higher the number of missing values, the less confident the overall disagreement is.

We apply Monge's algorithm to our proposed algorithm in order to improve efficiency. The first step of Monge's algorithm is to sort the dataset according to a key attribute. Sorting on some attributes may lead to better results than sorting on the others. The key attribute can be determined by a training process. However, no single key will be sufficient to catch all matching records in general (Hernandez & Stolfo, 1998). Hernandez and Stolfo suggested a multi-pass approach that executes several independent runs of the algorithm, each time using a different key attribute. On the other hand, the multi-pass approach will increase the computation time. In this study, we only consider the single-pass approach.

The procedure for the revised detection method is shown in Figure 5.1. First, the whole dataset is sorted by a chosen key attribute. The window size w is set in step 2, which defines the range of nearby records being compared. The window is represented as a priority queue, which can contain at most w elements (i.e., clusters). The algorithm sequentially examines each record R_i in the sorted dataset starting from the top. In step 7, R_i is first compared with the representative record (the record that represents the cluster; we use the first record of each cluster to simplify the process) of each existing cluster C_j in a priority queue q. If a comparison suggests a match (i.e.,

the disagreement value of the two records is less than a given threshold) between R_i and C_j's representative, R_i will be merged into C_j. If R_i fails to find a match, it will continue to compare with the non-representative records (i.e., records except the first one) of each C_j in q. If a match is found, R_i will be merged into the cluster where the matched record belongs. If R_i cannot be merged into any cluster in q (such as in the beginning when clusters do not exist in q), a singleton cluster is created for R_i in step 19 and is inserted into q in step 23. The lowest priority cluster in q (i.e., the cluster first put in the queue) will be dropped from q if a new cluster is inserted into an already full queue. If a dropped cluster contains more than one identity record, this indicates that deceptive identities are found.

An example would make this clustering process much easier to understand. Suppose the dataset is sorted on name and the windows size w (i.e., the capacity of the priority queue q) is set to 4. We start to look at the first record R_0 from the top of the sorted dataset. Because q is empty at the beginning, we do not have any clusters to compare against. Therefore, a new cluster C_0 is created with R_0 as its only record and is put in q. We then examine the next record R_1. We first compare R_1 with the representative record (R_0) of the only cluster C_0 in q (step 7). Suppose R_1 matches R_0 (i.e., the disagreement value of the two records is less than a given threshold), we include R_1 in C_0 (step 8) and go back to step 4 to examine the next record R_2. Similarly, R_2 is first compared with R_0, the representative record of cluster C_0. If the two records do not match, R_2 is compared with R_1, the non-representative record in C_0 (step 14). If R_2 and R_1 match, R_2 is included in C_0. If they do not match, a new cluster C_1 is created

with R_2 as its only record and becomes the second element in q. This procedure is repeated until all records are examined. The first cluster (for example, C_0) will be removed from q when q is full (i.e., the number of clusters in q is equal to w). Therefore, a new record will only be able to compare the records contained in q.

The time complexity of the proposed adaptive detection method becomes $O(w'N)$ (the time complexity of the merging phase) if $w'<logN$, or otherwise $O(NlogN)$ (the time complexity of the sorting phase), where w' is the window size and N is the total number of records in the dataset. Compared to the pair-wise comparison algorithms, the adaptive detection method is expected to be much more efficient.

```
procedure AdaptiveDetection ()
1:          Sort the data set according to a key field;
2:          Set a window size w;
3:          Create a priority queue q of size w;
4:          LOOP: record R_i in sorted dataset    //scan the sorted dataset sequentially
5:              IF: R_i is not a member of any clusters in q
        //Compare R_i with the representative record of each cluster in q
6:                  LOOP: cluster C_j in q
7:                      IF Distance(R_i , Representative(C_j)) < threshold
8:                          Union(R_i, C_j);    //include R_i to the cluster C
9:                          GOTO step 4
10:                 END IF
11:             END LOOP
        //if no match is found,
        //compare R_i with the rest records of each cluster in q,
12:             LOOP: cluster C_j in q
13:                 LOOP: R in cluster C_j
14:                     IF Distance(R_i , R) < threshold
15:                         Union(R_i, C_j);    //include R_i to the cluster C_j
16:                         GOTO step 4
17:                     END IF
                        END LOOP
18:             END LOOP
        //if no match is found, create a new cluster for R_i and enque
19:             C_new = NewCluster(R_i);
        //if q is full, dequeue the cluster that first entered q
20:             IF: Size(q) = w
21:                 q.Deque();
22:             END IF
23:             q.Enque(C_new);
24:         END IF
25: END LOOP
end AdaptiveDetection
```

Figure 5.1 Procedures of the adaptive detection algorithm

5.5 Experiments

In this section we aim to test the effectiveness and the efficiency of the proposed adaptive detection algorithm. Experiments are conducted to answer the following questions:

1) Will the detection accuracy be maintained when employing the adaptive detection algorithm?
2) Can the adaptive detection algorithm detect deceptive identity records that contain missing values?
3) How does the adaptive detection algorithm perform with large datasets?

5.5.1 Performance Matrix

Algorithm performance is measured in terms of detection effectiveness and efficiency.

1) Detection Accuracy: We evaluate the algorithm's detection accuracy by using three kinds of measures: recall, precision, and F-measure. Those measures are widely used in information retrieval (Salton, 1988). Precision, in this scenario, is defined as the percentage of correctly detected deceptive identities in all deceptive identities suggested by the algorithm. Recall is the percentage of deceptive identities correctly identified. F-measure is a well-accepted single measure that combines recall and precision.

Suppose a set of identities D contains m unique individuals and each individual has at least one identity. Each individual may have a set of different identities denoted as D_i ($1 \le i \le m$ and $|D_i| \ge 1$). Let d_{ij} ($1 \le i \le m, j \ge 1$) denote the j^{th} identity of the i^{th}

individual. The detection algorithm groups all identities into n clusters based on identified identity deception. That is, deceptive identities that are considered as referring to the same individual by the detection algorithm are grouped into the same cluster. Each cluster identified by the algorithm is denoted as C.

$$C_k = \{d_{ij} \mid d_{ij} \in D \,\&\, d_{ij} \text{ referring to the } k^{th} \text{ individual}\}$$

where $k = 1, 2, \ldots, n$. The clusters have the following properties:

$$C_k \cap C_{k'} = \varnothing$$
$$\bigcup_k C_k = D \quad .$$

Identities of the same cluster are considered to refer to the same person, while identities of different clusters are considered irrelevant. To make performance measures of clustering results comparable to those of the pair-wise comparison method, we convert the clustering results to a matrix that is often generated by the pair-wise comparison method. For example, suppose person A has two different identities $\{A_1, A_2\}$ while person B has three identities $\{B_1, B_2, B_3\}$. Suppose the adaptive detection algorithm identifies two clusters: $\{A_1, A_2, B_1\}$ and $\{B_2, B_3\}$. A pair-wise comparison matrix is constructed from the clusters as shown in Figure 5.2. Each superdiagonal element in the matrix represents the comparison result between any two identity records. It is labeled as one when the two identity records are grouped in the same cluster by the algorithm; otherwise, it is labeled as zero. We will have four outcomes defined in Table 5.1. In this example, we have TP=2, FP=2, TN=4, and FN=2.

	A1	A2	B1	B2	B3
A1		1	1	0	0
A2			1	0	0
B1				0	0
B2					1
B3					

Figure 5.2 A pair-wise comparison matrix constructed from the two clusters

Table 5.1 Classification of algorithm outcomes

	Identities of the same person	**Identities of different persons**
Identities considered to refer to the same person	True Positive (TP)	False Positive (FP)
Identities considered not to refer to the same person	False Negative (FN)	True Negative (TN)

Based on the algorithm outcomes, we compute recall and precision as the following:

$$call = \frac{TP}{TP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

A well-accepted single measure that combines recall and precision, called F-measure, is often considered an overall measure of accuracy and is defined as:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

2) Efficiency and Scalability: Efficiency is measured by the number of comparisons that the algorithm requires in order to detect all deceptive identities within a dataset. Algorithm completion time is a supplementary efficiency measure.

According to the Longman Web Dictionary, scalability of an algorithm can be defined as the degree to which the algorithm becomes more efficient as the data volume increases. We define scalability to be proportional to the number of identities processed per unit of time, i.e.,

$$Scalability \propto \frac{Number\ of\ records\ in\ a\ dataset}{Completion\ time}$$

5.5.2 Experimental Design

In our experiments we compared the performance of the proposed adaptive detection algorithm to that of the record comparison algorithm. We did not compare to the performance of other deception detection techniques, because they are not directly comparable. We aim to examine how the algorithm's performance improves when incorporating techniques that handle the problems of missing values and large volumes of data. We expect that those techniques developed in the proposed algorithm will also apply to other computational deception detection techniques reviewed in section 2.2 and will improve their performance.

The datasets of deceptive identities used in our experiments were manually extracted by our police detective expert who has served law enforcement for 30 years. The sampling method the expert used was convenience sampling, in which he looked through the list of all identity records and chose the deceptive identity records that he ran into. Because deceptive identities are sparsely distributed in the criminals'

database, convenience sampling is more feasible than random sampling to locate deceptive identity records for experimental purposes.

1) Test Bed: We chose criminal identity records stored in the Tucson Police Department (TPD) as our test bed. According to the U.S. Census Bureau, Tucson's population ranked 30th among U.S. cities with populations of 100,000 and greater. The Federal Bureau of Investigation also reported that Tucson's crime index ranked 20th highest among U.S. cities in 2001 and is higher than the national average. Therefore, data kept in the TPD is representative of those stored in other agencies in terms of variety and data volume.

The TPD maintains about 1.3 million person records in the database. Each record uniquely identifies a person by a set of identity attributes. In this experiment we only focus on four attributes in which identity deception usually occurs: name, address, DOB, and SSN. The name attribute of each identity record is mandatory in the TPD and always has a value. We found a large number of missing values in the other three attributes; 76% of these records contain missing values in at least one attribute. Among these incomplete records we found that 42% contain one missing attribute, 29% have two missing attributes, and 4% of the records were missing all attribute values except for name. The distribution of different missing types is shown in Table 5.2. Certain missing types, such as address-missing, DOB-missing, and address-DOB-missing, are rare in the TPD database. Since all fields except name can be missing in the TPD database, we chose name as the sorting key for the adaptive detection algorithm in hypotheses testing.

Table 5.2 Different missing types in identity records of the TPD

Missing type	Number of identity records	Percentage
Complete	311,151	24.139%
SSN-missing	540,849	41.960%
Address-missing	298	0.023%
DOB-missing	1,470	0.114%
SSN-Address-missing	293,595	22.777%
SSN-DOB-missing	83,952	6.513%
Address-DOB-missing	25	0.002%
SSN-Address-DOB-missing	57,634	4.471%
Total:	**1,288,974**	**100%**

2) Hypotheses Testing: We expect the proposed adaptive detection algorithm, as compared to the pair-wise record comparison algorithm, to improve its efficiency in detecting deceptive identities without losing detection accuracy. Although we do not expect detection accuracy to maintain when a dataset has several missing attributes and a large percentage of missing values, we want to find out what circumstances could cause significantly lower accuracy rates for incomplete datasets. We also aim to find out whether the adaptive detection algorithm can find deceptive identities within an acceptable time (e.g., in minutes) when the dataset is large (e.g., in the order of millions). The hypotheses for testing the above objectives are discussed below.

a) Evaluating accuracy and efficiency: We compare the performance of the adaptive detection algorithm to that of the record comparison algorithm. Two hypotheses are proposed to compare the efficiency and the detection accuracy of the two algorithms. We use statistical *t*-tests in the comparisons to indicate the significance of any differences.

Hypothesis 1 (H1): There is no significant difference in detection effectiveness between the adaptive detection algorithm and the record comparison algorithm.

Hypothesis 2 (H2): There is no significant difference in detective efficiency between the adaptive detection algorithm and the record comparison algorithm.

Testing dataset: A police detective with 30 years of experience helped us identify 210 deceptive criminal identity records from the TPD database. The dataset involved 75 criminal individuals, each of whom had an average of 3 identity records. These identity records contain no missing values. All the addresses were manually converted to a standard format consisting of a street number, a street direction, a street name, and a street type.

Testing procedure: A 10-fold validation method was employed to validate the performance of the two algorithms. The dataset was randomly equally divided into 10 folds. Each time we used 9 folds for training and 1 fold for testing. In each training session, we determined an optimal threshold that distinguished between similar (i.e., deceptive) and dissimilar (i.e., irrelevant) records, when the highest F-measure was achieved. The threshold was then applied to the next testing session. Accuracy measures, as well as the number of comparisons and the completion time, were recorded for each testing session. Performance measures of the two algorithms were compared using a statistical *t*-test.

b) Evaluating the effects of missing values: We compare the detection accuracy of the algorithm when using a complete dataset and when using an incomplete dataset. Again, *t*-tests were used to indicate whether there was a significant difference in the

algorithm's detection accuracy. In order to examine how different types of incomplete datasets may affect the algorithm's detection accuracy, we vary the missing attribute(s) (i.e., attributes where missing values may occur) in the dataset and the percentage of incomplete records in the dataset. We learned from the TPD database that identity records missing more than two attribute values are rare. Therefore, we tested with incomplete datasets having no more than two attributes containing missing values.

Hypothesis 3 (H3): With the adaptive detection algorithm, there is no significant difference in detection effectiveness between identities having all attribute values and identities having at most two missing attribute values.

Testing datasets: First we conducted experiments using artificial incomplete datasets. In the TPD database, deceptive identities with certain missing attributes (e.g., DOB-missing or address-DOB-missing) are rare. With artificially generated incomplete datasets, we can construct various types of incomplete datasets by adjusting the composition of missing attributes as well as the percentage of incomplete records in each dataset. Incomplete datasets were derived from the complete dataset used in the previous experiment. For each dataset, we randomly chose a percentage (from 10% to 90% with an increment of 10%) of records from which we removed values in the intended missing attribute(s).

Second, we used a real incomplete dataset that was directly extracted from the TPD database by our police detective. Our intention is to avoid any systematic errors that might be caused by the artificially generated incomplete datasets. From the TPD database, we were able to draw a dataset of 210 deceptive records in which missing

values occurred in SSN only. Deceptive records missing values in other fields were not identified, either because certain missing types (e.g., address-missing, DOB-missing) were rare in the TPD database or because the police expert was not able to identify deceptive identities based on limited available values (e.g., SSN-Address-missing, SSN-DOB-missing).

Testing procedure: For each missing type we tested the proposed algorithm for several iterations, each of which had a different percentage (ranging from 10% to 90%) of missing values in the dataset for the intended field(s). During each iteration we used a 10-fold validation method to test the algorithm's detection accuracy. As in the previous experiments, an optimal threshold value was determined when the highest F-measure was achieved during the training session. The detection accuracy measures of the algorithm were recorded during the testing session. *T*-tests were used to compare F-measures achieved by the algorithm using incomplete datasets to those acquired using a complete dataset.

c) Evaluating scalability: In terms of scalability, we compare the adaptive detection algorithm to the record comparison algorithm when used to detect deception in large datasets (e.g., on the order of millions).

Hypothesis 4 (H4): There is no significant difference in scalability between the adaptive detection algorithm and the record comparison algorithm.

Testing datasets: We randomly selected 10,000 criminal identity records from the TPD database as the starting dataset for our scalability testing. We then increased

the size of the selection by 10,000 at a time until all identity records in the TPD database (about 1.3 million) were included.

Testing procedure: For each selected dataset, we detected deceptive identities using the adaptive detection algorithm and the record comparison algorithm respectively. The scalability of each algorithm, as defined earlier, was computed for each test. A *t*-test was performed to compare the scalability difference between the two algorithms over different sizes of datasets.

5.5.3 Results and Discussions

5.5.3.1 The Effectiveness of the Adaptive Detection Algorithm

Table 5.3 shows the detection accuracy, in terms of F-measure, achieved by the adaptive detection algorithm and the record comparison algorithm respectively. A *t*-test showed that there was no significant difference between the two algorithms (*p*-value = 0.659).

Algorithm efficiency measures achieved by the two algorithms, in terms of number of comparisons and completion time, are also listed in Table 5.3. Hypothesis 2 was also tested with a *t*-test and was rejected at a significant level (p-value << 0.05). The result showed that the adaptive detection algorithm is more efficient than the pair-wise record comparison algorithm.

5.5.3.2 Adaptive Detection Algorithm in Handling Missing Values

1) Testing with Artificially Generated Missing Values: We used *p*-values of *t*-tests to indicate whether there was a significant difference in detection accuracy between using a complete dataset and using a dataset that contained a certain

percentage of missing values in certain attributes. For each type of incomplete dataset (i.e., values missing in certain attributes), we plot *p*-values against the percentage of incomplete identity records contained in a dataset in order to indicate the significant changes in the algorithm's effectiveness. The effect of the amount of missing values on detection accuracy is clearly visible.

Table 5.3 Comparison between detection effectiveness of adaptive detection algorithm and record comparison algorithm

Fold	F Measure	
	Adaptive Detection	**Record Comparison**
1	1.000	1.000
2	1.000	1.000
3	0.906	1.000
4	1.000	1.000
5	1.000	1.000
6	1.000	1.000
7	1.000	0.977
8	1.000	0.963
9	1.000	0.962
10	0.936	0.864
Avg	**0.984**	**0.977**

(a) Algorithm effectiveness in terms of F-measure

Fold	Number of Comparisons		Completion Time (msec)	
	Adaptive Detection	**Record Comparison**	**Adaptive Detection**	**Record Comparison**
1	116	210	5.950	9.432
2	123	210	5.731	8.311
3	105	210	5.033	8.404
4	144	210	7.337	8.615
5	97	210	4.647	8.509
6	132	210	7.516	8.480
7	132	210	5.868	8.249
8	151	210	7.229	8.725
9	105	210	5.003	17.280
10	134	210	6.775	8.843
Avg.	**123.9**	**210**	**6.109**	**9.485**

(b) Algorithm efficiency in terms of number of comparisons and completion time

P-value figures in Figure 5.3 indicate the adaptive detection algorithm's performance differences between using a complete dataset and using a dataset in which identity records contain missing values for one attribute. When values were only missing for SSN, the detection accuracy (F-measure) of the adaptive detection algorithm did not significantly decrease if the percentage of incomplete records was less than 30%. Similarly, when values were only missing for DOB, the detection accuracy of the adaptive detection algorithm did not lower significantly if the percentage of incomplete records was less than 18%. However, there were significant variations in the detection accuracy when values were missing in the address attribute, regardless of the percentage of incomplete records.

P-value figures in Figure 5.4 show the adaptive detection algorithm's performance differences between using a complete dataset and using a dataset in which identity records contain missing values for two attributes. When values were missing exclusively in SSN and DOB, the detection accuracy of the adaptive detection algorithm did not significantly decrease if the percentage of incomplete records was less than 12%. Similar to the one-attribute-missing case, detection accuracy varied when there were missing values in the address field.

In order to explain why the existence of missing values in the address field brought variations to the algorithm's detection accuracy, we examined the characteristics of address values in the complete dataset and compared them to the SSN and the DOB. For each attribute, the distribution of disagreement values between related identities (i.e., different identities referring to the same individual) is shown in

Figure 5.5. We noticed that the distribution for the address attribute is very different from that for DOB or SSN. DOB and SSN both have a skewed distribution such that identities pointing to the same person mostly have very similar DOB or SSN values. Address, however, has a bipolar distribution of disagreement values. In our dataset, identities of the same individual sometimes have similar address values and sometimes have very different address values.

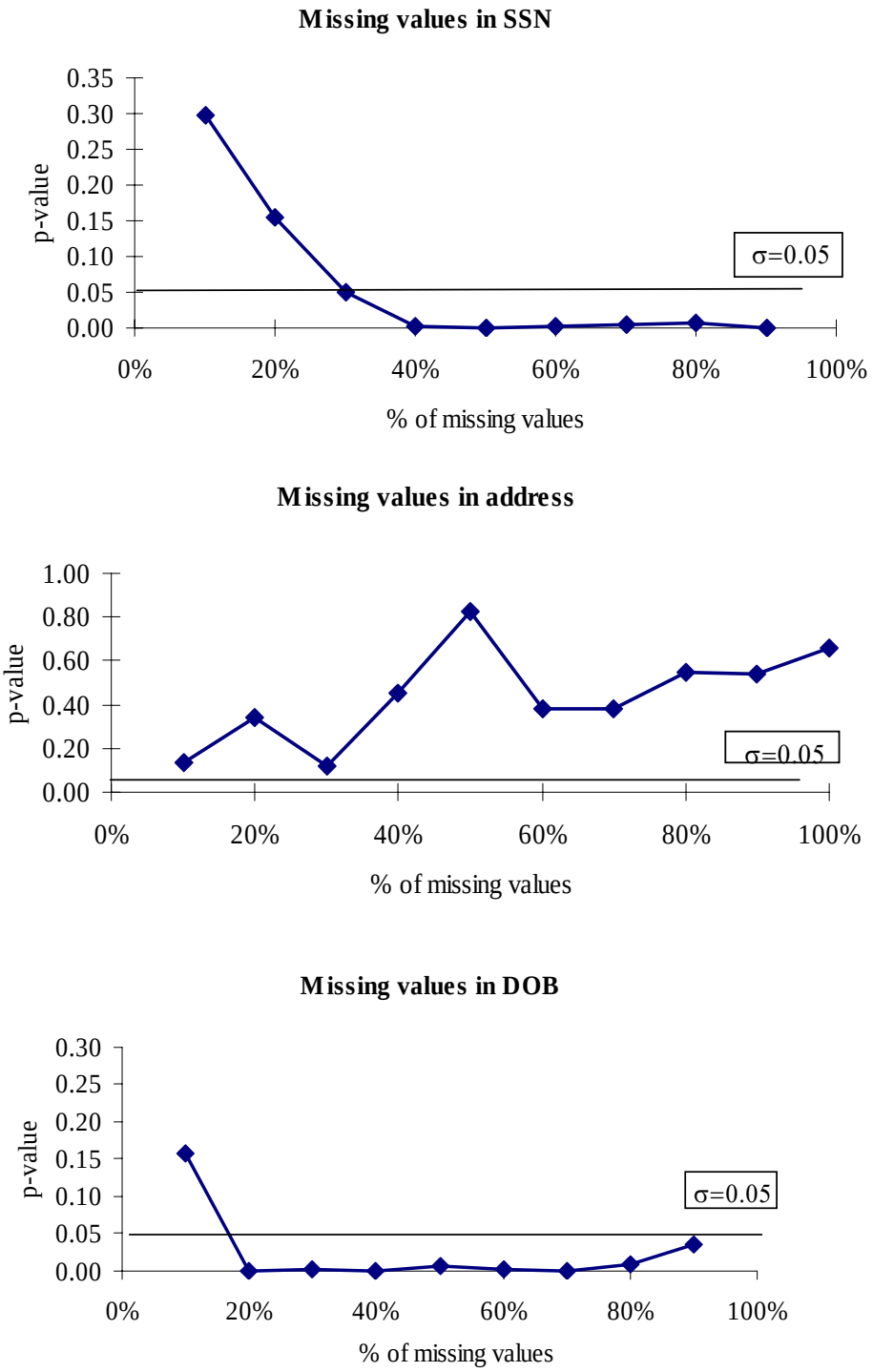


Figure 5.3 Performance comparison between the complete dataset and the datasets missing values in one attribute

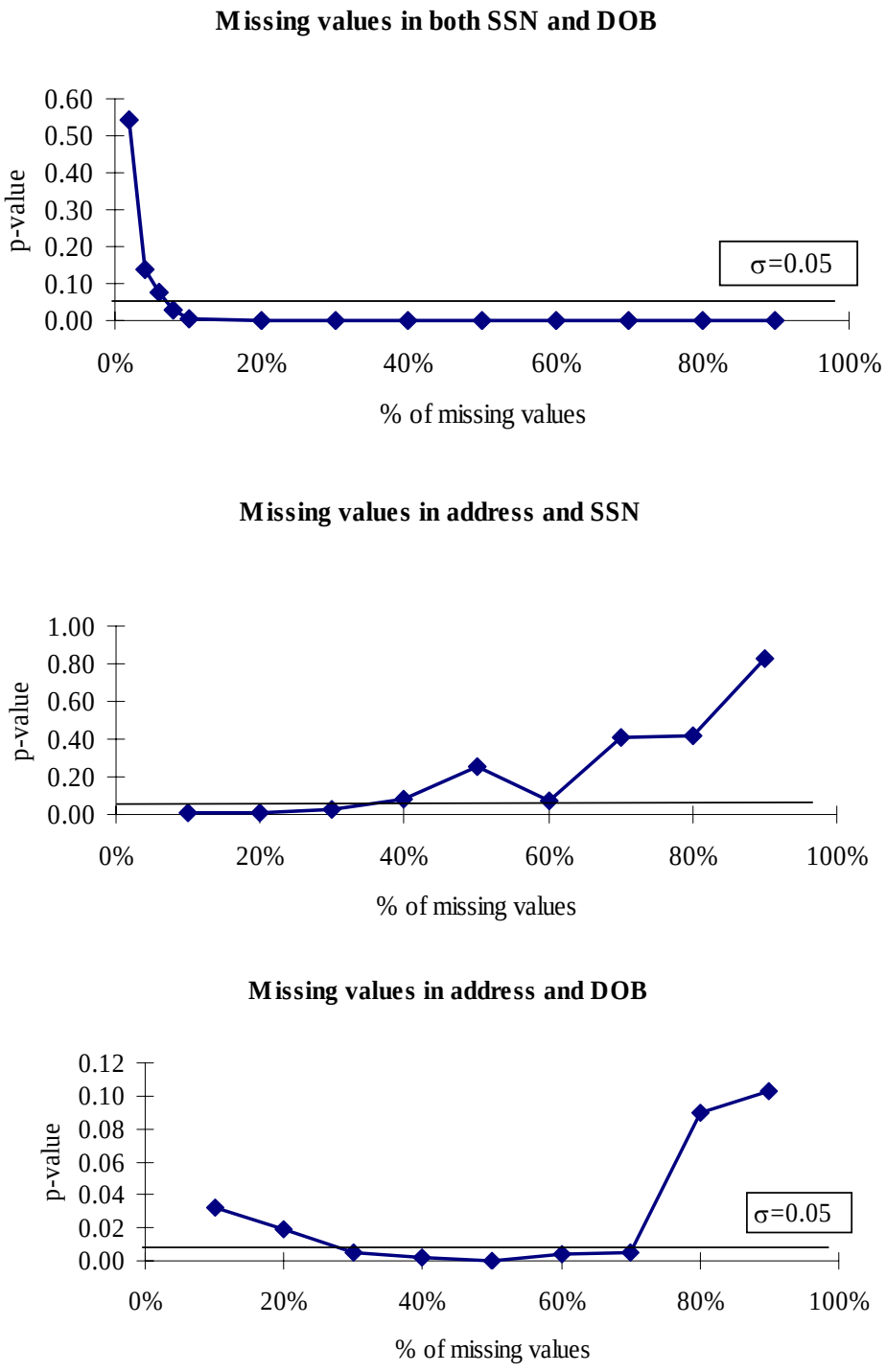


Figure 5.4 Performance comparison between the complete dataset and the datasets missing values in two attributes

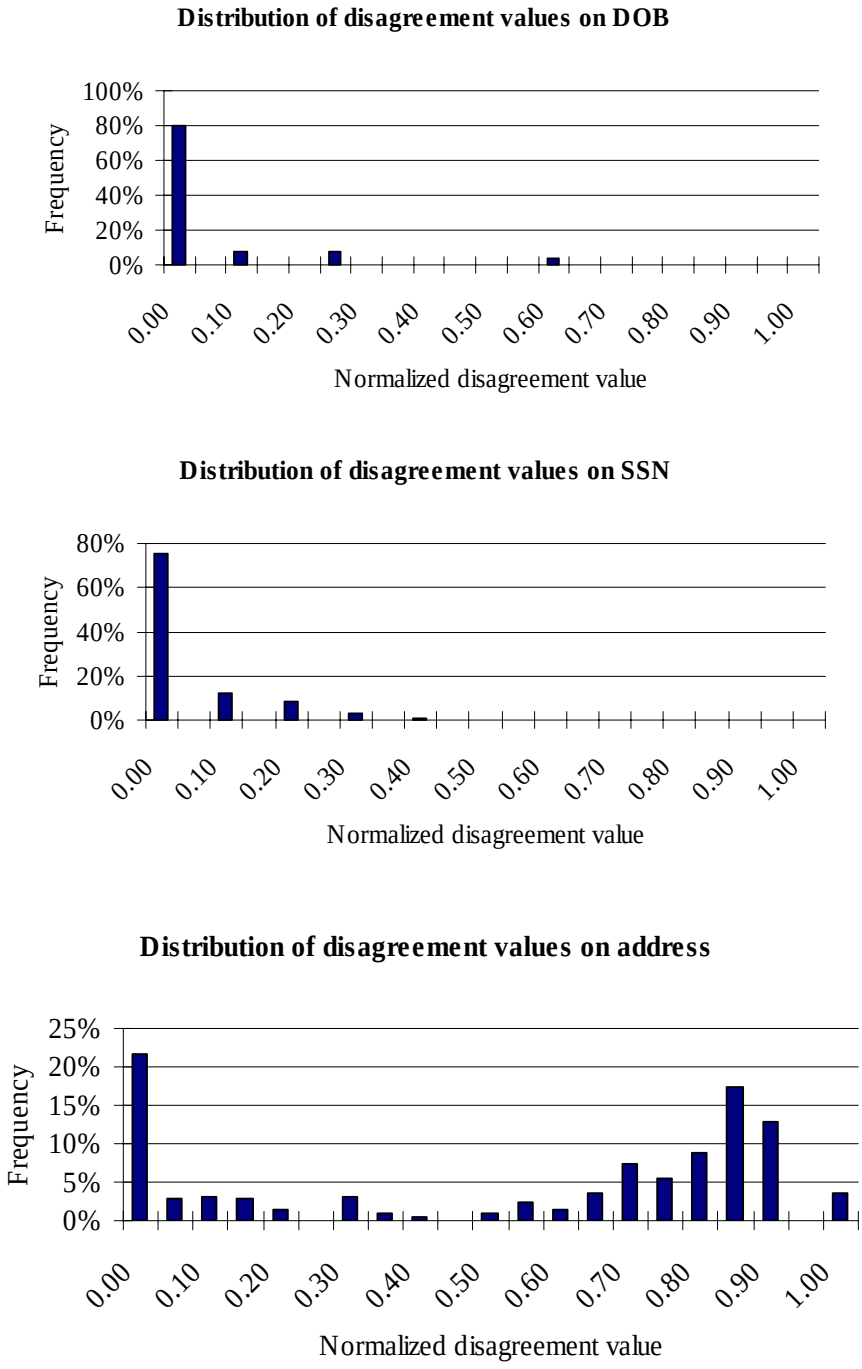


Figure 5.5 Distribution of disagreement values on each attribute.

Such a difference between address and the other two attributes might explain the difference in the algorithm's detection accuracy.

2) Testing with Real Missing Values: This dataset extracted from the TPD database had missing values in the SSN field only. As shown in Table 5.4, the adaptive detection algorithm was able to achieve on average a high precision of 93.7% and a recall of 73.6%. Compared to the detection performance using complete records, the detection precision was decreased for records with values missing in SSN. However, there was a significant decrease in the detection recall, which led to a significant drop in the overall F-measure. Two possible reasons may cause low detection recalls: either two identity records of the same individual are too far apart (e.g., much larger than the size of the sliding window in the adaptive detection algorithm), or the threshold value is too strict in determining deceptive identities.

Table 5.4 Detection performance with real missing values

Fold	**Recall**	**Precision**	**F-Measure**
1	0.786	1.000	0.880
2	0.500	1.000	0.667
3	1.000	1.000	1.000
4	0.417	1.000	0.588
5	0.750	1.000	0.857
6	0.846	0.846	0.846
7	0.846	0.786	0.815
8	0.615	1.000	0.762
9	0.688	1.000	0.815
10	0.917	0.733	0.815
Avg.	0.736	0.937	0.804

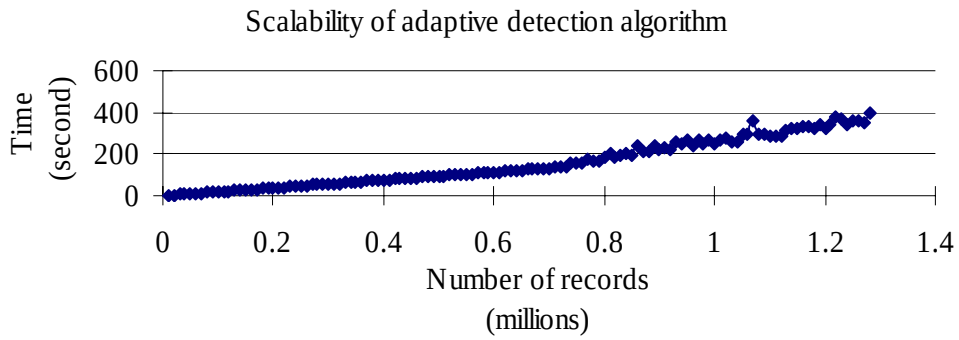
(a) Detection performance with records containing real missing values

Fold	Recall	Precision	F- Measure
1	1.000	1.000	1.000
2	1.000	1.000	1.000
3	1.000	1.000	1.000
4	1.000	1.000	1.000
5	1.000	1.000	1.000
6	1.000	1.000	1.000
7	1.000	1.000	1.000
8	0.857	1.000	0.923
9	1.000	1.000	1.000
10	0.880	1.000	0.936
Avg.	**0.974**	**1.000**	**0.986**

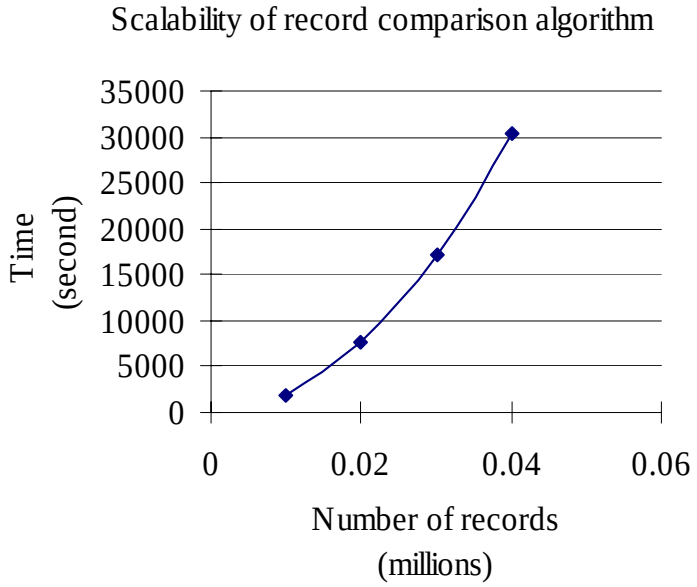
(b) Detection performance with complete records

5.5.3.3 Efficiency and Scalability

Scalability measures of the two algorithms are shown in Figure 5.6. The adaptive detection algorithm took 6.5 minutes for the adaptive detection algorithm to finish detecting deceptive identity in 1.3 million records. As the data volume increases, it maintained a gentle slope in the time it needed to finish detections. Note that the 6.5 minutes did not include the sorting time. Sorting was done within the database. It would add very minor overhead to the overall running time if the database was appropriately indexed. However, the detection time of the record comparison algorithm increased dramatically. It would have spent 87 days on the same task. Both algorithms were implemented in Java. Experiments were conducted on an HP PC with a Pentium-III 800MHz CPU and 256 MB ram.



(a) Scalability of the adaptive detection algorithm



(b) Scalability of the record comparison algorithm
Figure 5.6 Efficiency and scalability performance

5.5.3.4 A Case Study

To further evaluate the implication of our proposed algorithm, we tested it with another real dataset provided by the Pima County Sheriff Department (PCSD). PCSD serves 330,000 people living in the 7th largest county in the nation. We consider it as a representative of law enforcement agencies in the United States. The PCSD dataset contained over 1.3 million identity records. Residential address and

SSN information was not available in the dataset and was considered missing. We ignored those records that only had names because it is not reliable to determine deception solely by names. There were 700,686 identity records remaining in the testing dataset, each of which has values in the attributes of first name, last name, and DOB. With a window size of 10, our algorithm was able to identify 16,912 clusters. Identities of each cluster were considered to refer to the same person. We randomly chose 20 clusters and asked our police detective expert to evaluate each of them. The expert from the TPD confirmed that 11 out of 20 clusters were correctly grouped. There were six clusters that the expert from the TPD could not verify because of limited information. Three clusters were incorrectly clustered due to the use of common names and similar DOBs.

The expert from the TPD found this algorithm useful in finding both deceptive identity records and records that have data errors such as misspellings. Currently, the record management system used by this agency is not able to automatically group the identity records that refer to the same person. The six clusters that the expert from the TPD was unable to verify could also be useful in providing additional leads during investigation processes.

5.6 Conclusions and Future Work

In this chapter we discussed machine learning approaches to automatically detecting criminal identity deception. We proposed an adaptive detection algorithm that improved the record comparison algorithm in terms of efficiency, scalability, and ability to handle incomplete identities. Experiments showed that the proposed

algorithm greatly improved detection efficiency and achieved detection accuracy comparable to that of the pair-wise record comparison algorithm. Our experiments also showed that the detection accuracy of the adaptive detection algorithm was not affected when there was a small percentage of attribute values missing (less than 30% for missing values on SSN or less than 18% for missing values on DOB). In cases where there was a larger percentage of attribute values missing, the adaptive detection algorithm could still maintain detection precision of around 95%.

However, limitations exist in this research. The testing dataset is relatively small. The changing data characteristics of the testing dataset may affect the algorithm's performance. The algorithm's parameters (e.g., windows size of the priority queue and/or threshold values) may be adjusted when running the algorithm in a different dataset.

Our proposed algorithm assumes that all attributes are equally important. Therefore, it assigns an equal weight to each attribute when combining disagreement measures of the four attributes into an overall measure between two identity records. We may consider a different weighting schema. For example, in the future we may assign less weight to the address attribute because disagreement measures among related addresses introduce noise rather than contribute to the detection of deceptive identities. The assumption would also lead to the conclusion that two records, in which only the first name "John" was recorded, would have the same probability of describing the same person as two records, in which all of the fields exist. Intuitively, if name is the only available field to compare, one can only judge the probability that

two identities describe the same person solely by the names. However, the confidence in the match increases as more fields are available to compare.

One of the intentions of our proposed algorithm is to avoid pair-wise comparisons so that detection efficiency can be improved. However, detection effectiveness may be affected while the efficiency is improved under the assumption that two identities of the same individual sorted by an appropriate key are located close to each other. That assumption is however not guaranteed. It is possible that the two identities are located too far apart to be grouped into the same cluster. Although the algorithm did not cause a significant drop of detection efficacy in our experiments, we will consider more advanced clustering algorithms such as mixture models to avoid the assumption in future work.

In addition to detecting intentional deception, both record comparison algorithm and the proposed adaptive detection algorithm are capable of dealing with identity records having unintentional data errors such as misspellings. It might be interesting to differentiate between the patterns of deception and errors. However, we do not perceive any difference in terms of the algorithm's effectiveness.

In the future we intend to consider other identity- related information, such as biometrics, behavior characteristics, and social context. A good example of behavior characteristics is modus operandi (MO), which is often used to identify a criminal in crime investigation. The social context is a set of characteristics of the social system that a person usually lives. These types of information can also be helpful in determining a person's identity. The core function of our proposed algorithm is to

combine the disagreement measure of each of the four attributes and to determine the disagreement (or similarity) between two identity records. It is open to include more identity attributes when a disagreement measure can be defined for each attribute. A more comprehensive model that encompasses more identity attributes is desirable in future research.

CHAPTER 6: THE ARIZONA IDMATCHER: AN IDENTITY MATCHING SYSTEM USING THE MULTI-LAYER NAÏVE BAYES MODEL

In this essay we propose the Arizona IDMatcher, an identity matching system implemented using the multi-layer naïve Bayes model and the adaptive detection method.

6.1 Introduction

Identity management is central to various organizational practices, ranging from Customer Relationship Management in business domains to crime investigation in law enforcement. It is common to receive multiple copies of the same advertising mail from a company, where the only difference might be name spellings. In this case, bad practice in identity management costs money for the company and might negatively affect its customer relationships. In the homeland security domain identity management can be critically important. For example, the Transportation Security Administration (TSA) maintains a no-fly list consisting of the identities of those who are forbidden to board any aircraft. If a passenger's information matches one of the identities on the no-fly list, he/she is not allowed to board. However, the identity matching system employed by the TSA is not problem-free. There have been numerous reports of the system mistakenly matching innocent passengers to the identities on the list. That certainly caused trouble for the innocent passengers. On the other hand, if the TSA's system failed to identify a person on the list, the consequences would be more critical.

Identity management is challenging due to various quality issues. First, identity information is subject to data errors such as misspellings, inverted, or truncated information (McCallum-Bayliss, 2004). Second, intentional deception deteriorates the quality of identity information especially in some special domains such as financial fraud and crime (Wang, Chen, & Atabakhsh, 2004b). Third, legitimate identity changes may occur during one's life. For example, a woman may change her maiden name upon her marriage. All these issues may cause problems in retrieving or integrating identity information. Also, duplicate identity records may exist in record management systems. Previous studies have proposed several identity resolution techniques (Dey et al., 1998, , 2002; Fellegi & Sunter, 1969; Hernandez & Stolfo, 1998; IBM, 2006; Wang et al., 2004a). However, according to McCallum-Bayliss (2004), "no (identity resolution) system has yet to provide an approach to identity resolution with sufficient flexibility, adequate speed, and cultural understanding."

The rest of the chapter is organized as follows. In Section 6.2 we review two types of identity resolution techniques, namely heuristic matching techniques and algorithmic matching techniques. In Section 6.3 we propose the Arizona IDMatcher based on a probabilistic identity matching technique. In Section 6.4 we describe experiments that compare the performance of the Arizona IDMatcher against a leading identity resolution product, IBM Identity Resolution, whose design is based on a heuristic matching technique. We conclude our findings in Section 6.5.

6.2 Literature Review

6.2.1 Identity Features

Identity is a set of characteristic features that distinguish a person from others (Donath, 1998). Identity features can be categorized into three types: attributed identity, biometric identity, and biographical identity (Clarke, 1994). Attributed identity usually consists of identifiers assigned to an individual at birth, such as name, date and place of birth, mother's maiden name, and social security number. Biometric identity includes biometric features that are unique to an individual such as fingerprints, DNA, iris, hand geometry, and voice, among many others. Information that builds up over an individual's life span comprises the individual's biographical identity, examples of which are education and employment history, credit history, medical history, crime history, etc.

Identity records stored in law enforcement databases often contain attributed identities and sometimes biometric identities. The taxonomy of identity deception discussed in Chapter 2 indicates that similar values of name, DOB, ID numbers, and address prompt possible identity matches. Biometrics information is often touted as a reliable personal identifier. However, law enforcement databases only maintain fingerprints and DNA records for a small percentage of criminals because of the relatively high cost of collecting such data. On the other hand, biometric matching techniques may not identify individuals as effectively as people expect (Camp, 2003). Biometric information is subject to falsification as well (Matsumoto, Matsumoto, Yamada, & Hoshino, 2002).

6.2.2 Identity Matching Techniques

Based on the way decision models are constructed, existing identity matching techniques can be categorized into two types: heuristic techniques and machine learning techniques. Heuristic techniques often rely on domain experts to manually specify decision rules. With the expertise encoded, these techniques are expected to perform as well as human experts. Machine learning techniques use machine learning algorithms to automatically build decision models using a training dataset. Compared to heuristic techniques, they are more efficient with less or no human intervention. However, the decision models learned may be flawed due to factors such as noisy training data. In the rest of this section we review the two techniques respectively.

6.2.2.1 Heuristic Matching Techniques

Marshall et al. (2004) provided a simple identity matching heuristic based on domain experts' suggestions in a study on cross-jurisdictional information integration. The heuristic considers two identity records as referring to the same person only if their first name, last name, and date-of-birth (DOB) values are identical. Although effective, this technique is subject to great false negative rates. Because of the quality issues discussed earlier, it is very likely that identity records referring to the same individual may have disagreeing values in any of the three attributes.

An advanced heuristic matching technique, the IBM DB2 Identity Resolution, is designed for integrating identity records from different sources. Each incoming identity record is compared to the existing identities in its database to determine if the new record should be associated with an identity already recorded in the database. The

resolution process consists of four steps. First, the system generates a short list of existing identities that are possible matches to the incoming record. An existing identity is added to the candidate list when any of its key data elements (e.g., name, number, address) matches the corresponding data element of the incoming record. This step reduces search space by excluding the majority of existing identities whose key attribute values do not match those of the incoming record. Second, each identity from the candidate list is further evaluated by comparing to the incoming record. Names, identification numbers, addresses, and email addresses (if available) are compared to produce a baseline resolution score with a 100-point scale. The score is assigned by pre-defined rules. For example, if there is an exact name match, the resolution score increases 20 points. Certain rules provide immediate resolution decisions without further comparisons. For example, if the date-of-birth and the last name of a candidate identity are identical to those of the incoming record and the matching score of their first names is above 70, the incoming record is resolved to this candidate identity. The matching score of two first names is calculated by a name matching algorithm. However, to the best of our knowledge, the details of the algorithm are unknown. If the incoming record is unable to be resolved, the system proceeds to the confirm/deny attribute process if it is configured to do so. The resolution score between the incoming record and each identity from the candidate list is further refined based on user configurations. The last step compares the resulting resolution score of each pair of compared identities against a set of resolution rules to determine whether or not the incoming record should be resolved to an existing

identity or it represents a new identity. If the resolution score is greater than 100, they are resolved to the same identity. Otherwise, they remain separate identities. Once an incoming record has been resolved to an existing identity, the newly added information may prompt a resolution to another existing identity. Therefore, the last step is an iterative process, which sends the resolved identity though the resolution process again. This process continues until no more identities are resolved to an existing identity. As a commercial product, the IBM Identity Resolution provides a good system interface and connects to other tools, such as name and address standardization software, that may further improve the performance. However, deploying the system may require special expertise in system configurations for optimal system performance.

6.2.2.2 Machine Learning Matching Techniques

Machine learning matching techniques feature automated means of producing decision rules or models using machine learning algorithms. We categorize machine learning matching techniques into two categories based on the type of decision models. Deterministic techniques always produce the same decision (matching or non-matching) on different trials if the perceived similarities of two identities are the same. Probabilistic techniques, however, capture the uncertainty of matching by associating each matching decision with a probability rating between 0 and 1.

Deterministic Matching Techniques

A deterministic matching technique always produces matching decisions with certainty (the probability of 1). Given a pair of identities, it computes similarity scores

for each attribute value pair and combines them into an overall similarity rating. A matching decision is made when the overall similarity rating is greater than a matching threshold that can be determined by a training process. Weights may be used to represent relative importance of individual attributes when attribute similarity scores are combined into an overall rating.

Dey et al. (2002) proposed an integer programming approach for the general entity matching problem. In this approach attribute similarity scores are first combined into an overall similarity rating as a weighted-sum. Weights are elicited from users. Entities in two databases are matched in a way that minimized the total cost of type-I and type-II errors. The integer programming model is based on the assumption that one entity in a database can be matched to one and only one entity in the other database. However, this 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 attribute values of the two records. Experiments showed that this algorithm effectively identifies associations among records. However, this method lacks a training method to determine a matching decision threshold.

In a previous study we proposed a record comparison algorithm for detecting deceptive identities (Wang et al., 2004a). Assuming equivalent attribute importance, this technique calculates the overall similarity rating of an identity pair as a

normalized Euclidean distance over attribute similarity scores. A supervised training method determines a matching threshold using a set of identity pairs pre-classified by an expert. Matching decisions are made when future identity comparisons achieve similarity ratings greater than the threshold. Experiments showed that this technique was effective in linking matching identity records. However, the supervised training method makes this technique inefficient because manually generating the training dataset is time-consuming.

Probabilistic Matching Techniques

A probabilistic matching technique applies probability theories to capture the uncertainty of identity matching. Unlike a deterministic technique, a probabilistic technique associates a matching decision with a probability between zero and one. One advantage of probabilistic techniques is the ability to use unsupervised or semi-supervised learning, which requires minimal effort from human experts. Approaches based on probabilistic generative models are often used in these techniques to achieve unsupervised or semi-supervised learning.

Record linkage (RL) is a probabilistic technique originated in the area of statistics that identifies records corresponding to the same entity in one or more data sources. The basic ideas of record linkage were introduced by Newcombe et al. (1959). A formal mathematical definition was given by Fellegi and Sunter (1969). Given a comparison vector γ that consists of the attribute similarity scores of a record pair, RL calculates an odds ratio $R = m(\gamma)/u(\gamma)$, where $m(\gamma)$ is the probability that γ belongs to the matching set (M) and $u(\gamma)$ is the probability that γ belongs to the

non-matching set (U). A ratio greater than 0.5 means that the probability of a match is greater than the probability of a non-match. Two cut-off threshold values are determined according to the expected type-I and type-II error rates. Comparing R against the two threshold values, a record pair can be assigned into one of the three categories: matching, undecided, and non-matching. Undecided record pairs are subject to further clerical review. Early practices of record linkage use supervised learning to estimate the parameters such as $m(\gamma)$, $u(\gamma)$, and threshold values. Winkler (1998) and Jaro (1989) proposed unsupervised learning methods for record linkage. Their approaches assume that all comparison vectors are distributed according to a finite mixture with unknown parameters such as $m(\gamma)$ and $u(\gamma)$. An EM algorithm estimates these parameters by treating the class labels in a training dataset as missing. Although unsupervised learning avoids the time-consuming process of manually generating a training dataset, it only performed well in a few situations that were extremely favorable (Winkler, 2002). Studies also showed that unsupervised learning may not be preferable because unlabeled data alone often are not sufficient for training (Nigam, McCallum, Thrun, & Mitchell, 2000).

Bayesian network-based-techniques have been proposed for record linkage problems. Based on the formal Bayes theorem, a Bayesian network is close to human reasoning processes and produces classification results that can be easily interpreted. Ravikumar and Cohen (2004) modeled the problem of record linkage as a three-layer hierarchical graphical model (Figure 6.1) in an unsupervised setting. Experiments

showed that this technique achieved performance comparable to that of the supervised and semi-supervised record linkage techniques.

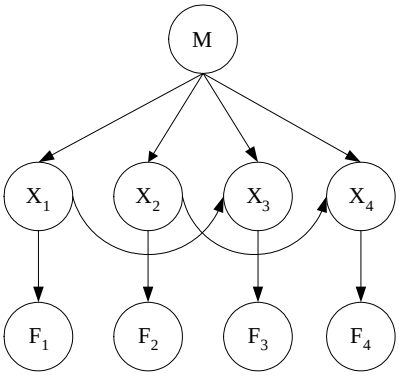


Figure 6.1 A three-layer hierarchical graphical model

In a previous study we proposed a multi-layer naïve Bayes model for identity matching by extending Ravikumar and Cohen's technique (Wang, Chen, & Atabakhsh, 2006). The advantage of our technique is four-fold. First, despite its strong assumption on attribute independency, the naïve Bayes model is often found more effective than general Bayesian Network models (Friedman, 1997; Langley, Iba, & Thompson, 1992; Titterington et al., 1981). Second, a multi-layer structure is able to model complex attribute dependencies. For example, the model shown in Figure 6.2 has four layers. Another layer of latent variable is added in order to represent the dependency between the full name matching decision and the matching decisions on three name components, namely first name, middle name, and last name. The third advantage comes with the use of semi-supervised learning. The inclusion of labeled data can exponentially reduce the probability of classification error (Castelli & Cover, 1995). Experiments showed that with 30% labeled training instances this technique achieved a performance not significantly different from that of fully supervised learning. Lastly,

the model is designed with a focus on the problem of identity matching. This model uses four key attributes, namely name, DOB, identification number and address, to uniquely represent an identity. Based on a previous case study on identity problems, those four attributes often provide useful hints for matching identities.

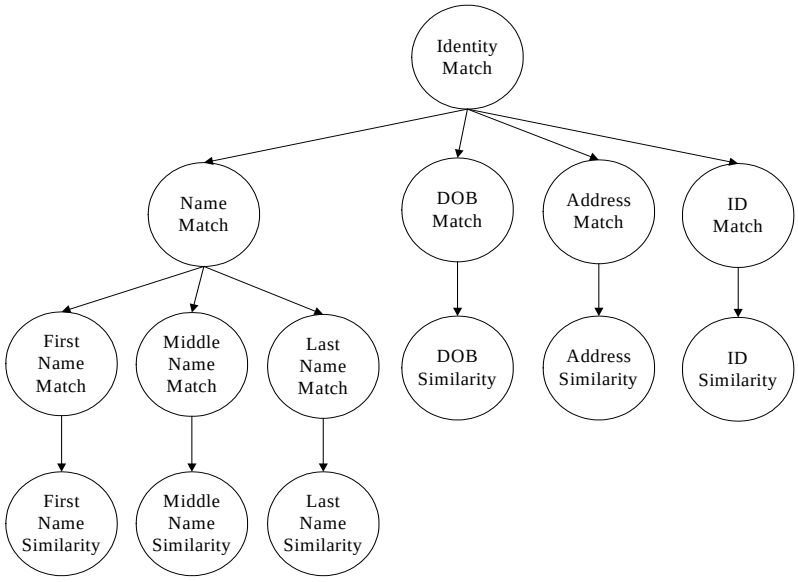


Figure 6.2 A multi-layer naïve Bayes model

6.2.3 Efficiency and Scalability Issues

The efficiency and scalability problem impacts many algorithms that process large amounts of data. It is a concern when matching identity records within or across systems that contain millions of records. The most reliable method is to compare every record with every other record. This pair-wise comparison, however, is the least efficient because of its poor computation complexity ($O(N^2)$).

Some studies in the area of database research have focused on the efficiency problem of record comparison. Hernandez and Stolfo (1995) presented a sorted neighborhood method (SNM) for the so-called merge/purge problems, in which data

were merged from multiple sources. The SNM has three steps: creating sorting keys, sorting data, and merging duplicates. A key is made by extracting a relevant attribute or a combination of relevant attributes. The selection of a key, determined mainly by domain dependent knowledge, is critical for final merging results (Hernandez & Stolfo, 1998). The dataset is then sorted by the selected key in the sorting phase. During the merging phase, a window of a fixed size sequentially moves through the sorted data set from the top. Every new record entering the window compares with the previous records in the window and looks for matching records. To maintain the fixed window size, the first record in the window is dropped when a new record enters a full window. The time complexity of the SNM is O(wN) (the time complexity of the merging phase) if $w < \log N$, or O($N\log N$) (the time complexity of the sorting phase), where w is the window size and N is the total number of records in the dataset. Experiments showed that the SNM could achieve high detection accuracy and greatly reduce running time. The SNM methods assume that duplicate records sorted by an appropriate key are located close to each other, which is not always the case. One may increase the window size to find potential duplicates, however this may increase the running time as well.

Monge (Monge, 1997; Monge & Elkan, 1997) proposed an adaptive detection algorithm that further improved efficiency over the SNM. Like the SNM, this method also starts by creating a sorting key and sorts the dataset with the key. While a window sequentially scans the sorted dataset, it does not compare each newly entering record with all existing records in the window. If there are duplicate records existing in the

window, the newly entering record only compares with one representative record and others are ignored when a match is found. Therefore, the actual number of comparisons *w'* that a newly entering record makes within the window varies. The time complexity of this algorithm is O(*w'N*), where *w'* is usually less than the window size *w* in the SNM. Consequently, this adaptive detection method is much more efficient than the SNM. Experiments showed that the detection accuracies of both methods were similar (Monge, 1997).

6.3 The Arizona IDMatcher

In this research we propose an identity matching system based on the multi-layer naïve Bayes model. The system is expected to achieve the following goals. First, it relies on minimal human effort to develop a matching decision model. Second, the system can be used in various domains with little modification and customization. Third, it is efficient in matching a large number of identity records within or across record management systems. In the rest of this section we provide an overview of our proposed identity matching system, the Arizona IDMatcher.

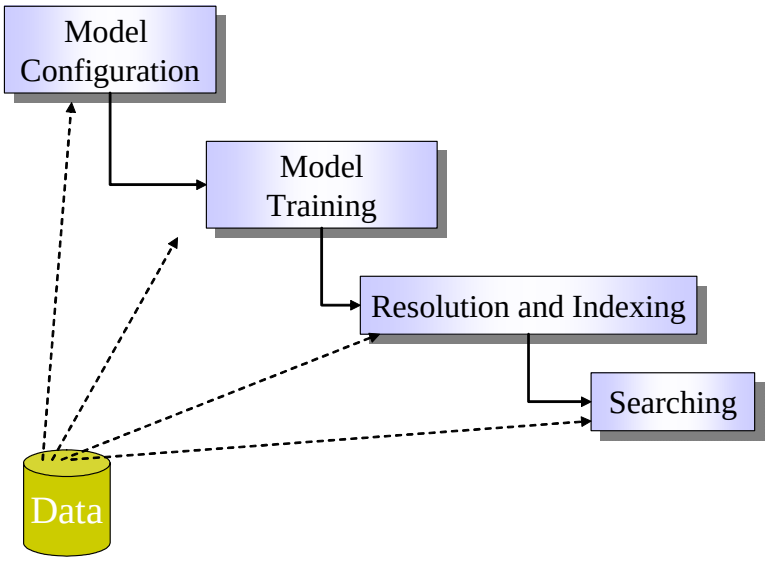


Figure 6.3 System overview of the Arizona IDMatcher

6.3.1 System Overview

As Figure 6.3 illustrates, the Arizona IDMatcher comprises four major components. We describe the functions and settings of each component in this section.

6.3.1.1 Model Configuration

This component enables users to view, modify, and define identity attributes used for identity resolution. By default, the system provides four built-in attributes including name, date-of-birth, identification number, and address. Default attribute dependencies are defined as the model shown in Figure 6.2. If a new attribute for identity resolution is defined, the user needs to specify the attribute type so that an appropriate similarity measure can apply. Table 6.1 lists the four attribute types supported by the system and the corresponding similarity measures. The user also needs to specify the dependencies of the new attribute in the naïve Bayes model.

Table 6.1 Attribute types supported and corresponding similarity measures

Attribute Types	Similarity Measures
Numeric	$1-\frac{\mid s_1 - s_2 \mid}{\max(s)-\min(s)}$
Binary	$\begin{cases}1, \text{when } s_1 \text{ and } s_2 \text{ agree}\\ 0, \text{otherwise}\end{cases}$
Nominal	$\begin{cases}1, \text{when } s_1 \text{ and } s_2 \text{ agree}\\ 0, \text{otherwise}\end{cases}$
Textual strings	$1-\frac{ED(S_1,S_2)}{MAX(\mid S_1 \mid,\mid S_2 \mid)}$ *

*: ED(•) is the Levenshtein edit distance function (Levenshtein, 1966) and |S| is a function that returns the length of the string S

6.3.1.2 Model Training

This component estimates the probability parameters of the multi-layer naïve Bayes model when training data are available. Some identity record management systems maintain a small portion of identity records whose associations have already been identified by previous human effort. For example, some criminal records in law enforcement databases may be identified and associated as a by-product of crime investigations. The system can construct a training dataset for the semi-supervised learning by combining a number of identity records whose associations are known with records whose associations are unknown. For the data sources where known associations do not exist, our system provides a default model with parameters trained in the previous study (Wang et al., 2006).

6.3.1.3 Resolution and Indexing

Once the decision model is parameterized, the system is ready to predict whether or not two identity records refer to the same person. The system assumes transitivity, which is defined as the following: if identity record *A* matches record *B* and *B* matches

record *C*, then *A* matches *C*. With the transitivity assumption the system produces a number of clusters of identity records. Records from the same cluster are considered as the same individual.

The indexing component defines the way in which identity records are chosen to be compared. The pair-wise comparison method compares every two identity records in one or more data sources. It is time-consuming when the number of records is large. The adaptive detection method is implemented based on Monge's algorithm. It is efficient by only comparing records whose key values are close. However, it requires users to specify a sorting attribute (e.g., last name) and may miss potentially matching records when the size of window *w'* is small.

6.3.1.4 Searching

The searching component provides two functions: verification and approximate search. With a verification interface users can browse and verify the matching decisions suggested by the system. The approximate search interface enables users to query the system about an identity and retrieve a list of possible matches. Possible matches are listed in the order of matching probabilities inferred by the multi-layer naïve Bayes model. Figure 6.4 illustrates the system architecture for implementation.

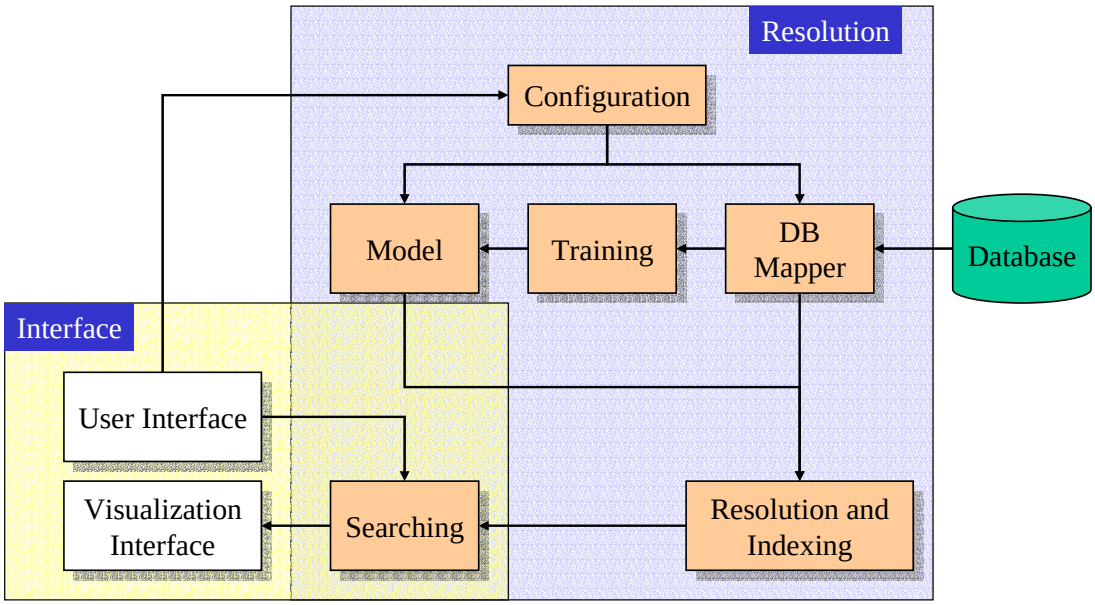


Figure 6.4 The Arizona IDMatcher system architecture for implementation

6.4 Experiment and Evaluation

In this section we evaluate the Arizona IDMatcher in terms of matching effectiveness and efficiency. We use the IBM Identity Resolution as our evaluation baseline.

6.4.1 Testbed

We used real criminal identity records from the Tucson Police Department (TPD) as our experimental testbed. We collected two datasets consisting of identity records that are related to gang crimes and narcotic crimes respectively. According to a police veteran who has worked in law enforcement for more than 30 years, identity records in these two datasets may present distinct characteristic in data completeness. Criminals related to gang crimes often undergo intensive police investigation. On the other hand, if a narcotic crime is not a felony these criminals may receive less attention from police investigators. In that case criminal identity information may be

incomplete and lack verification. Therefore, identity attributes of gang criminals have fewer missing values than those of narcotic criminals.

Each dataset contains a percentage of records that have been pre-associated in previous police investigations. The gang dataset contained 6,004 identity records with 2,195 unique identities while the narcotic dataset had 26,412 records with 15,386 unique identities. In our experiments we removed those previous associations and treated each identity record as a unique identity. However, there was one exception. When we trained the decision model in the Arizona IDMatcher, we used a small number of records with pre-associations as part of the training data.

6.4.2 Performance Measures

6.4.2.1 Matching Effectiveness

We measure the matching effectiveness using precision, recall, and F-measure, which are commonly used in the domain of information retrieval (Salton, 1988). Because matching results form clusters of identity records (each cluster represents a unique identity), we used the B-CUBED coreference scoring algorithm (Bagga & Baldwin, 1998) that computed precision, recall, and F-measure given two sets of clusters. One set is the truth set (i.e., the gold standard) and the other the output set produced by the IBM Identity Resolution or the Arizona IDMatcher. For an identity *i* in the output set, precision and recall are defined as follows:

$$\text{Precision}_i = \frac{number\ of\ correct\ elements\ in\ the\ output\ cluster\ containing\ identity_i}{number\ of\ elements\ in\ the\ output\ cluster\ containing\ identity_i}$$

$$\text{Recall}_i = \frac{number\ of\ correct\ elements\ in\ the\ output\ cluster\ containing\ identity_i}{number\ of\ elements\ in\ the\ truth\ cluster\ containing\ identity_i}$$

The overall precision and recall numbers for all output identities are computed by the following formulae:

$$\text{Precision} = \sum_{i=1}^{N} w_i * Precision_i$$

$$\text{Recall} = \sum_{i=1}^{N} w_i * Recall_i$$

where N is the total number of identities and w_i is the weight assigned to entity i. Bagga and Baldwin suggested the use of equal weights ($1/N$) for every entity i. F-measure is computed as:

$$\text{F-measure} = \frac{2 * Precision * Recall}{Precision + Recall}$$

6.4.2.2 Efficiency

Efficiency is measured by the completion time that a system spends in finding all matches within a dataset.

$$\text{Efficiency} \propto Algorithm\ Completion\ Time$$

6.4.3 Experimental Settings

We installed the IBM Identity Resolution Version 4.1.0 (released on May 26, 2006) on a Dell Dimension Desktop with a 2.5 GHz Pentium 4 CPU and 512MB RAM. The operating system of the computer is Windows XP Professional. The database that the IBM Identity Resolution connected to was Microsoft SQL Server 2000 with Service Pack 4. We used all default configurations that were shipped with the IBM Identity Resolution. We set the “Primary Matching Configuration” to “Default w/Name Only” as suggested by the user guide. The utility program “AFP” that came with the IBM Identity Resolution was used to convert datasets into XML

format so that they could be loaded into the system using “Pipeline,” another utility program.

We ran the Arizona IDMatcher on the same desktop computer. For the adaptive detection method, we arbitrarily set the window size to 20. Last name was used to sort the dataset because there were no missing values in last name and it has more distinguishing power than first name.

6.4.3.1 Testing with the Gang Dataset

We first evaluated the Arizona IDMatcher and the IBM Identity Resolution using the gang dataset. We ran the Arizona IDMatcher using three different indexing modes: pair-wise comparison, adaptive detection, and adaptive detection with local training. In the first two indexing modes we used the default parameters for the multi-layer naïve Bayes model. In the adaptive detection with local training mode, we first generated a training dataset by randomly selecting 1,000 record comparisons among records whose associations pre-existed in the dataset (i.e., labeled training instances) and 1,000 record comparisons among those whose associations were unknown (i.e., unlabeled instances). We estimated the parameters of the multi-layer naïve Bayes model using the semi-supervised learning technique proposed in (Wang et al., 2006).

To evaluate the performance of the two systems, we need a gold standard with record associations correctly identified. However, such a gold standard is difficult to identify. Existing record associations in the gang dataset cannot be used as a gold standard. Although the associations were made based on previous police

investigations and are accurate, it is still possible that many true associations among records have not been uncovered since all records have not been investigated. In this experiment we defined a pseudo gold standard that acted as a gold standard. Considering the three output sets produced by the existing record associations in the police dataset (DB), the IBM Identity Resolution (IBM), and the Arizona IDMatcher (AZ) respectively, a pseudo gold standard was defined as the existing record associations in the dataset combined with the overlapping matching results between the output sets of the IBM Identity Resolution and the Arizona IDMatcher. Figure 6.5 illustrates the definition, in which the area enclosed by the bold line represents the gold standard.

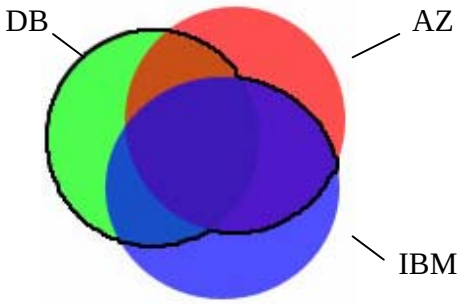


Figure 6.5 The pseudo gold standard

The pseudo gold standard may suffer from the following problems. First, the pseudo gold standard may have false matches when pre-associations are incorrectly made by police investigators or both the IBM Identity Resolution and the Arizona IDMatcher agree on an incorrect matching decision. Second, there might be true matches identified by either the Arizona IDMatcher or the IBM Identity Resolution. However, those true matches are not included in the pseudo gold standard because they are not identified by both systems.

The matching results for the gang dataset are summarized in Table 6.2. Among the three indexing modes of the Arizona IDMatcher, the pair-wise comparison took 11 hours to finish and was the least efficient. Therefore, we excluded pair-wise comparison from the indexing modes in later experiments. The two indexing modes using adaptive detection were the most efficient, taking only 22 seconds, while the IBM Identity Resolution spent about 21 minutes. The Arizona IDMatcher with adaptive detection and local training achieved the best recall. It also had the best compression ratio. The one with adaptive detection achieved slightly worse performance. Considering it was using default parameters, the results were still acceptable. The IBM Identity Resolution achieved the best precision, which means the matches it identified were mostly true matches.

Table 6.2 Matching performance for the gang dataset

	IBM Identity Resolution	**AZ IDMatcher (Pair-wise)**	**AZ IDMatcher (Adaptive)**	**AZ IDMatcher (Adaptive+ Local Training)**
Number of Identity Records	6,004			
Number of Unique Identities Identified by the System	2,147	2,210	2,218	**2,133**
Compression Ratio*	35.8%	36.8%	36.9%	**35.5%**
Completion Time	20M39S	10HR59M42S	**22S**	**22S**
Precision	**0.999734**	0.997934	0.998215	0.995392
Recall	0.996252	0.989007	0.987758	**0.998445**
F-Measure	**0.997990**	0.993450	0.992959	0.996916

*: Compression Ratio was computed as (Number of Unique Identities Identified by the System)/Number of Identity Records

6.4.3.2 Testing with the Narcotic Dataset

Table 6.3 summarizes the matching results for the narcotic dataset. Similar to the gang dataset, the Arizona IDMatcher performed more efficiently than the IBM Identity Resolution. The Arizona IDMatcher with adaptive detection and local training

achieved a better recall and a lower precision than that with adaptive detection only. The overall performance of the two indexing modes, however, was comparable for the narcotic dataset. The missing values in the narcotic dataset might affect the effectiveness of local training. The IBM Identity Resolution achieved a better recall and a better F-measure than the Arizona IDMatcher.

Table 6.3 Matching performance for the narcotic dataset

	IBM Identity Resolution	**AZ IDMatcher (Adaptive detection)**	**AZ IDMatcher (Adaptive detection+ Local Training)**
Number of Identity Records	26,412		
Number of Unique Identities Identified by the System	15,339	15,483	**15,337**
Compression Ratio	**58.1%**	58.6%	**58.1%**
Completion Time	2H29M33S	**3M01S**	**3M01S**
Precision	0.999458	**0.999539**	0.994301
Recall	**0.999910**	0.963226	0.967735
F-Measure	**0.999684**	0.981047	0.980838

We examined the details of those matching results produced by the Arizona IDMatcher. There were 284 clusters of records that did not agree with the pseudo gold standard. We randomly sampled 30 out of the 284 clusters and manually verified the matching results. Table 6.4 summarizes the verification results. Five clusters actually were true positive matches that both police investigators and the IBM Identity Resolution failed to identify. Those true matches should have been included in the gold standard. That means the recall, precision, and F-measure ratings of the Arizona IDMatcher were underestimated using the pseudo gold standard. Also, it is likely that the recall of the IBM Identity Resolution was overestimated because some true matches were missing from the pseudo gold standard. There were also six false positive matches in the sampled clusters. They occurred when their attribute values were similar but did not match. The other 19 clusters had missing true matches (i.e.,

false negative matches). Most of them were caused by last names that were common in this dataset (e.g., Smith, Martinez, Garcia, and Jones). As we introduced previously, the Arizona IDMatcher with adaptive detection only compares records within a window size. It is possible that two matching records with a common last name are still located too far apart after sorting on last name. . Hence, we proposed to sort the records not only by last name, but by last name and first name. We expected the extra sorting could improve the recall rating of the Arizona ID matcher, however, at the cost of longer processing time. Experimental results (Table 6.5) confirmed our expectation. The performance of the Arizona IDMatcher and the IBM Identity Resolution was close using the new sorting method.

Table 6.4 Verification of 30 identity clusters disagreed with the pseudo gold standard

Verification Result	**Number of Clusters**	**Cause**	**Example**
True positive	5	Attribute values were similar but not identical. These matches were omitted by both police investigators and the IBM Identity Resolution.	Name: Bryan vs. Brian Wilmer vs. Willmer Angie vs. Angelina DOB and ID numbers: 2 digits difference at most
False positive	6	Attribute values were similar but did not match.	Name: Donnely vs. Donohue Nancy vs. Randy Maria vs. Marcella Maria vs. Tania DOB and ID numbers: More than 2 digits difference
False negative	19	Common last names.	Garcia, Johnson, Jones, Gonzales, Smith, Martinez, etc.

Table 6.5 Experimental results when sorting by last name and first name.

	Arizona IDMatcher		**IBM Identity Resolution**
	Sort by Last Name Only	Sort by Last Name and First Name	
Number of Unique Identities Identified by the System	15,337	15,219	15,339
Compression Rate	41.9%	**42.4%**	41.9%
Completion Time	**3M01S**	13M36S	2H29M33S
Precision	0.994301	0.993665	**0.999886**
Recall	0.967735	**0.999962**	0.999962
F-Measure	0.980838	0.996803	**0.999924**

The values of effectiveness measures shown in the experimental results were close across different systems being evaluated. This was caused by the fact that the majority of the three output sets overlapped.

6.5 Conclusions

Various business practices require managing identity information in an effective and efficient way. However, the quality issues of identity information make this task non-trivial. Heuristic techniques such as the IBM Identity Resolution have been developed to tackle the identity matching problem. However, deploying such a system may require special expertise in system configuration and customization for optimal system performance. In this chapter we proposed an alternative approach, the Arizona IDMatcher. It relies on a machine learning algorithm to automatically generate a decision model for identity matching. Therefore, it needs minimal human effort. Experiments showed that the Arizona IDMatcher was extremely efficient in detecting matching identity records. Comparing to the IBM Identity Resolution, the Arizona IDMatcher achieved better recall ratings and comparable overall F-measure ratings.

We also have several caveats for this research. First, the pseudo gold standard created biased performance measures. Some true matches identified by the Arizona IDMatcher were missing. Therefore, the precision, recall, and F-measure ratings of the Arizona IDMatcher were underestimated and the recall of the IBM Identity Resolution was overestimated. Second, the IBM Identity Resolution may utilize third-party utility software (e.g., Name Manager, Group1 Address Standardization) to further improve its performance. We excluded the use of utility software in our experiments because we wanted to evaluate the Identity Resolution itself. In practice name and address standardization software may improve the performance for both IBM Identity Resolution and Arizona IDMatcher.

In the future we plan to consider biometric identity features into our identity matching model. Although biometric information is only available for a small percentage of criminal records in current law enforcement databases, it has its value in providing features that are more difficult to falsify than attributed features. As the cost of biometric technologies falls, the growth of biometric information is foreseeable. We plan to use both attributed features and biometric features so that identity matching may still proceed when one type of features is unavailable. The false rates of identity matching using both types of features are expected to be lower than those using

We believe the Arizona IDMatcher can be useful for law enforcement and intelligence agencies in their fight against crime and terrorism. We may also extend the use of the system to other types of entity matching such as vehicle and product.

CHAPTER 7: CONTRIBUTIONS AND FUTURE RESEARCH

Due to the rapid development of information technologies, especially the network technologies, business activities have never been as integrated as they are now. Business decision making often requires gathering information from different sources. However, many information systems were developed without the consideration of being compatible to one another. Other problems such as poor data quality and information overload may also arise during information retrieval and integration across heterogeneous information systems.

This dissertation focuses on the problem of entity matching, the process that associates corresponding information elements within or across information systems. It is devoted to providing complete and accurate information for business decision making. A series of studies are presented to address various challenges faced in entity matching. This section concludes this dissertation by summarizing the theoretical, technical, and empirical contributions, discussing its relevance to MIS research, and proposing future research directions.

7.1 Contributions

This dissertation makes *theoretical contributions* in the following perspectives.

- First, it provides a theoretical foundation for entity matching by connecting the similarity and categorization theories in cognitive science to entity matching techniques. The theories provide guidance in the development of entity matching techniques. For example, the feature

contrast similarity model (Tversky, 1977) is considered superior to geometric models because it is similar to the human reasoning process. Our proposed multi-layer naïve Bayes model follows the feature contrast model by modeling entities as a vector of features rather than a point in a multidimensional space.

- Second, the case study on criminal identity deception contributes to the special domain of law enforcement by building a taxonomy of criminal identity deception patterns. The findings provide a theoretical basis for developing advanced identity matching techniques.

Our studies make *technical contributions* in the three challenges of entity matching.

- First, we proposed a case-study-based methodology for identifying key entity features. Other features are excluded from entity matching so that the matching process can be more efficient.
- Second, we proposed both a deterministic matching technique and a probabilistic matching technique for entity matching. Experiments showed that both techniques were effective in linking deceptive criminal identities. The probabilistic matching technique used a semi-supervised learning method, which required less human intervention in the training process and achieved matching performance comparable to that of the deterministic matching technique with fully supervised learning.

- Third, we proposed to apply the adaptive detection algorithm (Monge, 1997) to entity matching. This technique reduced the matching search space and significantly improved matching efficiency without significant accuracy loss.
- The last contribution is the use of a system development approach to evaluating the impacts of the technology. It is an integral part of theory building, observation, and experimentation. Problems encountered during this process can be used to modify the theories and techniques from which the system is derived.

Empirical contributions of this dissertation are as follows.

- We compared the Arizona IDMatcher against the IBM Identity Resolution, a leading industry product developed based on heuristic decision rules. The experimental results provided recommendations for users in the field of identity matching. The Arizona IDMatcher was able to capture more true matches than the IBM Identity Resolution (i.e., high recall). However, the matches identified by the IBM Identity Resolution were mostly true matches (i.e., high precision). Users may choose an identity matching system based on their needs.
- Throughout the dissertation we focused on the problem of identity matching in the law enforcement domain. We chose this topic because the identity problem is central to various law enforcement and

intelligence activities. Research findings on this problem may impact important issues related to national security.

7.2 Relevance to MIS Research

The decision making process often combines different sources of data and knowledge available in various forms (Bolloju, Khalifa, & Turban, 2002). One of the knowledge management principles that helps achieve collaborative knowledge bases is to provide tools to transform scattered data into meaningful business information and support all types of decision makers (Ba, Lang, & Whinston, 1997; Bolloju et al., 2002). However, data heterogeneity makes it difficult to combine data from different sources and to transform scattered data into meaningful information. Thus, entity matching techniques are necessary in all kinds of information processing and knowledge management tasks that gather data from distributed sources.

Hevner et al. (2004) categorized much of the research in the Information Systems discipline into two paradigms: behavioral science and design science. This dissertation follows the design science paradigm because it proposes IT artifacts that "extend the boundaries of human and organizational capabilities." Our proposed entity matching techniques rely on machine learning algorithms that automatically generate decision models with minimal human intervention. Our goal is to achieve matching effectiveness similar to what a human can achieve, but with efficiency that is beyond a human's capability.

7.3 Future Research Directions

The entity matching techniques based on machine learning algorithms can still be improved in terms of matching effectiveness. As Chapter 6 indicates, machine learning matching techniques are often good at recall numbers but suffer in precision. It is a natural extension to improve the machine learning techniques further in terms of precision.

In addition, other related issues such as data availability and privacy can be explored. In this dissertation we assume accesses to different information sources are granted full control. In the real world, there might be information access policies that impose constraints on the data made available by different sources. Privacy is a big concern when integrating identity information across different systems. How to develop an entity matching system that provides effective and efficient decision support without infringing privacy and security policies is still an open question.

REFERENCES

Allison, P. D. (2001). *Missing Data*. Thousand Oaks, CA: Sage.

Anderberg, M. R. (1973). *Cluster Analysis for Applications*. New York: Academic Press.

Ashby, F. G., & Alfonso-Reese, L. A. (1995). Categorization as Probability Density Estimation. *Journal of Mathematical Psychology, 39*, 216-233.

Ashby, F. G., & Perrin, N. A. (1988). Toward a Unified Theory of Similarity and Recognition. *Psychological Review, 95*(1), 124-150.

Aubry, A. S. J., & Caputo, R. R. (1980). *Criminal Interrogation* (3rd ed.): Charles C Thomas Publisher.

Ba, S., Lang, K. R., & Whinston, A. B. (1997). Enterprise Decision Support Using Intranet Technology. *Decision Support Systems, 20*, 99-134.

Bagga, A., & Baldwin, B. (1998). *Algorithms for Scoring Coreference Chains.* Paper presented at the The First International Conference on Language Resources and Evaluation Workshop on Linguistics Coreference.

Bilenko, M., Mooney, R., Cohen, W. W., Ravikumar, P., & Fienberg, S. (2003). Adaptive Name Matching in Information Integration. *IEEE Intelligent Systems, 18*(5), 16-23.

Bolloju, N., Khalifa, M., & Turban, E. (2002). Integrating Knowledge Management into Enterprise Environments for the Next Generation Decision Support. *Decision Support Systems, 33*, 163-176.

Brown, D. E., & Hagen, S. C. (2003). Data Association Methods with Applications to Law Enforcement. *Decision Support Systems, 34*(4), 369-378.

Buck, S. F. (1960). A Method of Estimating Missing Values in Multivariate Data Suitable for Use With an Electronic Computer. *Journal of Royal Statistical Society, B22*, 302-306.

Buller, D. B., & Burgoon, J. K. (1998). Emotional Expression in the Deception Process. In P. A. Andersen & L. K. Guerrero (Eds.), *Handbook of Communication and Emotion* (pp. 381-402). San Deigo, CA: Academic Press.

Burgoon, J. K., Buller, D. B., Guerrero, L. K., Afifi, W., & Feldman, C. (1996). Interpersonal Deception: XII. Information Management Dimensions

Underlying Deceptive and Truthful Messages. *Communication Monographs, 63*, 50-69.

Camp, J. (2003). *Identity in Digital Government.* Paper presented at the 2003 Civic Scenario Workshop: An Event of the Kennedy School of Government, Cambridge, MA 02138.

Castelli, V., & Cover, T. M. (1995). On the Exponential Value of Labeled Samples. *Pattern Recognition Letters, 16*(1), 105-111.

Chen, A. L. P., Tsai, P. S. M., & Koh, J. L. (1996). Identifying Object Isomerism in Multidatabase Systems. *Distributed and Parallel Databases, 4*(2), 143-168.

Clarke, K. C. (1999). *Getting Started with Geographic Information Systems* (2nd ed.). Upper Saddle River, NJ: Prentice Hall.

Clarke, R. (1994). Human Identification in Information Systems: Management Challenges and Public Policy Issues. *Information Technology & People, 7*(4), 6-37.

Cliffton, C., Housman, E., & Rosenthal, A. (1997). *Experience with a Combined Approach to Attribute-Matching across Heterogeneous Databases.* Paper presented at the 7th IFIP 2.6 Working Conference on Database Semantics, Leysin, Switzerland.

Cohen, J. (2001). Errors of Recall and Credibility: Can Omissions and Discrepancies in Successive Statements Responably be Said to Undermine Credibility of Testimony? *Medico-Legal Journal, 69*, 25-34.

Dechter, R. (1996). *Bucket Elimination: A Unifying Framework for Probabilistic Inference.* Paper presented at the 12th Conference on Uncertainty in Artificial Intelligence, Portland, Oregon.

Deen, S. M., Amin, R. R., & Taylor, M. C. (1987). Data Integration in Distributed Databases. *Ieee Transactions on Software Engineering, 13*(7), 860-864.

DePaulo, B. M., & Pfeifei, R. L. (1986). On-the-job Experience and Skill at Detecting Deception. *Journal of Applied Social Psychology, 16*, 249-267.

Dey, D., Sarkar, S., & De, P. (1998). A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases. *Management Science, 44*(10), 1379-1395.

Dey, D., Sarkar, S., & De, P. (2002). A Distance-Based Approach to Entity Reconciliation in Heterogeneous Databases. *IEEE Transactions on Knowledge*

and Data Engineering, 14(3), 567-582.

Donath, J. S. (1998). Identity and Deception in the Virtual Community. In M. Smith & P. Kollock (Eds.), *Communities in Cyberspace*. London: Routledge.

Ekman, P. (1985). *Telling Lies: Clues to Deceit in The Marketplace, Politics and Marriage*. New York: W. W. Norton.

Ekman, P. (1992). *Telling Lies: Clues to Deceit in the Marketplace, Politics and Marriage* (3rd ed.). New York: W. W. Norton.

Ekman, P., & O'Sullivan, M. (1991). Who Can Catch a Liar? *American Psychologist, 46*(9), 913-920.

Fan, W. G., Lu, H. J., Madnick, S. E., & Cheung, D. (2001). Discovering and reconciling value conflicts for numerical data integration. *Information Systems, 26*(8), 635-656.

Fan, W. G., Lu, H. J., Madnick, S. E., & Cheung, D. (2002). DIRECT: a system for mining data value conversion rules from disparate data sources. *Decision Support Systems, 34*(1), 19-39.

FBI. (2001). *September 11 Hijackers: Names and photographs on FBI.GOV.* Federal Bureau of Investigation, U.S. Department of Justice. Available: http://www.fbi.gov/pressrel/pressrel01/092701hjpic.htm.

Fellegi, I. P., & Sunter, A. B. (1969). A Theory for Record Linkage. *Journal of the American Statistical Association, 64*(328), 1183-1210.

Friedman, J. H. (1997). On bias, variance, 0/1 - Loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery, 1*(1), 55-77.

GAO. (2004). *Law Enforcement: Information on Timeliness of Criminal Fingerprint Submissions to the FBI* (GAO-04-260): United States General Accounting Office (GAO).

Gill, L. (2001). *Methods for Automatic Record Matching and Linkage and Their Use in National Statistics* (Technical Report National Statistics Methodological Series No. 25, National Statistics). London: Oxford University.

Glasser, M. (1964). Linear Regression Analysis with Missing Observations among the Independent Variables. *Journal of the American Statistical Association, 59*, 834-844.

Gyimah, S. O. (2001). *Missing Data in Quantitative Social Research*. London, Canada: Department of Sociology, The University of Western Ontario.

Haitovsky, Y. (1968). Missing Data in Regression Analysis. *Journal of Royal Statistical Society, B30*, 67-82.

Hample, D. (1980). Purposes and Effects of Lying. *Southern Speech Communication Journal, 46*, 33-47.

Hand, D. J., & Yu, K. (2001). Idiot's Bayes - Not So Stupid After All? *International Statistical Review, 69*(3), 385-398.

Hernandez, M. A., & Stolfo, S. J. (1995). *The Merge/purge Problem for Large Databases.* Paper presented at the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, CA.

Hernandez, M. A., & Stolfo, S. J. (1998). Real-world Data is Dirty: Data Cleansing and the Merge/purge Problems. *Data Mining Knowledge Discovery, 2*(1), 9-37.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly, 28*(1), 75-105.

HomeOffice, U. K. (2002). *Identity Fraud: A Study*: United Kingdom HomeOffice.

IBM. (2006). Entity Analytic Solutions: http://www-306.ibm.com/software/data/db2/eas/.

Identity Theft and Assumption Deterrence Act, (1998).

Jaro, M. A. (1976). *UNIMATCH: A Record Linkage System, User's Manual*. Washington, DC: U.S. Bureau of the Census.

Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association, 89*, 414-420.

Johnson, R. A., & Wichern, D. W. (1989). *Applied Multivariate Statistical Analysis*. Englewood Cliffs: N.J.: Prentice Hall.

Jones, G. (2001). *E-Commerce and Identity Fraud*. Experian Co. (UK). Available: http://press.experian.com/documents/e-comm.pdf.

Khoumbati, K., Themistocleous, M., & Irani, Z. (2006). Evaluating the Adoption of Enterprise Application Integration in Health-Care Organizations. *Journal of Management Information Systems, 22*(4), 69-108.

Kim, J., & Curry, J. (1977). The Treatment of Missing Data in Multivariate Analysis. *Sociological Methods and Research, 6*, 206-240.

Kim, W., Choi, B. J., Hong, E. K., Kim, S. K., & Lee, D. (2003). A taxonomy of dirty data. *Data Mining and Knowledge Discovery, 7*(1), 81-99.

Kim, W., & Seo, J. (1991). Classifying Schematic and Data heterogeneity in Multidatabase Systems. *IEEE Computer, 24*(12), 12-18.

Knapp, M. L., & Comadena, M. E. (1979). Telling it like it isn't: A review of theory and research on deceptive communication. *Human Communication Research, 5*, 270-285.

Kohnken, G. (1987). Training Police Officers to Detect Deceptive Eyewitness Statements: Does it work? *Social Behavior, 2*, 1-17.

Kraut, R. E., & Poe, D. (1980). On The Line: The Deception Judgements of Customs Inspectors and Laymen. *Journal of Personality and Social Psychology, 39*, 784-798.

Landers, T., & Rosenberg, R. L. (1982). *An Overview of MULTIBASE*. Amsterdam: North-Holland.

Langley, P., Iba, W., & Thompson, K. (1992). *An Analysis of Bayesian Classifiers.* Paper presented at the Proceedings of the Tenth National Conference on Artificial Intelligence.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet Physics Doklady, 10*, 707-710.

Levitin, A., & Redman, T. (1995). Quality Dimensions of a Conceptual View. *Information Processing and Management: an International Journal, 31*(1), 81-88.

Li, C., & Biswas, C. (2002). Unsupervised Learning with Mixed Numeric and Nominal Data. *IEEE Transactions on Knowledge and Data Engineering, 14*(4), 673-690.

Li, C., Chang, E., Garcia-Molina, H., & Wiederhold, G. (2002). Clustering for Approximate Similarity Search in High-Dimensional Spaces. *IEEE Transactions on Knowledge and Data Engineering, 14*(4), 792-808.

Low, W. L., Lee, M. L., & Ling, T. W. (2001). A Knowledge-based Approach for Duplicate Elimination in Data Learning. *Information Systems, 26*, 585-606.

Marshall, B., Kaza, S., Xu, J., Atabakhsh, H., Petersen, T., Violette, C., & Chen, H. (2004). *Cross-Jurisdictional criminal activity networks to support border and transportation security.* Paper presented at the 7th Annual IEEE Conference on Intelligent Transportation Systems (ITSC 2004), Washington, D.C.

Matsumoto, T., Matsumoto, H., Yamada, K., & Hoshino, S. (2002). Impact of Artificial Gummy Fingers on Fingerprint Systems, *SPIE, Optical Security and Counterfeit Deterrence Techniques IV* (Vol. 4677).

McCallum-Bayliss, H. (2004). Identity Resolution in a Global Environment. *IT Professional, 6*(6), 21-26.

Mitchell, R. W. (1966). A Framework for Discussing Deception. In R. W. Mitchell & N. S. Mogdil (Eds.), *Deception: Perspectives on Human and Nonhuman Deceit* (Vol. 3-4). Albany: State University of New York Press.

Monge, A. E. (1997). *Adaptive Detection of Approximately Duplicate Database Records and the Database Integration Approach to Information Discovery.* University of California, San Diego.

Monge, A. E., & Elkan, C. P. (1997). *An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records.* Paper presented at the ACM-SIGMOD Workshop on Research Issues on Knowledge Discovery and Data Mining, Tucson, AZ.

Motro, A. (1987). Superviews - Virtual Integration of Multiple Databases. *Ieee Transactions on Software Engineering, 13*(7), 785-798.

Myrtveit, L., Stensrud, E., & Olsson, U. H. (2001). Analyzing Data Sets with Missing Data: An Empirical Evaluation of Imputation Methods and Likelihood-Based Methods. *IEEE Transactions of Software Engineering, 27*(11).

Navarro, G. (2001). A Guided Tour to Approximate String Matching. *ACM Computing Survey, 33*(1), 31-88.

Newcombe, H. B., Kennedy, J. M., Axford, S. J., & James, A. P. (1959). Automatic Linkage of Vital Records. *Science, 130*(3381), 954-959.

Nigam, K., McCallum, A. K., Thrun, S., & Mitchell, T. (2000). Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning, 39*, 103-134.

Nosofsky, R. M. (1986). Attention, Similarity, and the Identification-Categorization Relationship. *JOurnal of Experimental Psychology: General, 115*, 39-57.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems : networks of plausible inference*. San Mateo, CA: Morgan Kaufmann Publishers.

Porter, E. H., & Winkler, W. E. (1997). Approximate String Comparison and Its Effect on an Advanced Record Linkage System. In W. Alvey & B. Jamerson (Eds.), *Record Linkage Techniques-1997: Proceedings of an International Workshop*

and Exposition (pp. 190-202). Arlington, VA.

Quinlan, J. R. (1986). Induction of Decision Tree. *Machine Learning, 1*, 81-106.

Rahm, E., & Bernstein, P. A. (2001). A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal, 10*(4), 334-350.

Ravikumar, P., & Cohen, W. W. (2004). *A Hierarchical Graphical Model for Record Linkage.* Paper presented at the 20th Conference on Uncertainty in Artificial Intelligence (UAI '04), Banff Park Lodge, Banff, Canada.

Redman, T. C. (1998). The Impact of Poor Data Quality on the Typical Enterprises. *Communications of the ACM, 41*(3), 79-82.

Rosch, E. (1973). Natural Categories. *Cognitive Psychology, 4*, 328-350.

Rosch, E. (1977). *Human Categorization*. London: Academic Press.

Rubin, D. B. (1987). *Multiple Imputation for Nonresponse in Surveys*. New York: Wiley.

Russell, R. C. (1918). Improvements in Indexes, US Patent 1,261,167, *US Patent 1,261,167*.

Salton, G. (1988). *Automatic text processing : the transformation, analysis, and retrieval of information by computer*: Addison-Wesley Pub.

Schafer, J. L. (1997). *Analysis of Incomplete Multivariate Data*. London: Chapman & Hill.

Seifert, J. W. (2004). Data mining and the search for security: Challenges for connecting the dots and databases. *Government Information Quarterly, 21*(4), 461-480.

Shepard, R. N. (1962a). The Analysis of Proximities - Multidimensional-Scaling with an Unknown Distance Function .1. *Psychometrika, 27*(2), 125-140.

Shepard, R. N. (1962b). The Analysis of Proximities - Multidimensional-Scaling with an Unknown Distance Function .2. *Psychometrika, 27*(3), 219-246.

Shepard, R. N. (1963). Analysis of Proximities as a Technique for the Study of Information-Processing in Man. *Human Factors, 5*(1), 33-48.

Snyder, J. M. (2000). Online Auction Fraud: Are the Auction Houses Doing All They Should or Could to Stop Online Fraud? *Federal Communications Law Journal, 52*(2), 453-472.

Spaccapietra, S., Parent, C., & Dupont, Y. (1992). Model Independent Assertions for Integration of Heterogeneous Schemas. *VLDB Journal, 1*(1), 81-126.

Templeton, M., Brill, D., Dao, S. K., Lund, E., Ward, P., Chen, A. L. P., & MacGregor, R. (1987). Mermaid: A Front End to Distributed Heterogeneous Databases. *Proceedings IEEE, 75*(5), 695-708.

Timm, H., & Klawonn, F. (1999). *Different Approaches for Fuzzy Cluster Analysis with Missing Values.* Paper presented at the 7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen.

Titterington, D. M., Murray, G. D., Murray, L. S., Spiegelhalter, D. J., Skene, A. M., Habbema, J. D. F., & Gelpke, G. J. (1981). Comparison of Discrimination Techniques Applied to a Complex Data Set of Head Injured Patients. *Journal of the Royal Statistical Society Series a-Statistics in Society, 144*, 145-175.

Toth, S. (2003). *Need Fuels Demand for False IDs: for Jobs, Documents are the Key*. South Bend Tribune. Available: http://www.southbendtribune.com/stories/2003/07/27/local.20030727-sbt-FULL-A1-Need_fuels_demand_fo.sto.

Trout, J. (1997). *The New Positioning: The Latest on the World's #1 Business Strategy* (1st ed.): McGraw-Hill.

Tversky, A. (1977). Features of Similarity. *Psychological Review, 84*(4), 327-352.

Verykios, V. S., Elmagarmid, A. K., & Houstis, E. N. (2000). Automating the Approximate Record Matching Process. *Information Sciences, 126*(1-4), 83-98.

Vrij, A. (2000). *Detecting Lies and Deceit: The Psychology of Lying and the Implication for Professional Practice*: John Willey & Sons, Ltd.

Wang, G., Chen, H., & Atabakhsh, H. (2004a). Automatically Detecting Deceptive Criminal Identities. *Communications of the ACM, 47*(3), 71-76.

Wang, G., Chen, H., & Atabakhsh, H. (2004b). Criminal identity deception and deception detection in law enforcement. *Group Decision and Negotiation, 13*(2), 111-127.

Wang, G. A., Chen, H., & Atabakhsh, H. (2006). A Multi-layer Naive Bayes Model for Approximate Identity Matching. *Lecture Notes in Computer Science, 3975*, 479-484.

Wang, Y. R., & Madnick, S. (1989). The Interdatabase Instance Identification Problem in Integrating Autonomous Systems, *Fifth International Conference Data*

Engineering.

White, A. P., Liu, W. Z., Hallissey, M. T., & Fielding, J. W. L. (1996). A Comparison of Two Classification Techniques in Screening for Gastro-Esophageal Cancer. *Applications and Innovations in Expert Systems IV*, 83-97.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage, *Survey Research Methods Section, American Statistical Association* (pp. 354-359).

Winkler, W. E. (1998). *Using the EM Algorithm for Weight Computation in the Fellegi-Sunter Model of Record Linkage.* Paper presented at the Section on Survey Research Methods, American Statistical Association.

Winkler, W. E. (1999). *The State of Record Linkage and Current Research Probelms*: Internal Revenue Services Publication R99/04.

Winkler, W. E. (2002). *Methods for Record Linkage and Bayesian Networks.* Paper presented at the Section on Survey Research Methods, American Statistical Association, Alexandria, Virginia.

Wu, S., & Manber, U. (1992). Fast Text Searching Allowing Errors. *Communications of the ACM, 35*(10), 83-91.

Yang, Y., & Webb, G. I. (2002). *A Comparative Study of Discretization Methods for Naive-Bayes Classifiers.* Paper presented at the 2002 Pacific Rim Knowledge Acquisition Workshop, Tokyo, Japan.

Yang, Y., & Webb, G. I. (2003). *On Why Discretization Works for Naive-Bayes Classifiers.* Paper presented at the 16th Australian Joint Conference on Artificial Intelligence, Perth, Australia.

Zhang, H. (2005). Exploring Conditions for the Optimality of Naïve Bayes. *International Journal of Pattern Recognition and Artificial Intelligence, 19*(2), 183-198.

Zhao, H. M., & Soofi, E. S. (2006). Exploring attribute correspondences across heterogeneous databases by mutual information. *Journal of Management Information Systems, 22*(4), 305-336.