The VLDB Journal (2003) 12: 28–40 / Digital Object Identifier (DOI) 10.1007/s00778-002-0072-y

A Bayesian decision model for cost optimal record matching

Vassilios S. Verykios[1], **George V. Moustakides**[2], **Mohamed G. Elfeky**[3]

[1] College of Information Science and Technology, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104-2875, USA
[2] Computer Engineering and Informatics, University of Patras, Greece
[3] Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1398, USA

Edited by F. Lochovsky. Received: February 9, 2001
Published online: November 14, 2002 –

Abstract. In an error-free system with perfectly clean data, the construction of a global view of the data consists of linking – in relational terms, joining – two or more tables on their key fields. Unfortunately, most of the time, these data are neither carefully controlled for quality nor necessarily defined commonly across different data sources. As a result, the creation of such a global data view resorts to approximate joins. In this paper, an optimal solution is proposed for the matching or the linking of database record pairs in the presence of inconsistencies, errors or missing values in the data. Existing models for record matching rely on decision rules that minimize the probability of error, that is the probability that a sample (a measurement vector) is assigned to the wrong class. In practice though, minimizing the probability of error is not the best criterion to design a decision rule because the misclassifications of different samples may have different consequences. In this paper we present a decision model that minimizes the cost of making a decision. In particular: (a) we present a decision rule: (b) we prove that this rule is optimal with respect to the cost of a decision: and (c) we compute the probabilities of the two types of errors (Type I and Type II) that incur when this rule is applied. We also present a closed form decision model for a certain class of record comparison pairs along with an example, and results from comparing the proposed cost-based model to the error-based model, for large record comparison spaces.

Keywords: Record linkage – Data cleaning – Cost optimal statistical model

1 Introduction

In today's competitive business environment, corporations in the private sector are being driven to focus on their customers in order to maintain and expand their market share. This shift is resulting in customer data and information about customers being viewed as a corporate asset. In the public sector, the very large expansion of the role of the government resulted in an unprecedented increase in the demand for detailed information. Only recently has the data analytic value of these administrative records been fully realized. Of primary concern is that, unlike a purposeful data collection effort, the coding of the data is not carefully controlled for quality. Likewise, data objects are not necessarily defined commonly across databases nor in the way data consumers would want. Two of the serious concerns which arise in this context are: (a) how to identify records across different data stores that refer to the same entity; and (b) how to identify duplicate records within the same data store.

If each record in a database or a file carried a unique, universal and error-free identification code, the only problem would be to find an optimal search sequence that would minimize the total number of record comparisons. In most cases, encountered in practice, the identification code of the record is neither unique nor error-free. In some of these cases, the evidence presented by the identification codes, (i.e., primary key, object id, etc.), may possibly point out that the records correspond or that they do not correspond to the same entity. However, in the large majority of practical problems, the evidence may not clearly point to one or the other of these two decisions. Thus, it becomes necessary to make a decision as to whether or not a given pair of records must be treated as though it corresponds to the same real world entity. This is called the record matching or linking problem [13,1,8,18,16,9,10].

In this paper, we consider record matching as a pattern classification task. Classification is one of the primary tasks of data mining [4]. In classification problems, the goal is to correctly assign cases (tests, measurements, observations, etc.) to one of a finite number of classes. Most of the currently available algorithms for classification are designed to minimize zero-one loss or error rate: the number of incorrect predictions made or, equivalently, the probability of making an incorrect prediction. This implicitly assumes that all errors are equally costly, but in most applications, like in record matching, this is rarely the case. For example, in database marketing the cost of mailing to a non-respondent is very small, while the cost of not mailing to someone who would respond is the entire profit loss. In real-world applications, there are many different types of cost involved [26] such as the cost of tests, the cost of the teacher, etc. In this study we consider only the cost of misclas-

sification error which is related to assigning different weights to different misclassification errors. An extension of this work to more general cost models is one of our future goals. In general, misclassification costs may be described by an arbitrary cost matrix C, with elements of the form c_{ij}, meaning the cost of predicting that an example belongs to class i when in fact it belongs to j. It must be emphasized that assembling a cost matrix (populating its cells) is an application specific task and it must be made either by a domain expert or if training data is available, the costs maybe determined automatically. The important thing to note here is that given a cost matrix our decision model generates a space that minimizes the overall cost of the record matching process.

Bayes decision theory is a fundamental statistical approach to the problem of pattern classification. The Bayesian approach is based on the assumption that the decision problem is posed in probabilistic terms, and that all of the relevant probability values are known. In this paper, we propose a constant error cost Bayesian model which means that the cost of a certain type of error (the value in the cost matrix) may be constant (the same value for all cases). In some cases, we are uncertain about the actual costs. To account for this uncertainty, we can use a probability distribution over a range of possible costs. To keep the presentation simple, we do not consider probability distributions over costs in this study.

The large volume of applications spanning the range of cases from (a) an epidemiologist, who wishes to evaluate the effect of a new cancer treatment by matching information from a collection of medical case studies against a death registry in order to obtain information about the cause and the date of death, to (b) an economist, who wishes to evaluate energy policy decisions by matching a database containing fuel and commodity information for a set of companies against a database containing the values and the types of goods produced by the companies, signifies the tremendous impact and applicability of the problem addressed in this paper.

The rest of this paper is organized as follows. Section 2 provides some background information, and the notation that is used throughout this paper. Section 3 introduces the cost optimal model, along with the thresholds of the three decision areas, and the probabilities of errors. Section 4 provides a detailed formulation of the model when the comparison vector components are conditionally independent random variables. An example is also given to illustrate how the model can be applied. Section 5 provides some information about the experimental environment that we used in order to perform the experiments and Sect. 6 presents the results from a sample of the experiments that we run by using it. Finally, Sect. 7 provides concluding remarks and guidelines for future extensions of this work.

2 Background

Record matching or linking is the process of identifying records, in a data store, that refer to the same real world entity or object. There are two types of record matching. The first one is called *exact* or *deterministic* and it is primarily used when there are unique identifiers for each record. The other type of record matching is called *approximate*. In this paper, we focus only on the second type of matching. Table 1 shows pairs of records that have been identified as duplicates in a customer database of a telecommunication company. The decision, as to the matching status of a pair of records, is based on the comparison of common characteristics between the corresponding pair of records. These common characteristics are related to the similarities in the schema of the corresponding records. For example, a customer table may have two different schema representations in two databases of customer data. The first table may store information from the service department while the second one may store information from the billing department. Despite the differences in the representation of the two tables, we always expect that overlapping information (i.e., name, address, sex, marital status, etc.) is present and this information can be used for the identification of matches between records from different databases that refer to the same customer.

Table 1. Duplicated records in a name/address database

Name	*Address*
EAGLE LIFT & EQUIP INC	412 OAK 05
EAGLE LIFT & EQUIPMENT	412 OAK ST
EAGLE LIFT & EQUIP INC	412 OAK 05
EAGLE LIFT & EQUIPMENT	412 OAK STREET
RIVER EAGLE DISTRBG CO	2346 RUST
RIVER EAGLE DISTRIBUTING	2346 RUST AV
RIVER EAGLE DISTRBG CO	2346 RUST
RIVER EAGLE DISTRIBUTING	2346 RUST AVE
EAGLE RIDGE INSTITUTE	601 NE 63RD
EAGLE RIDGE BUSINESS OFFI	601 NE 63RD

The two principal steps in the record matching process are the searching of potentially linkable pairs of records – searching step – and the decision whether or not a given pair is correctly matched – matching step. The aim of the searching step must be the reduction of the number of failures to bring linkable records together for comparison. For the matching step, the problem is how to enable the computer to decide whether or not a pair of records corresponds to the same entity, when one part of the identifying information agrees and another part disagrees. In the remaining of this section we provide information about the notation that we will use, we discuss existing techniques which have been deployed for the record matching process, and we also review decision models which have been built for the matching step.

2.1 Notation

In the product space of two tables, a *match* M is a pair that represents the same entity and a *non-match* U is a pair that represents two different entities. Within a single database, a *duplicate* is a record that represents the same entity as another record in the same database. Common record identifiers such as names, addresses and code numbers (SSN, object identifier), are the matching variables that are used to identify matches. The vector, that keeps the values of all the attribute comparisons for a pair of records (comparison pair) is called *comparison vector* $\underline{x}$. The set of all possible vectors, is called *comparison space* X. A record matching rule is a decision

rule that designates a comparison pair either as a *link* A_1, a *possible link* A_2, or a *non-link* A_3, based on the information contained in the comparison vector. Possible links are those pairs for which identifying information is not sufficient to determine whether a pair is a match, or a non-match. Typically, manual review is required in order to decide upon the matching status of possible links. *False matches* (Type I errors) are those non-matches that are erroneously designated as links by a decision rule. *False non-matches* (Type II errors) are either (a) matches designated as non-links by the decision rule, or (b) matches that are not in the set of pairs to which the decision rule is applied.

For an arbitrary comparison vector $\underline{x} \in X$, we denote by $P(\underline{x} \in X|M)$ or $f_M(\underline{x})$ the frequency of the occurrence or the conditional probability of the pattern $\underline{x}$ among the comparison vectors which are matches. Similarly, we denote by $P(\underline{x} \in X|U)$ or $f_U(\underline{x})$ the conditional probability of pattern $\underline{x}$ among the comparison vectors which are non-matches. Note that the agreement of the comparison vector $\underline{x}$ can be defined as specifically as one wishes and this completely rests to the selection of the components of the comparison vector. We denote by d the predicted class of a pair of records, and by r the actual matching status of a pair of records. Let us also denote by $P(d = A_i, r = j)$ and $P(d = A_i|r = j)$ correspondingly, the joint and the conditional probability that the decision A_i is taken, when the actual matching status (M or U) is j. We also denote by c_{ij} the cost of making a decision A_i when the comparison record corresponds to some pair of records with actual matching status j. When the dependence on the comparison vector is obvious by the context, we eliminate the symbol $\underline{x}$ from the probabilities. Finally we denote the a priori probability of M or else $P(r = M)$ as π_0 and the a priori probability of U or else $P(r = U)$ as $1 - \pi_0$.

2.2 Decision models for record matching

In 1950s, Newcombe et al. [20–22] introduced concepts of record matching that were formalized in the mathematical model of Fellegi and Sunter [5]. Newcombe recognized that record linkage is the statistical problem of deciding which record pair of potential comparisons should be regarded as linked in the presence of errors of identifying information. Fellegi and Sunter formalized this intuitive recognition by defining a linkage rule as a partitioning of the comparison space into the so-called "linked" subset, a second subset for which the inference is that the record pairs refer to different underlying units and a complementary third set where the inference cannot be made without further evidence.

Fellegi and Sunter in [5], made the concepts introduced by Newcombe et al. in [21] rigorous by considering ratios of probabilities of the form:

$$R = P(\underline{x} \in X|M)/P(\underline{x} \in X|U) \quad (1)$$

where $\underline{x}$ is an arbitrary agreement pattern in the comparison space X. The theoretical decision rule is given by:

(a) If $R >$ UPPER, then designate pair as link.
(b) If LOWER $\leq R \leq$ UPPER, then designate the pair as a possible link and hold for clerical review.
(c) If $R <$ LOWER, then designate the pair as non-link.

The UPPER and LOWER cutoff thresholds are determined by a priori error bounds on false matches and false non-matches. Fellegi and Sunter [5] showed that the decision rule is optimal in the sense that for any pair of fixed upper bounds on the rates of false matches and false non-matches, the manual/clerical review region is minimized over all decision rules on the same comparison space X. If now, one considers the costs of the various actions, that might be taken, and the utilities associated with their possible outcomes, it is desirable to choose decision rules that will minimize the costs of the operation. Nathan in [19] proposes a model that involves minimization of a cost function, but restricts detailed discussion to cases in which the information used for matching appears in precisely the same form, whenever the item exists in either input source. Du Bois's [23] approach attempts to maximize the set of correct matches by minimizing the set of erroneous matches. Tepping in [25] provides a graphical representation of a solution methodology that minimizes the mean value of the cost under the condition that the expected value of the loss is a linear function of the conditional probability that the comparison pair is a match. The application of his mathematical model involves the estimation of the cost function for each action, as a function of the probability of a match, and the estimation of the probability, that a comparison pair is a match. The estimation of the cost function is often extremely difficult. Usually the cost consists of two classes of components. The first class consists of the cost of actual operations that may be involved and the second consists of the less tangible losses associated with the occurrence of errors in matching. The former can often be estimated very well, but the estimates of the latter may depend upon judgment in large part. Pinheiro and Sun [24] present a text similarity measure based on dynamic programming for matching verbatim text fields. Based on the similarity measures for each corresponding pair of fields, they build a classification model using logistic regression to predict whether any two records are matched or not. N-grams is another approach for computing the distance between two strings. The N-grams comparison function forms the set of all the sub-strings of length n for each string. String comparisons by using trigrams ($n = 3$) was used by Hylton [10] for the linkage of records of bibliographical data. The N-grams method was extended to what is referred to as Q-grams by Gravano et al. [7] for computing approximate string joins efficiently.

2.3 Intelligent search of the comparison space

Errors, in the form of failures to bring potentially linkable pairs of records together for comparison, could be reduced to zero simply by comparing each record with all the others. However, wherever the files are large, such a procedure would generally be regarded as excessively costly, if there are many wasted

comparisons of pairs of records that are not matched. For this reason, it is usual to order the records in the database by using identifying information that is common to all of them. The ordering can be performed either on the key, or on some other combination of record fields, or even on parts of the fields. In the exact matching, sorting of the file or of the database can be used to reduce the complexity of identifying duplicate records [2]. In the approximate record matching, various compression codes, i.e., phonetic codes, can be used to mask some of the errors that frequently appear in typical record fields such as names. There is a number of systems to do this and the most common of which is known as the Soundex code [22]. The Soundex code is a phonetic coding scheme, which is based on the assignment of code digits which are the same, for any phonetically similar group of components.

Often, we need to make a compromise between the number of record pairs, that are compared, and the completeness of the matching process. The searching process must be intelligent enough and exclude from comparison, record pairs that completely disagree with each other. In order to do that, the searching process must identify only those record pairs which have a high probability of matching (prospective matches) and leave uninspected those pairs that look very different (not prospective matches). Several techniques have been developed in the past for searching the space of record pairs. The first one, was presented early on in a paper by Newcombe [22] and is called, *blocking*. In this approach, the database is scanned by comparing only those records that agree on a user-defined key, which for example can be the key used to sort the records. The characteristics used for blocking purposes are known as *blocking variables*. Kelley in [12] presents results related to a method for determining the best blocking strategy.

Another technique for cutting down the number of unwanted comparisons in the approximate record matching, is to scan the database by using a fixed size window and check for matches by comparing every pair of records that falls inside that window, assuming that the records are already sorted. This approach is known as the *sorted-neighborhood* approach and has been proposed by Hernadez and Stolfo in [9]. Because of the various types of errors that exist in the data sets that are compared, it is very common that the information selected for blocking or sorting the data sets contains errors. If that happens, we expect that some records to be clustered far away from those records with which they should be compared to. In this case, a *multi-pass approach*, proposed in [9], can be used. In this approach, a number of different blocking variables, or sorting keys, can be used for clustering the records in different ways. The database is then scanned as many times as the number of the different keys. The results from independent passes are combined to give the final set of matching records. An extension of the multi-pass approach has also been implemented by the same group of researchers. On top of the multi-pass approach, the transitive closure of the results of independent passes is computed. A similar approach, that has been proposed independently by Monge et al. in [17], makes use of an algorithmic technique that identifies the connected components of a graph. By considering each record cluster as a connected component, this process can be effectively used to select the records that belong to the same cluster. Both groups of researchers presented very similar results, regarding the accuracy and the cost of the searching process.

Most recently, record matching has been investigated in the data cleaning context. Lee et al. [14] extend the equational theory for record matching to a complete knowledge-based framework for data cleaning. Galhardas et al. [6] propose a declarative language for the logical specification of data cleaning operations, along with a framework for specifying various data cleaning techniques at the logical and physical database level; record matching is one of these techniques. Finally, in [3], Elfeky et al. demonstrate a Record Linkage Toolbox that can make use of a variety of statistical and machine learning techniques for solving the record matching problem.

3 The cost optimal Bayesian decision model

In the course of the record matching process, we observe a comparison vector $\underline{x}$ and we want to decide upon whether the comparison record corresponds to a matched pair M of source records (A_1); or whether the comparison record corresponds to an unmatched pair U of source records (A_3). In a Bayesian decision setting, the decision rule, may be written as follows: if $p(M|\underline{x}) \geq p(U|\underline{x})$ then it is decided that $\underline{x}$ belongs to M, otherwise it is decided that $\underline{x}$ belongs to U. This decision rule indicates that if the probability of the match class M, given the comparison vector $\underline{x}$, is larger than the probability of the non-match class U, $\underline{x}$ is classified to M, and vice versa. For example, if the binary comparison vector value $(1, 1, 0)$ has a probability of appearing 75% among matches and only 25% among non-matches, then the rule of minimum error assigns the comparison vector to the set M. In addition, if we consider the misclassification costs, and it so happens that misclassifying an unmatched comparison vector to a linked one is at least three times more severe than misclassifying a matched comparison vector to an unlinked one, then the rule of minimum cost will assign the vector $(1, 1, 0)$ to the set U instead. Based on this example we can see how the cost of different decisions affects the results produced by the error-based and cost-based models.

However, in some cases we will find ourselves unable to make either of these decisions at specified levels of error or cost so we will allow a third decision (A_2). In order to be able to make a decision, we assume that the distributions of the random comparison vectors are known. Determining the distributions of the random vectors requires a pre-processing phase of training. For the training phase, a set of classified random vectors is required, which can be used for determining these a priori matching probabilities. Manual inspection of pairs of records is required for labeling their corresponding comparison vectors with the correct class label. Other approaches can also be applied on this process. For example, a domain expert may assign a priori probabilities of M and U for each one of the comparison vector fields. Another approach is to use a clustering algorithm (i.e., EM) to automatically determine these a priori probabilities. The important thing to note here is that for this model to work, we need to know: (a) the a priori matching probabilities of the random comparison vectors; and (b) the various costs that should be assigned to different classifications/misclassifications. The model that we are building will determine the necessary and sufficient criterion for testing the M hypothesis against the U hypothesis and vice versa, and the thresholds required for this reason.

Table 2. Costs of various decisions

Cost	Decision	Actual Matching Status
c_{10}	A_1	M
c_{11}	A_1	U
c_{20}	A_2	M
c_{21}	A_2	U
c_{30}	A_3	M
c_{31}	A_3	U

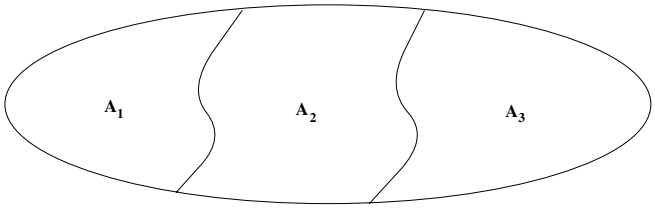


Fig. 1. A partitioning of the decision space

Let us denote by c_{ij} the cost of making a decision A_i when the comparison record corresponds to some pair of records with actual matching status j. Each one of the decisions that are made, based on the existing evidence about the linking status of a comparison pair, is associated with a certain cost that has two aspects. The first aspect is related to the decision process itself and is associated with the cost of making a particular decision; for example, how many value comparisons are needed in order to decide, affects the cost of this decision. The second aspect is associated with the cost of the impact of a certain decision; for example, making a wrong decision should always cost more than making the correct decision. Table 2 illustrates the costs for all the various decisions that could be made during the record matching process.

We need to minimize the mean cost $\bar{c}$ that results from making a decision. The mean cost is written as follows:

$$\begin{aligned}\bar{c} = {} & c_{10} \cdot P(d = A_1, r = M) + c_{11} \cdot P(d = A_1, r = U) \\ & + c_{20} \cdot P(d = A_2, r = M) + c_{21} \cdot P(d = A_2, r = U) \\ & + c_{30} \cdot P(d = A_3, r = M) + c_{31} \cdot P(d = A_3, r = U).\end{aligned} \quad (2)$$

From the Bayes theorem, the following is true:

$$\begin{aligned}& P(d = A_i, r = j) = P(d = A_i | r = j) \cdot P(r = j), \\ \text{where} \quad & i = 1, 2, 3 \text{ and } j = M, U.\end{aligned} \quad (3)$$

Let us also assume that $\underline{x}$ is a comparison vector drawn randomly from the space of the comparison vectors which is shown in Fig. 1. Then the following equality holds for the conditional probability $P(d = A_i | r = j)$:

$$\begin{aligned}& P(d = A_i | r = j) = \sum_{\underline{x} \in A_i} f_j(\underline{x}), \\ \text{where} \quad & i = 1, 2, 3 \text{ and } j = M, U.\end{aligned} \quad (4)$$

where f_j is the probability density of the comparison vectors when the actual matching status is j.

We also denote the a priori probability of M or else $P(r = M)$ by π_0 and the a priori probability of U or else $P(r = U)$ as $1 - \pi_0$.

The mean cost $\bar{c}$ in Eq. 2 based on Eq. 3 is written as follows:

$$\begin{aligned}\bar{c} = {} & c_{10} \cdot P(d = A_1 | r = M) \cdot P(r = M) + c_{11} \\ & \cdot P(d = A_1 | r = U) \cdot P(r = U) \\ & + c_{20} \cdot P(d = A_2 | r = M) \cdot P(r = M) + c_{21} \\ & \cdot P(d = A_2 | r = U) \cdot P(r = U) \\ & + c_{30} \cdot P(d = A_3 | r = M) \cdot P(r = M) + c_{31} \\ & \cdot P(d = A_3 | r = U) \cdot P(r = U).\end{aligned} \quad (5)$$

By using Eq. 4, Eq. 5 becomes:

$$\begin{aligned}\bar{c} = {} & c_{10} \cdot \sum_{\underline{x} \in A_1} f_M(\underline{x}) \cdot P(r = M) + c_{11} \\ & \cdot \sum_{\underline{x} \in A_1} f_U(\underline{x}) \cdot P(r = U) \\ & + c_{20} \cdot \sum_{\underline{x} \in A_2} f_M(\underline{x}) \cdot P(r = M) + c_{21} \\ & \cdot \sum_{\underline{x} \in A_2} f_U(\underline{x}) \cdot P(r = U) \\ & + c_{30} \cdot \sum_{\underline{x} \in A_3} f_M(\underline{x}) \cdot P(r = M) + c_{31} \\ & \cdot \sum_{\underline{x} \in A_3} f_U(\underline{x}) \cdot P(r = U).\end{aligned} \quad (6)$$

By substituting the a priori probabilities of M and U in Eq. 6, we get the following equation:

$$\begin{aligned}\bar{c} = {} & c_{10} \cdot \pi_0 \cdot \sum_{\underline{x} \in A_1} f_M(\underline{x}) + c_{11} \cdot (1 - \pi_0) \cdot \sum_{\underline{x} \in A_1} f_U(\underline{x}) \\ & + c_{20} \cdot \pi_0 \cdot \sum_{\underline{x} \in A_2} f_M(\underline{x}) + c_{21} \cdot (1 - \pi_0) \cdot \sum_{\underline{x} \in A_2} f_U(\underline{x}) \\ & + c_{30} \cdot \pi_0 \cdot \sum_{\underline{x} \in A_3} f_M(\underline{x}) + c_{31} \cdot (1 - \pi_0) \cdot \sum_{\underline{x} \in A_3} f_U(\underline{x}).\end{aligned} \quad (7)$$

which by dropping the dependent vector variable $\underline{x}$, and combining the information for each part of the decision space, can be rewritten as follows:

$$\begin{aligned}\bar{c} = {} & \sum_{\underline{x} \in A_1} [f_M \cdot c_{10} \cdot \pi_0 + f_U \cdot c_{11} \cdot (1 - \pi_0)] \\ & + \sum_{\underline{x} \in A_2} [f_M \cdot c_{20} \cdot \pi_0 + f_U \cdot c_{21} \cdot (1 - \pi_0)] \\ & + \sum_{\underline{x} \in A_3} [f_M \cdot c_{30} \cdot \pi_0 + f_U \cdot c_{31} \cdot (1 - \pi_0)].\end{aligned} \quad (8)$$

Every point $\underline{x}$ in the decision space A, belongs either in partition A_1, or in A_2 or in A_3 and it contributes additively in the mean cost $\bar{c}$. We can thus assign each point independently either to A_1, or A_2 or A_3 in such a way that its contribution to the mean cost is minimum. This will lead to the optimum selection for the three sets which we denote by A_1^o, A_2^o, and A_3^o. Based on this observation, a point $\underline{x}$ is assigned to the three optimal areas as follows:

To A_1^o if:

$$\begin{aligned} & f_M \cdot c_{10} \cdot \pi_0 + f_U \cdot c_{11} \cdot (1-\pi_0) \\ & \quad \leq f_M \cdot c_{30} \cdot \pi_0 + f_U \cdot c_{31} \cdot (1-\pi_0) \\ \text{and,} \quad & f_M \cdot c_{10} \cdot \pi_0 + f_U \cdot c_{11} \cdot (1-\pi_0) \\ & \quad \leq f_M \cdot c_{20} \cdot \pi_0 + f_U \cdot c_{21} \cdot (1-\pi_0). \end{aligned}$$

To A_2^o if:

$$\begin{aligned} & f_M \cdot c_{20} \cdot \pi_0 + f_U \cdot c_{21} \cdot (1-\pi_0) \\ & \quad \leq f_M \cdot c_{30} \cdot \pi_0 + f_U \cdot c_{31} \cdot (1-\pi_0) \\ \text{and,} \quad & f_M \cdot c_{20} \cdot \pi_0 + f_U \cdot c_{21} \cdot (1-\pi_0) \\ & \quad \leq f_M \cdot c_{10} \cdot \pi_0 + f_U \cdot c_{11} \cdot (1-\pi_0). \end{aligned}$$

And to A_3^o if:

$$\begin{aligned} & f_M \cdot c_{30} \cdot \pi_0 + f_U \cdot c_{31} \cdot (1-\pi_0) \\ & \quad \leq f_M \cdot c_{10} \cdot \pi_0 + f_U \cdot c_{11} \cdot (1-\pi_0) \\ \text{and,} \quad & f_M \cdot c_{30} \cdot \pi_0 + f_U \cdot c_{31} \cdot (1-\pi_0) \\ & \quad \leq f_M \cdot c_{20} \cdot \pi_0 + f_U \cdot c_{21} \cdot (1-\pi_0). \end{aligned}$$

We thus conclude from the above that:

$$A_1^o = \left\{ \underline{x} : \frac{f_U}{f_M} \leq \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{30}-c_{10}}{c_{11}-c_{31}} \text{ and, } \frac{f_U}{f_M} \leq \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{20}-c_{10}}{c_{11}-c_{21}} \right\} \quad (9)$$

$$A_2^o = \left\{ \underline{x} : \frac{f_U}{f_M} \geq \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{20}-c_{10}}{c_{11}-c_{21}} \text{ and, } \frac{f_U}{f_M} \leq \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{30}-c_{20}}{c_{21}-c_{31}} \right\} \quad (10)$$

$$A_3^o = \left\{ \underline{x} : \frac{f_U}{f_M} \geq \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{30}-c_{10}}{c_{11}-c_{31}} \text{ and, } \frac{f_U}{f_M} \geq \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{30}-c_{20}}{c_{21}-c_{31}} \right\} \quad (11)$$

The inequalities above give rise to three different threshold values in the decision space. We denote these thresholds as:

$$\kappa = \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{30}-c_{10}}{c_{11}-c_{31}}, \quad (12)$$

$$\lambda = \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{20}-c_{10}}{c_{11}-c_{21}}, \quad (13)$$

$$\mu = \frac{\pi_0}{1-\pi_0} \cdot \frac{c_{30}-c_{20}}{c_{21}-c_{31}}. \quad (14)$$

In the discussion below, we investigate what kind of relationships hold among these values, in order to concretely define the decision regions. For the sake of simplicity in the presentation, we have eliminated the common factor $\pi_0/(1-\pi_0)$ from all of the threshold values in the proofs below. This can be easily done by a simple transformation of variables. Now, we first observe that in order for the intermediate decision area A_2^o to exist, the following relationship should hold for the pair of thresholds λ and μ:

$$\lambda = \frac{c_{20}-c_{10}}{c_{11}-c_{21}} \leq \frac{c_{30}-c_{20}}{c_{21}-c_{31}} = \mu \quad (15)$$

Based on Eq. 15, we can easily prove that the threshold value $\kappa = (c_{30}-c_{10})/(c_{11}-c_{31})$ lies between λ and μ. Indeed:

$$\lambda = \frac{c_{20}-c_{10}}{c_{11}-c_{21}} \Rightarrow \lambda \cdot (c_{11}-c_{21}) = c_{20}-c_{10} \quad (16)$$

and,

$$\lambda \leq \frac{c_{30}-c_{20}}{c_{21}-c_{31}} \Rightarrow \lambda \cdot (c_{21}-c_{31}) \leq c_{30}-c_{20} \quad (17)$$

By adding by parts Eq. 16 and Eq. 17, we have:

$$\begin{aligned} & \lambda \cdot (c_{11}-c_{21}) + \lambda \cdot (c_{21}-c_{31}) \\ & \quad \leq (c_{20}-c_{10}) + (c_{30}-c_{20}) \Rightarrow \\ & \quad \lambda \cdot (c_{11}-c_{31}) \leq c_{30}-c_{10} \Rightarrow \\ & \quad \lambda \leq \frac{c_{30}-c_{10}}{c_{11}-c_{31}} \Rightarrow \\ & \quad \lambda \leq \kappa. \end{aligned}$$

By following the same reasoning, we can easily show that $\kappa \leq \mu$. Thus, this proves our argument that

$$\lambda = \frac{c_{20}-c_{10}}{c_{11}-c_{21}} \leq \kappa = \frac{c_{30}-c_{10}}{c_{11}-c_{31}} \leq \frac{c_{30}-c_{20}}{c_{21}-c_{31}} = \mu \quad (18)$$

If $\lambda > \mu$, then the area A_2^o disappears. If this holds, then by using the same derivation sequence as above, we can show that $\mu < \kappa < \lambda$. This inequality implies that there are only two regions in the decision area which are separated by the threshold κ. Such a case occurs because of the fact that the assigned costs to the decision area A_2^o, turn out to be high when they are compared to the other two alternative decisions A_1^o and A_3^o. Therefore, the test tends to completely avoid decision A_2^o. The necessary and sufficient condition that guarantees existence of A_2^o is Eq. 15 ($\lambda \leq \mu$).

3.1 Optimality of the decision model

We can now prove that the decision model that we proposed (i.e., the sets A_1^o, A_2^o, and A_3^o) is an optimal one. Based on the discussion above we know that $A = A_1 \bigcup A_2 \bigcup A_3$, where A_1, A_2 and A_3 are pair-wise disjoint. Every point also, belongs to either one of these decision areas. We also introduce the indicator function I_C of a set C, as the function which takes the value of 1 if the point $\underline{x}$ belongs to C and the value 0, otherwise. Note that we can formally write Eq. 8 as:

$$\overline{c} = \sum_{\underline{x} \in A_1} z_1(\underline{x}) + \sum_{\underline{x} \in A_2} z_2(\underline{x}) + \sum_{\underline{x} \in A_3} z_3(\underline{x}) \quad (19)$$

where $z_i(x)$, $i = 1, 2, 3$ denote the expressions inside the corresponding sums in Eq. 8.

Using the indicator functions, we can write:

$$\overline{c} = \sum_{\underline{x} \in A_1} z_1(\underline{x}) + \sum_{\underline{x} \in A_2} z_2(\underline{x}) + \sum_{\underline{x} \in A_3} z_3(\underline{x}) \quad (20)$$

$$= \sum_{\underline{x} \in A} [z_1(\underline{x}) \cdot I_{A_1}(\underline{x}) + z_2(\underline{x}) \cdot I_{A_2}(\underline{x}) + z_3(\underline{x}) \cdot I_{A_3}(\underline{x})] \quad (21)$$

$$\geq \sum_{\underline{x} \in A} \min\{z_1(\underline{x}), z_2(\underline{x}), z_3(\underline{x})\} \quad (22)$$

$$\stackrel{\text{def}}{=} \sum_{\underline{x} \in A_1^o} z_1(\underline{x}) + \sum_{\underline{x} \in A_2^o} z_2(\underline{x}) + \sum_{\underline{x} \in A_3^o} z_3(\underline{x}) \quad (23)$$

3.2 Error estimation

The probability of errors can now be easily computed. There are two types of errors. The first one is called Type I error and it occurs when a *non-link* action is taken although the two records are actually matched. The probability of this error can be estimated as follows:

$$P(d = A_3, r = M) = P(d = A_3 | r = M) \cdot P(r = M) = \pi_0 \cdot \sum_{\underline{x} \in A_3} f_M(\underline{x}). \quad (24)$$

The second type of error is called Type II error and it occurs when the *link* action is taken although the pair of records is actually non-matched. The probability of this error can be estimated as follows:

$$P(d = A_1, r = U) = P(d = A_1 | r = U) \cdot P(r = U) = (1 - \pi_0) \cdot \sum_{\underline{x} \in A_1} f_U(\underline{x}). \quad (25)$$

4 Conditionally independent binary components

The exact record linkage decision rule depends on the probability distributions assumed. In this section, we treat one interesting distribution, and we provide closed form formulas for the weights. We also provide an example.

4.1 Case study

Suppose that the vector $\underline{x}$ is a random variable having binary (0 or 1) components. Further, suppose that the components of these vectors are conditionally independent given the actual value of the matching status. By conditional independence in this case, we mean that the formulas for the distribution can be expanded as follows:

$$f_j(\underline{x}) = f_j^1(x_1) \cdot f_j^2(x_2) \cdots f_j^n(x_n), \text{ where } j = M, U. \quad (26)$$

Let us define values of the components of the distribution for specific values of their arguments, x_i:

$$f_M^i(x_i = 1) = p_i \quad (27)$$

$$f_M^i(x_i = 0) = 1 - p_i \quad (28)$$

$$f_U^i(x_i = 1) = q_i \quad (29)$$

$$f_U^i(x_i = 0) = 1 - q_i \quad (30)$$

Table 3. Probabilities of agreement and disagreement

Attribute	*Under M*		*Under U*	
	p_i	$1 - p_i$	q_i	$1 - q_i$
Last Name	0.90	0.10	0.05	0.95
First Name	0.85	0.15	0.10	0.90
Sex	0.95	0.05	0.45	0.55

We also consider the logarithm of the likelihood ratio $\frac{f_U}{f_M}$.

$$\log \frac{f_U}{f_M} = \log \frac{f_U^1(x_1) \cdot f_U^2(x_2) \cdots f_U^n(x_n)}{f_M^1(x_1) \cdot f_M^2(x_2) \cdots f_M^n(x_n)} \quad (31)$$

which can also be written as follows:

$$\log \frac{f_U}{f_M} = \log \frac{f_U^1(x_1)}{f_M^1(x_1)} + \log \frac{f_U^2(x_2)}{f_M^2(x_2)} + \cdots + \log \frac{f_U^n(x_n)}{f_M^n(x_n)} = \sum_{i=1}^{n} \log \frac{f_U^i(x_i)}{f_M^i(x_i)} \quad (32)$$

Now, we note that since x_i can only assume the values of 1 or 0:

$$\log \frac{f_U(x_i)}{f_M(x_i)} = x_i \cdot \log \frac{q_i}{p_i} + (1 - x_i) \cdot \log \frac{1 - q_i}{1 - p_i} = x_i \cdot \log \frac{q_i(1 - p_i)}{p_i(1 - q_i)} + \log \frac{1 - q_i}{1 - p_i} \quad (33)$$

Based on Eq. 33, Eq. 32 can be written as follows:

$$\log \frac{f_U}{f_M} = \sum_{i=1}^{n} x_i \cdot \log \frac{q_i(1 - p_i)}{p_i(1 - q_i)} + \sum_{i=1}^{n} \log \frac{1 - q_i}{1 - p_i} \quad (34)$$

4.2 Example

We assume that two records are being compared and that a decision will be made as to their matching status based on a comparison of three attributes: last name, first name and sex. For each attribute there will be two possible outcomes: either that they agree or they do not agree. Thus, the comparison space contains eight 3-component vectors. For simplicity we also assume that the probabilities of agreement or disagreement of the attributes are independent under both M and U. Table 3 gives the probabilities of agreement and disagreement under both M and U.

Let us denote by a_1 the comparison of last name, by a_2 the comparison of the first name, and by a_3 the comparison of the sex attribute. Let us also denote a comparison vector $\underline{x} = (x_1, x_2, x_3)$, where $x_i = 1$ if attribute i agrees, and $x_i = 0$ is attribute i disagrees. By using Eq. 34, we can compute the logarithm of the likelihood ratio of variable $\underline{x}_i$. In Table 4 we show the likelihood ratios for this variable.

In order to decide upon the assignment of the comparison vectors to decision regions, we need to assign values to the various costs. Based on the impact of our decisions, we have made the following value assignments: $c_{10} = 0$, $c_{20} = 0.2$,

Table 4. Likelihood ratios for the comparison vectors and their assignment to decision areas

i	$\underline{x}_i$	$\log(\frac{f_U}{f_M})$	Decision Area
1	$(0,0,0)$	2.795	A_3
2	$(0,0,1)$	1.429	A_3
3	$(0,1,0)$	1.088	A_3
4	$(1,0,0)$	0.562	A_2
5	$(0,1,1)$	-0.272	A_2
6	$(1,0,1)$	-0.804	A_1
7	$(1,1,0)$	-1.145	A_1
8	$(1,1,1)$	-2.511	A_1

$c_{30} = 1$, $c_{11} = 1$, $c_{21} = 0.2$, and finally $c_{31} = 0$. We also assume that the a priori probability that a certain vector belongs to M is equal to the a priori probability that the same vector belongs to U. For this reason, the ratio $\pi_0/(1-\pi_0)$ is equal to 1. By using Eqs. 12, 13, and 14, we compute the values for $\kappa = 1$, $\lambda = 0.25$ and $\mu = 4$. We observe that the values for the thresholds satisfy the necessary and sufficient condition in Eq. 15. In order to be consistent with the case study we also need to take the logarithms of the threshold values. By doing this we obtain: $\log(\kappa) = 0$, $\log(\lambda) = -0.602$ and $\log(\mu) = 0.602$. The values for the two thresholds are equal in absolute values, because we selected the costs in such a way that λ to be the inverse of μ. Based on the values for the thresholds, we can assign the comparison pairs to one of the three decision areas. The assignment is shown in Table 4.

5 Prototype experimental system

In order to validate and evaluate the proposed decision model, we build an experimental system. The system relies on a database generator [8] that automatically generates source data, with a priori known characteristics. This system also allows us to perform controlled studies so as to establish the accuracy or else the overall error, and the percentage of comparison pairs which are assigned to the decision area A_2, in which further manual inspection is needed in order to identify the matching status. The database generator provides a large number of parameters including the size of the database, the percentage of duplicate records in the database, and the percentage of the error to be introduced in the duplicated records. Each one of the generated records, by the database generator, consists of the following fields: (a) social security number; (b) first name; (c) middle initial; (d) last name; (e) street number; (f) street address; (g) apartment number; (h) city; (i) state; and (j) zip code. Some of the fields can be empty as well. As reported in [8] the names were chosen randomly from a list of $63,000$ real names. The cities, the states, and the zip codes (all from the USA) come from publicly available lists.

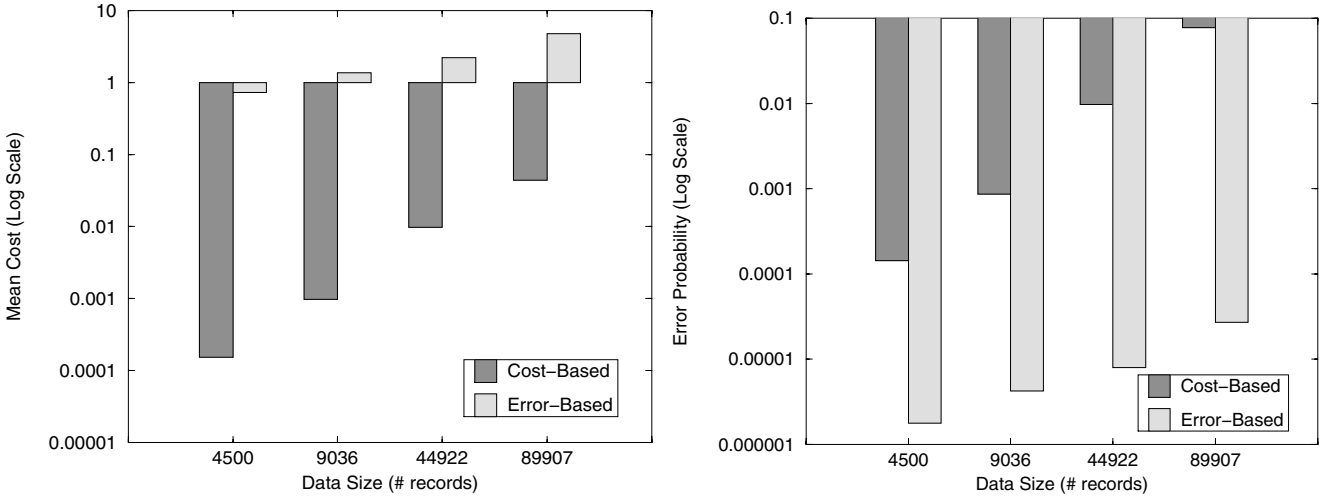
Our system generates two different databases for each set of experiments. The first database is used for training the decision model and the second for testing it. The training process includes the determination of the required parameters by the decision model. Both databases are generated by using almost the same parameter settings. It is only the number of records and the number of record clusters in each database that can be different. A record cluster is a group of records in the same database that refer to the same person. All the records in the same cluster are considered as duplicates. In practice, the size of the training set must be very small compared to the size of the test set in which the model is applied to, in order for the training phase to be efficient. In real life, the test set is the actual set of records which must be matched or unduplicated. According to our results, the ratio between the size of the training and the test database is more than 10.

The training and the test databases are used correspondingly for generating the training comparison space and the test comparison space. As we mentioned earlier on, the comparison space is populated by comparison vectors which correspond to a component by component comparison of a pair of database records. In our system, we can explicitly select the type of the comparison, to be performed between each pair of values corresponding to the same attribute, and the type of the comparison result. In this study, the comparison vector has binary components and for this reason the comparison result can either be 0 or 1. In order to convert to binary, non-binary components we used an entropy-based discretization method. The parameters to be used, for the generation of the training and the test comparison space, can be selected to be different in our system.

Some of the options that are provided to the users of the experimental system, for the generation of the training and test comparison spaces, include: (a) the pre-conditioning of the database records; (b) the selection of the sorting keys to be used for sorting the original database records; (c) the functions to be used for the comparison of each record attribute; (d) the searching strategy along with its parameters if applicable; and (e) the thresholds for the decision model. For the pre-conditioning of the database records, we may select to convert all the characters to uppercase or lowercase, and compute the Soundex code of the last name. Any subset or part of the record fields can be used as a sorting key. Among the functions to be selected for comparing pairs of field values, the most frequently used are the Hamming distance for numerical attributes, and the edit distance [15], the n-grams [10], the Jaro distance [11], and the Smith-Waterman algorithm [18] for character string attributes. For the searching strategy, the experimental system currently supports the blocking and the sorted-neighborhood approach. In the sorted-neighborhood approach the window size to be used should also be provided as an input parameter to the system. The last part of the parameters that are required by the system include the threshold values, which delimit the three decision areas in the proposed model. These thresholds can be computed in a straightforward manner by using Eq. 12, Eq. 13 and Eq. 14, provided that the user has selected the corresponding values for the costs shown in the Table 2 and the a priori probabilities of the M and U.

6 Experiments and results

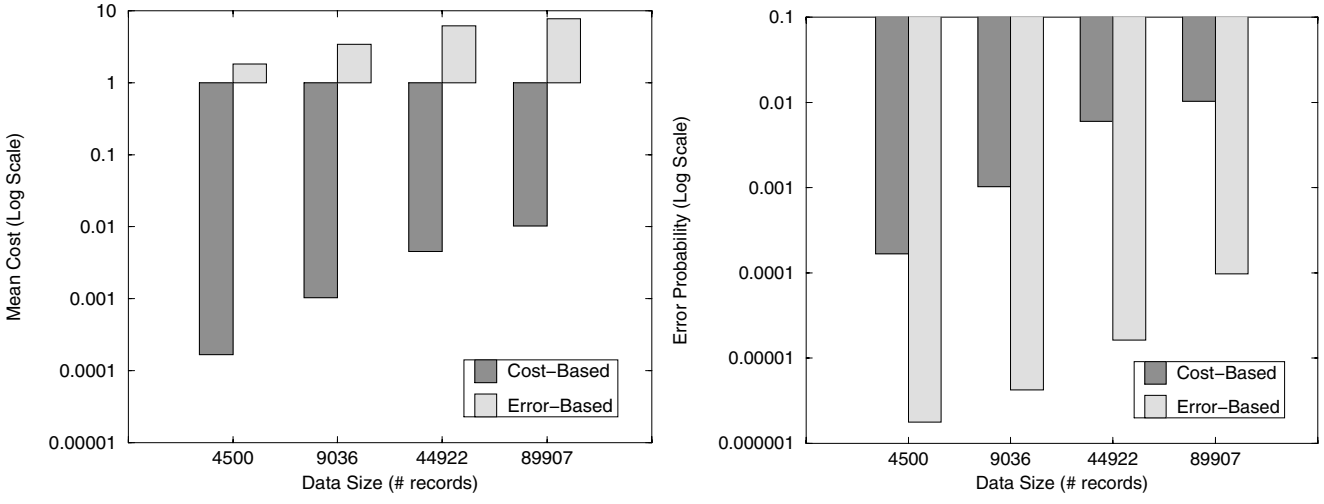
Our main goal in the experimentation with the prototype system was to compare the efficiency of the proposed cost-based approach with the error-based one [5]. In order to compare these two models, we use two metrics: the mean cost given by Eq. 8, and the error probability given by Eq. 24 (Type-I error) and Eq. 25 (Type-II error).



(a) Mean cost

(b) Error probability

Fig. 2. $c_{20} = 0.3, c_{21} = 0.2$



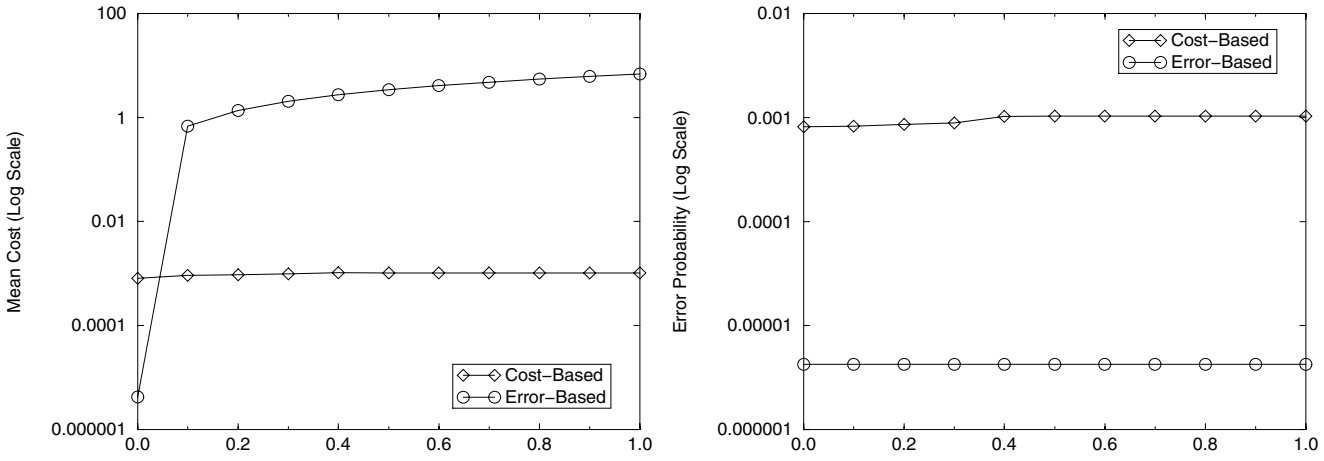
(a) Mean cost

(b) Error probability

Fig. 3. $c_{20} = c_{21} = 0.5$

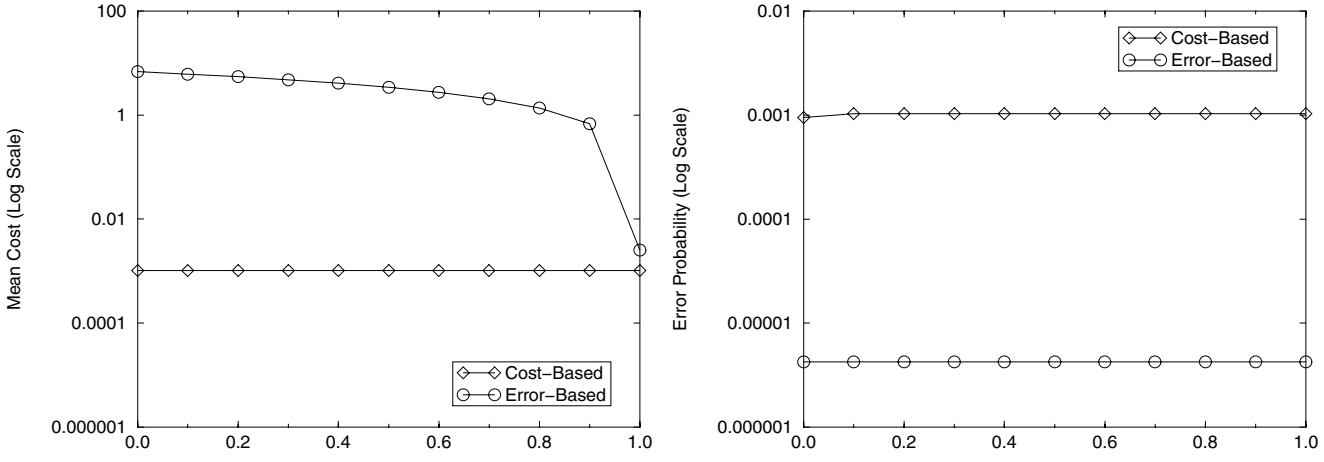
As a pre-processing step, we converted all the characters in both databases to uppercase letters and we computed the Soundex code of the last name, which was used for blocking. In order to present comparable results in our experiments, we have normalized the range of values for the costs, in such a way that $c_{10} = c_{31} = 0$, and $c_{11} = c_{30} = 1$. This means that the correct decision has zero cost, and the wrong decision has maximum cost. Notice that we will use the value 1 as the maximum cost. Hence, the values assigned to the other costs: c_{20}, and c_{21} will be less than or equal to 1. We will derive the necessary condition between the costs c_{20} and c_{21} so as for the decision area A_2 to exist. By substituting the values of $c_{10} = 0, c_{30} = 1, c_{11} = 1$, and $c_{31} = 0$ into the Eq. 15, we get that the sum of the costs should be less or equal to 1, or else $c_{20} + c_{21} \leq 1$. When the equality holds (i.e., $c_{20} + c_{21} = 1$) then all of the three thresholds (κ, λ and μ) coincide, and their value depends only on the value of the π_0.

The probabilities of agreement and disagreement under both the match and the non-match hypothesis are given in Table 5. These values can be easily computed by using the information in the training comparison space since the actual matching status is considered known. This is possible, because each database record has been assigned a cluster identifier by the database generator, which is used for the identification of the cluster that each record belongs to. Based on these probabilities, and the selection of the values for the costs in the A_2 area, c_{20}, and c_{21}, we can compute the thresholds in the decision model, along with the percentage of error and the



(a) Mean cost (b) Error probability

Fig. 4. $c_{20} = c_{21} = 0, 0.1, ..., 1$



(a) Mean cost (b) Error probability

Fig. 5. $c_{20} + c_{21} = 1, c_{20} = 0, 0.1, ..., 1$

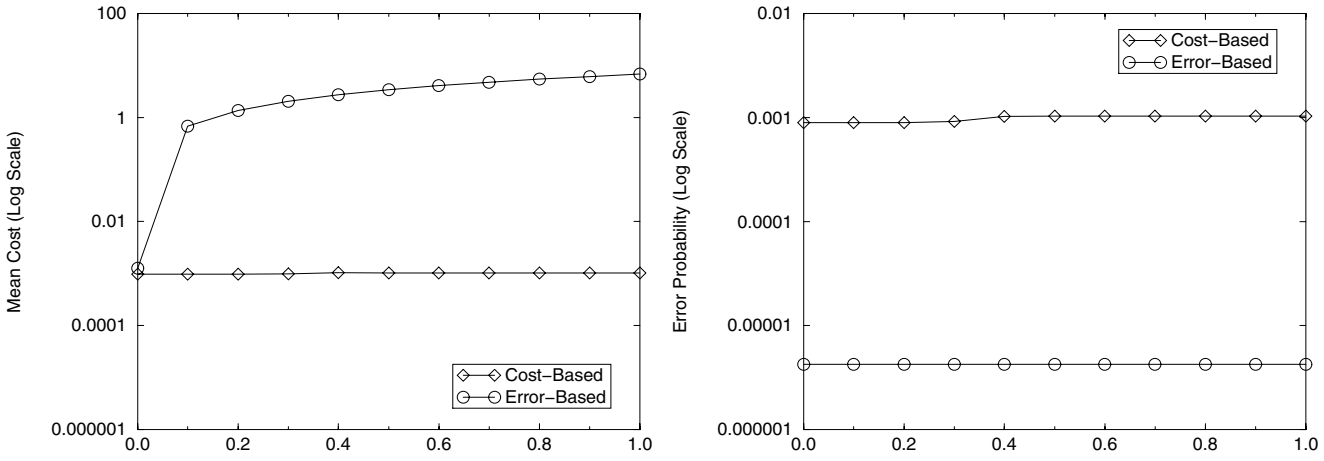
percentage of comparison vectors in the testing set which was assigned in the area A_2.

In all the experiments performed, we use blocking as the searching method and the Soundex code of the last name as the blocking key. The first and third row in the cost matrix are fixed, i.e., $c_{10} = c_{31} = 0$, and $c_{11} = c_{30} = 1$. This means that the correct decision has zero cost, and the wrong decision has maximum cost. Notice that we will use the value 1 as the maximum cost. Hence, the values assigned to the other costs: c_{20}, and c_{21} will be less than or equal to 1.

The first set of experiments examines different sizes of data sets with different assignments of costs. Figures 2 and 3 show the results using $c_{20} = 0.3$, $c_{21} = 0.2$, and $c_{20} = c_{21} = 0.5$, respectively. Both Figs. 2(a) and 3(a) show that the cost-based model has a lower mean cost value for the different data sizes. However, Figs. 2(b) and 3(b) show that the error-based model

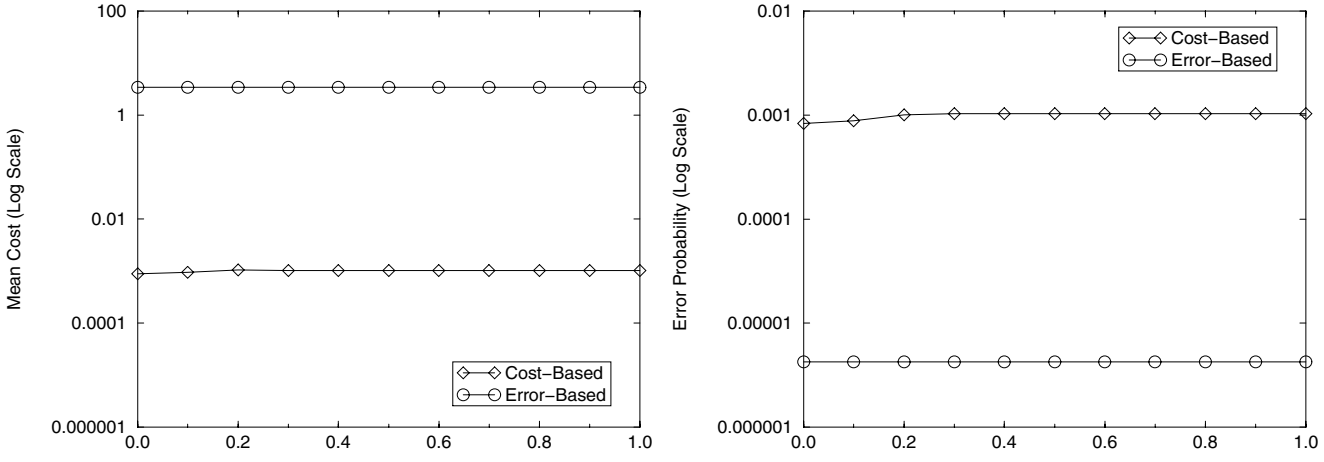
Table 5. Probabilities of agreement and disagreement in the training comparison space

Attribute	*Under M*		*Under U*	
	p_i	$1 - p_i$	q_i	$1 - q_i$
SSN	0.749	0.251	0.025	0.975
First Name	0.837	0.163	0.004	0.996
Middle Initial	0.983	0.017	0.017	0.983
Last Name	0.949	0.051	0.034	0.966
Street Number	0.619	0.381	0.013	0.987
Street Address	0.654	0.346	0.004	0.996
Apartment Number	0.869	0.131	0.004	0.996
City	0.750	0.250	0.004	0.996
State	0.954	0.046	0.004	0.996
Zip Code	0.936	0.064	0.004	0.996



(a) Mean cost (b) Error probability

Fig. 6. $c_{20} = 0.5, c_{21} = 0, 0.1, ..., 1$



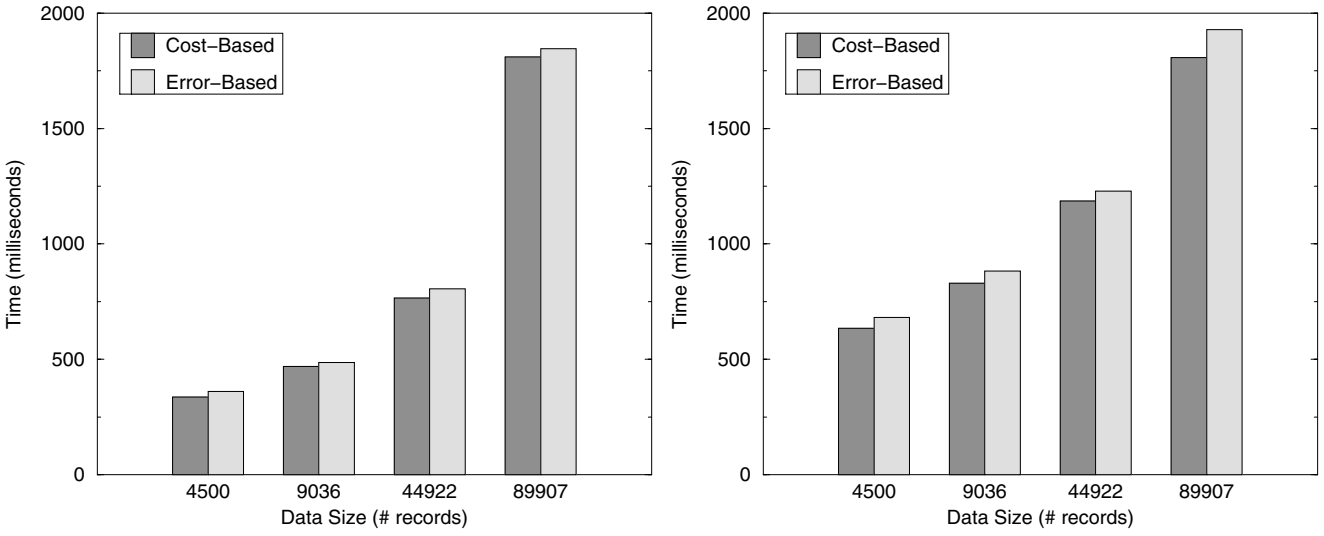
(a) Mean cost (b) Error probability

Fig. 7. $c_{21} = 0.5, c_{20} = 0, 0.1, ..., 1$

has a lower error probability value for the different data sizes since it produces less wrong decisions.

The second set of experiments examines the effect of increasing the values of c_{20}, and c_{21} in several ways by using one data set. Figures 4, 5, 6, and 7 show the results of these experiments. The figures show that the cost-based model has always a lower mean cost value than the error-based model. Moreover, since the error-based model does not rely on the costs in determining the matching status of the record pairs, the error probability value is the same in all the experiments, and is lower than its counterpart in the cost-based model. Figure 5(a) shows that as far as the two costs c_{20}, and c_{21} add up to 1, the mean cost of the cost-based model has the same value. Figures 4(a), 6(a), and 7(a) show that the mean cost of the cost-based model increases slightly till it converges to the same value. A similar behavior for the error probability value for the cost-based model can be noticed from Figs. 4(b), 5(b), 6(b), and 7(b).

Finally, a time performance experiment is exploited to compare both models considering the computation time. Figure 8 shows the results of this experiment using two different searching methods, blocking and the sorted-neighborhood. The costs are fixed to $c_{10} = c_{31} = 0$, $c_{11} = c_{30} = 1$, and $c_{20} = c_{21} = 0.5$. Figure 8 shows that the cost-based model usually takes less computation time than the error-based model. Moreover, the computation time increases as the data size increases, which is understandable, as the number of record pairs the model works on increases. This is also the reason that sorted-neighborhood method takes more time than the blocking one since it produces more record pairs.



(a) Blocking

(b) Sorted-neighborhood

Fig. 8. Time performance

7 Conclusions

This paper presents a new cost optimal decision model for the record matching process. The proposed model uses the ratio of the prior odds of a match along with appropriate values of thresholds to partition the decision space into three decision areas. The model that we presented, is similar with the one proposed by Fellegi and Sunter [5] as it uses the same criterion for discriminating between matches and non-matches. The major difference between our model and all the other already existing models is that it minimizes the cost of making a decision rather than the probability of error in a decision. Our model is also much more efficient than other error-based models, as it does not resort to the sorting of the agreement and of the disagreement ratios in order to select the threshold values.

Our future plans regarding this work, are to develop a cost decision model with a decision space of higher dimensionality. This will allow the decision maker to use the model in its full swing, as the model will provide a more precise and accurate decision for the record matching problem. In our future endeavors, we are also considering the design of a model for cost and time optimal record matching. By using such a model, it will be feasible not only to make a decision based on the entire comparison vector, but also to acquire as many comparison components as possible, in order to make a certain decision. This will save some very important computation time, and at the same time it will facilitate online decision making in the record matching context.

References

1. W. Alvey, B. Jamerson (1997) Record linkage techniques – 1997. Proc. International Workshop and Exposition, March, Federal Committee on Statistical Methodology, Office of Management and Budget
2. D. Bitton, D.J. DeWitt (1983) Duplicate record elimination in large data files. ACM Trans Database Syst 8(2):255–265
3. M.G. Elfeky, V.S. Verykios, A.K. Elmagarmid (2002) TAILOR: A record linkage toolbox. Proc. 18th International Conference on Data Engineering, pp 17–28
4. U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (1996) Advances in knowledge discovery and data mining. AAAI/MIT
5. I.P. Fellegi, A.B. Sunter (1969) A theory for record linkage. J Am Stat Assoc 64(328):1183–1210
6. H. Galhardas, D. Florescu, D. Shasha, E. Simon, C.-A. Saita (2001) Declarative data cleaning: language, model, and algorithms. Proc. 27th International Conference on Very Large Data Bases, pp 371–380
7. L. Gravano, P.G. Ipeirotis, H.V. Jagadish, N. Koudas, S. Muthukrishnan, D. Srivastava (2001) Approximate string joins in a database (almost) for free. Proc. 27th International Conference on Very Large Data Bases, pp 491–500
8. M.A. Hernadez (1996) A generalization of band joins and the merge/purge problem. Ph.D. thesis, Department of Computer Sciences, Columbia University
9. M.A. Hernadez, S.J. Stolfo (1998) Real-world data is dirty: data cleansing and the merge/purge problem. Data Mining Knowl Discovery 2(1):9–37
10. J.A. Hylton (1996) Identifying and merging related bibliographic records. Master's thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Mass., USA
11. M.A. Jaro (1989) Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. J Am Stat Assoc 84, no. 406, 414–420
12. P.R. Kelley (1985) Advances in record linkage methodology: a method for determining the best blocking strategy. Proc. Workshop on Exact Matching Methodologies, pp 199–203
13. B. Kliss, W. Alvey (1985) Record linkage techniques – 1985. Proc. Workshop on Exact Matching Methodologies, May, Department of the Treasury, Internal Revenue Service, Statistics Income Division

14. M.L. Lee, H. Lu, T.W. Ling, Y.T. Ko (1999) Cleansing data for mining and warehousing. Proc. 10th International Conference on Databases and Expert Systems Applications
15. U. Manber (1989) Introduction to algorithms. Addison-Wesley, Reading, Mass., USA
16. A.E. Monge, C.P. Elkan (1996) The field matching problem: algorithms and applications. Proc. 2nd International Conference on Knowledge Discovery and Data Mining, pp 267–270
17. A.E. Monge, C.P. Elkan (1997) An efficient domain-independent algorithm for detecting approximately duplicate database records. Proc. SIGMOD Workshop on Research Issues on DMKD, pp 23–29
18. A.E. Monge (1997) Adaptive detection of approximately duplicate records and the database integration approach to information discovery. Ph.D. thesis, Department of Computer Science and Engineering, University of California, San Diego, Calif., USA
19. G. Nathan (1967) Outcome probabilities for a record matching process with complete invariant information. J Am Stat Assoc 62(318):454–469
20. H.B. Newcombe, J.M. Kennedy (1962) Record linkage: making maximum use of the discriminating power of identifying information. Comm ACM 5:563–566
21. H.B. Newcombe, J.M. Kennedy, S.J. Axford, A.P. James (1959) Automatic linkage of vital records. Science 130(3381):954–959
22. H.B. Newcombe (1967) Record linking: the design of efficient systems for linking records into individual and family histories. Am J Hum Genet 19(3)
23. N.S. D'Andrea Du Bois (1969) A solution to the problem of linking multivariate documents. J Am Stat Assoc 64(325):163–174
24. J.C. Pinheiro, D.X. Sun (1998) Methods for linking and mining heterogeneous databases. Proc. 4th International Conference on Knowledge Discovery and Data Mining, pp 309–313
25. B.J. Tepping (1968) A model for optimum linkage of records. J Am Stat Assoc 63:1321–1332
26. P.D. Turney (2000) Types of cost in inductive concept learning. Workshop on Cost-Sensitive Learning at the 17th International Conference on Machine Learning, Stanford, Calif., USA