Optimal Stopping: A Record-Linkage Approach

GEORGE V. MOUSTAKIDES
University of Patras, Greece
and
VASSILIOS S. VERYKIOS
University of Thessaly, Greece

Record-linkage is the process of identifying whether two separate records refer to the same real-world entity when some elements of the record's identifying information (attributes) agree and others disagree. Existing record-linkage decision methodologies use the outcomes from the comparisons of the whole set of attributes. Here, we propose an alternative scheme that assesses the attributes sequentially, allowing for a decision to made at any attribute's comparison stage, and thus before exhausting all available attributes. The scheme we develop is optimum in that it minimizes a well-defined average cost criterion while the corresponding optimum solution can be easily mapped into a decision tree to facilitate the record-linkage decision process. Experimental results performed in real datasets indicate the superiority of our methodology compared to existing approaches.

Categories and Subject Descriptors: G.3 [**Mathematics of Computing**]: Probability and Statistics—*Probabilistic algorithms*; H.2.4 [**Database Management**]: Systems—*Textual databases*

General Terms: Management, Performance

Additional Key Words and Phrases: Record-linkage, optimal stopping, duplicate detection

ACM Reference Format:
Moustakides, G. V. and Verykios, V. S. 2009. Optimal stopping: A record-linkage approach. ACM J. Data Inform. Quality 1, 2, Article 9 (September 2009), 34 pages. DOI = 10.1145/1577840.1577844. http://doi.acm.org/10.1145/1577840.1577844.

Authors' addresses: G. V. Moustakides, Department of Electrical and Computer Engineering, University of Patras, 26500 Rion, Greece; email: moustaki@ece.upatras.gr; V. S. Verykios, Department of Computer and Communication Engineering, University of Thessaly, 38221 Volos, Greece; email: verykios@inf.uth.gr.

1. INTRODUCTION

Record-linkage has a long history of uses for statistical surveys and administrative data files. It refers to the process of identifying records on the same or two different databases that contain information about the same entity. These records usually correspond to a person, an organization, or an institution although they could also be places or residences, real estate properties, criminal cases, licenses to carry out an activity such as the sale of pesticides or drugs or anything else that could be the subject of a database record. There are two basic reasons to try to link records: data collation and list construction. In the data collation setting, a project might require data that are not all available from the same database. Such a project might involve checking the consistency between earnings reported on income tax returns and earnings reported by employers to the Social Security Administration. From the list construction point of view, some projects require a list of all members of a population to serve as a sampling frame, the contact list for a census, or for collation with data from other sources. It often happens that there is no single list of the population, but that a combination of several lists can be expected to include all or most of it. Usually, there is some overlap between these lists. To avoid biasing the sample or census, we must delete duplicate records so that each member of the population is included only once. This “deduplication” process requires that we are able to identify people or other entities that are included in more than one list.

The record-linkage process would be greatly simplified if each individual had used the same unique identifier (such as the driver’s license number of the full Social Security Number) in each database. In this case, matching (linking) records across databases would have been easy. However, in the absence of a unique identifier, it is necessary to use combinations of fields in order to match records. Matches based on the comparison of corresponding fields such as first name, last name, address, and date of birth are inherently inferential, and for this reason prone to higher rates of error such as false matches (a match is indicated when in fact the two records refer to different individuals) or false nonmatches (a nonmatch is indicated when the two records refer to the same individual). The former type of record-linkage that relies on unique identifiers is called *exact* or *deterministic* and refers to the matching of records that either contain unique identifiers or their information is error free. The latter type of linkage is known as *probabilistic* or *approximate*. Probabilistic record-linkage refers to the process of linking records that they lack a unique and universal identifier and/or may have become inconsistent because of data entry errors, misspellings, missing information, etc. In this article we focus on the second type of record-linkage.

The two principal steps in the record-linkage process are: (a) the *searching step* for potentially linkable pairs of records and (b) the *matching step* for determining the linkage status of a pair of records. The main goal of the searching step is to keep to a minimum the number of pairs of records that are brought together for comparison. With respect to the matching step, the main problem is to automatically decide upon the matching status of a pair of records when some of the record information is in agreement while the rest is not. The matching step makes use of the outcomes of the comparisons among all the available pairs in order to decide whether the pair matches (correspond to the same entity) or does not match. More extensive introductory information regarding record-linkage and its applications can be found in Winkler [1995].

In this article we introduce a new direction for saving computational time in the record-linkage process while we keep the searching step and a large part of the matching step intact. The approach that we propose is sequential in nature, and optimizes the matching of records by minimizing the number of field comparisons that are necessary for the decision process to deliberate, without affecting the quality of the final decision. More specifically we argue that, as opposed to the classical record-linkage methodology where the entire set of attributes for a pair of records under comparison must be examined, on the average a significantly smaller subset of these attributes is actually needed to achieve similar performance results (such as the same probability of error). The savings in computation time is enormous if we consider the expansion of the comparison space in regular record-linkage scenarios that is analogous to the cardinality (millions of records) and the degree (hundreds of attributes) of the databases to be linked, as well as the strict deadlines usually imposed upon the record-linkage officers for running the process on a weekly basis. For the development of our optimum sequential scheme we heavily rely on results and methodologies coming from *optimal stopping theory for Markov process* [Shiryayev 1978].

The rest of this article is organized as follows. In Section 2 we make a brief literature review and in Section 3 we present the basic elements of the probabilistic record-linkage theory. Section 4 contains our main theoretical developments while Section 5 is devoted to practical issues as fine-tuning of our decision scheme; equivalent tree-structure representation and a blocking type alternative implementation. Section 6 contains possible extensions and variants and Section 7 presents a thorough experimental evaluation. The article is completed with Section 8 which contains our concluding remarks.

2. RELATED WORK

Newcombe et al. [1959] were the first to introduce the ideas of computerized record-linkage. In their inaugural works Newcombe et al. [1959] and Newcombe and Kennedy [1962] proposed decision rules that rely on odds ratios of frequencies, that have been computed apriori, for sorting out matches from nonmatches. Fellegi and Sunter [1969] established the foundations of record-linkage by demonstrating the optimality of decision rules proposed by Newcombe et al. under certain fixed upper bounds on the rates of false matches and false nonmatches. The underlying assumption of these early models was the conditional independence of the fields in the agreement pattern. Winkler [1993] showed how to estimate the model parameters using the EM algorithm [Dempster et al. 1977] and demonstrated that a properly applied EM algorithm

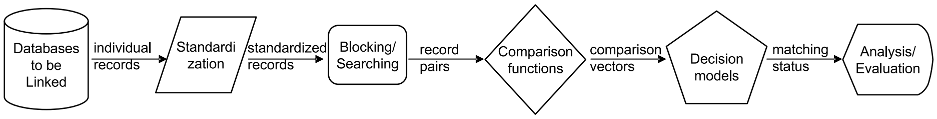

Fig. 1. Process diagram of the record-linkage process.

provides suitable estimates of optimal parameters in a number of situations. Extensions of the basic parameter estimation approach have been proposed to those cases where the different fields used in the EM algorithm can have dependencies upon one another.

Tepping [1968] was the first to propose a record-linkage model focusing on the costs of the matching decisions rather than purely on the errors by presenting a graphical approach for estimating the likelihood thresholds. Verykios et al. [2003] developed a formal framework for the cost-based approach, introduced by Tepping, by demonstrating an analytical solution for the computation of the thresholds for the three decision areas problem. Bilenko et al. [2003], by using SVMlight [Joachims 1999] for learning how to combine the independent results of individual fields, indicated that the SVM approach outperforms other simpler approaches like treating the entire record as one field. A number of record-linkage models relying on supervised and semisupervised techniques appear in Cohen and Richman [2002], McCallum and Wellner [2004], and Singla and Domingos [2004]. Various record-linkage approaches which rely on active learning have been proposed in Sarawagi and Bhamidipaty [2002] and Tejada et al. [2002].

Distance-based approaches alleviate the problem of having available training data, or some domain expert. Monge and Elkan [1996] proposed a string matching algorithm for detecting similar records by applying a general-purpose field matching algorithm. Cohen [2000] suggested combining the TF.IDF weighting scheme with the cosine metric to measure the similarity of records. Guha et al. [2004] proposed creating a distance metric that is based on ranked list merging. Ananthakrishna et al. [2002] proposed a distance metric that takes into account the cooccurrence similarity of a pair of records. Recently, Chaudhuri et al. [2003] proposed a new framework by observing that the distance thresholds for detecting duplicate records should vary according to the record at hand. Finally, various rule-based approaches presented in Hernández and Stolfo [1995] and Galhardas et al. [2001] as well as some unsupervised learning approaches proposed in Ravikumar and Cohen [2004] and Bhattacharya and Getoor [2005] constitute a representative sample of different solutions which have been proposed for the record-linkage problem.

3. BACKGROUND

Figure 1 illustrates the big picture of the record-linkage process. The various steps of the process, along with the transformations the input data are undergone, are explained in the following paragraphs.

In the product space of two database tables, a *match* is a pair of records that represent the same entity and a *nonmatch* is a pair that represent two different entities. Within a single database, a *duplicate record* represents the same entity as another record in the same database. Instead of considering all record pairs in the product space, the search of matches is usually constrained to those pairs that agree on certain fields or parts of fields, which are known as *blocking* variables or criteria. Errors resulting from failing to compare tentative matches (missed matches are those false nonmatches that do not agree on a set of blocking criteria) is a side effect of blocking.

Matching variables are common identifiers (such as name, address, annual receipts, or tax code number) that are used to identify matches. Where possible, sequences of strings of characters like business names and addresses need to be *standardized* which is to be parsed or separated into components so as to allow for better comparison and hence improve matching accuracy. The basic ideas of standardization are to replace the many spelling variations of commonly occurring words with standard spellings and use certain keywords found during standardization as hints for parsing procedures. A user dealing with a special population can improve the standardization of the data by using certain terms pertaining to the population. For example, if matching involves a list of military personnel, the list of titles to be used might be augmented with the various terms for military ranks, such as “sergeant” and “captain.”

A *record-linkage decision rule* is a rule that designates a pair either as a link, a possible link, or a nonlink. Possible links are those pairs for which the identifying data are insufficient to provide evidence whether a pair is a match. In this situation, clerks review possible links and determine their match status. Mistakes can and do actually occur in matching such as false matches and false nonmatches. Generally, *link/nonlink* refers to designations under decision rules and *match/nonmatch* refers to true status. The *matching weight* or *score* is a number assigned to a pair that simplifies assignment of link and nonlink status via decision rules. A procedure or matching variable has more *distinguishing power* if it is more appropriate to delineate matches and nonmatches than another.

For many projects, automated matching decision rules are developed using ad hoc and intuitive approaches. Ad hoc rules are easily developed and may yield good results. The disadvantage is that ad hoc rules may not be applicable to pairs that are different from those used in defining the rule. Users seldom evaluate ad hoc rules with respect to false match and false nonmatch rates. In the 1950’s, Newcombe et al. [1959] introduced concepts of record-linkage that were formalized in the mathematical model of Fellegi and Sunter [1969]. Fellegi and Sunter’s paper provides: (a) methods for estimating outcome probabilities that do not rely on intuition or past experience, (b) estimates of error rates that do not require manual intervention, and (c) automatic threshold choice based on estimated error rates. Fellegi and Sunter considered the likelihood ratio of the form $\mathcal{L} = \mathbb{P}[\gamma\,|\mathcal{M}]/\mathbb{P}[\gamma\,|\mathcal{U}]$, where γ is an arbitrary agreement pattern in the comparison space and $\mathcal{M}$ and $\mathcal{U}$ are the two classes of matches and nonmatches. The ratio $\mathcal{L}$ or any monotonically increasing function of it is referred to as a matching weight or score while the probabilities are called matching parameters. A decision rule $\mathcal{D}$ provides three outcomes for record pairs and is given by the following.

$$\mathcal{D} = \begin{cases} \text{match} & \text{if } \mathcal{L} \geq \text{UPPER} \\ \text{possible match} & \text{if LOWER} < \mathcal{L} < \text{UPPER} \\ \text{nonmatch} & \text{if } \mathcal{L} \leq \text{LOWER} \end{cases}$$

The cutoff thresholds UPPER and LOWER are determined by apriori error bounds on false matches and false nonmatches. For the decision rule to work, the matching parameters (probabilities $\mathbb{P}[\gamma\,|\mathcal{M}]$, $\mathbb{P}[\gamma\,|\mathcal{U}]$, and $\mathbb{P}[\mathcal{M}]$) must be computed. Fellegi and Sunter's paper provides methods of estimating matching probabilities and error rates as well as the appropriate thresholds based on estimated error rates.

Fellegi and Sunter showed that the decision rule they proposed is optimal in that for any pair of fixed upper bounds on the rates of false matches and false nonmatches, the clerical review region is minimized over all decision rules on the same comparison space. The theory holds on any subset such as pairs agreeing on a postal code, street name, or part of a name field. In actual applications the optimality of their decision rule heavily depends on the estimated matching parameters. For a thorough and complete presentation of various models for record-linkage the interested reader should refer to Elmagarmid et al. [2007].

4. MAIN RESULTS

Suppose we have a set of K attributes[1] and *we have specified the order by which they are going to be compared*. Let $\{\xi_n\}_{n=1}^{K}$ denote the sequence of comparison outcomes, with $\xi_n \in \{0, 1\}$ and "0" and "1" denoting agreement and disagreement, respectively. For each attribute we are given the probabilities $p_n^{\mathcal{U}}(\xi)$, $p_n^{\mathcal{M}}(\xi)$, $n = 1, \ldots, K, \xi \in \{0, 1\}$, with the first quantity denoting the probability of the comparison of the nth attribute to produce the outcome ξ when the true hypothesis is $H_{\mathcal{U}}$ (nonmatch) and the second when the true hypothesis is $H_{\mathcal{M}}$ (match). Notice that these prior probabilities need not be the same for every attribute; this is why they depend on n. We also assume that the random variables ξ_n are *independent under each hypothesis H_i*. This means that the set of outcomes $\{\xi_1, \ldots, \xi_n\}$ has a conditional joint probability which is given by the following product

$$\mathbb{P}[\xi_1, \ldots, \xi_n | H_i] = \prod_{k=1}^{n} p_k^i(\xi_k);\ i = \mathcal{U}, \mathcal{M}, \tag{1}$$

with $\mathbb{P}[A|B]$ denoting the probability of the event A conditioned on the event B. Since our intention is to follow a Bayesian approach, we need to specify the *prior* probability $\mathbb{P}[H_{\mathcal{M}}] = p$ of having a match and therefore $\mathbb{P}[H_{\mathcal{U}}] = 1 - p$ is the probability of having a nonmatch.

Existing decision schemes compare first the whole set of attributes thus generating K outcomes $\{\xi_1, \ldots, \xi_K\}$ which are then used to make a selection

[1] In the sequel the words "attributes" and "fields" are used interchangeably.

$\mathcal{D}$ with $\mathcal{D} \in \{1, \ldots, L\}$. In other words, the decision scheme selects among L different possibilities which, without loss of generality, we enumerate with the first L positive integers. Actually these integers are simply labels for more practically meaningful decisions. As we pointed out in Section 3 the most common example corresponds to the case where $L = 3$ with $\mathcal{D} = 1$ denoting selection in favor of "nonmatched," $\mathcal{D} = 2$ selection in favor of "need of clerical review," and $\mathcal{D} = 3$ selection in favor of "matched."

In this work, instead of first comparing all attributes and then using the outcomes to make a selection, we propose the application of a sequential scheme that exploits the attributes gradually. We start by comparing the first attribute that generates the outcome ξ_1. The accumulated information in this first stage is simply the set $\{\xi_1\}$. *We then ask ourselves whether the accumulated information is adequate to make a reliable selection from the L existing possibilities*. If the answer is positive we *stop* using any additional attributes and proceed to the selection process, by selecting one of the L possibilities, and the whole process terminates; if the answer is negative we *continue* and use the second attribute. In the latter case, by comparing the second attribute we generate the outcome ξ_2 which enriches the accumulated information with a new element thus becoming $\{\xi_1, \xi_2\}$. Again we ask ourselves whether this information is adequate to make a reliable selection or not. If the answer is positive we stop using any additional attributes, we proceed to the selection process, we select one out of the L possibilities, and the whole process is terminated; if the answer is negative we continue with the third attribute. The process is repeated until we either stop and make a selection or exhaust all attributes and then enforce a final selection. What is crucial in this scheme is the fact that at any stage the decision stop/continue *is based on the accumulated information until this stage* and not on any other (for example, future stage) information.

If $\mathcal{N}$ denotes the number of attributes we used, it is clear that $\mathcal{N}$ can be any integer less than or equal to the maximum number K of available attributes. In other words, with the sequential scheme we just described we are not forced to necessarily use all attributes. Notice also that $\mathcal{N}$ is *random*, since the decision stop/continue we make at every stage is based on the random data we have accumulated up to that point. This clearly means that different record comparisons will require different number $\mathcal{N}$ of attributes. Another important characteristic is the fact that since we can stop at any stage $\mathcal{N}$ *we can use a different selection strategy $\mathcal{D}_\mathcal{N}$ per stage*. Of course, $\mathcal{D}_\mathcal{N}$ must comply with the same basic rule as our stop/continue decision, namely, it must rely only on the information accumulated up to stage $\mathcal{N}$ which is the set $\{\xi_1, \ldots, \xi_\mathcal{N}\}$.

Let us summarize our sequential scheme. We observe that it is comprised of a pair $(\mathcal{N}, \mathcal{D}_\mathcal{N})$, where $\mathcal{N}$ is a random variable that takes values in the set $\{0, \ldots, K\}$ and $\mathcal{D}_\mathcal{N}$ is a random variable that depends on $\mathcal{N}$ and takes values in the set $\{1, \ldots, L\}$. The first random variable indicates at which attribute we stop and the second which possibility to select. $\mathcal{N}$, however, is not any random variable with values in the specified set. It is characterized by a very important property: The event $\{\mathcal{N} = n\}$ (the decision to stop at stage n) depends *only* on the set $\{\xi_1, \ldots, \xi_n\}$, that is, the information accumulated up to stage n. Such random variables are known under the name of *stopping times* (s.t.). Furthermore, for the random variable $\mathcal{D}_{\mathcal{N}}$ the event $\{\mathcal{D}_{\mathcal{N}} = j\}$ (selection of possibility j), is based on the information accumulated until the time of stopping $\mathcal{N}$, that is, $\{\xi_1, \dots, \xi_{\mathcal{N}}\}$. Our goal is to optimally select the s.t. $\mathcal{N}$ and the selection rule $\mathcal{D}_{\mathcal{N}}$. As we are going to see, the latter problem is rather straightforward whereas for the former there exists a very strong supporting theory known as *optimal stopping theory*. Since our intention is to find a scheme which is optimum, it is clear that we first need to define an appropriate performance measure. Our intention is to adopt a Bayesian approach where we are going to penalize with known costs the use of attributes and the result of our final selection. This will give rise to an *average cost* (exactly as in the classical nonsequential case) which we will attempt to minimize by properly selecting $\mathcal{N}$ and $\mathcal{D}_{\mathcal{N}}$.

4.1 A Bayesian Setup

Suppose we are given costs c_n, $n = 1, \dots, K$, with $c_n > 0$ denoting the cost of using the nth attribute. This cost indicates the complexity of computing the comparison outcome between the two values of the corresponding attribute from the pair of compared records. Consider also constants $C_{ji} \geq 0$, $j = 1, \dots, L$; $i = \mathcal{U}, \mathcal{M}$, with C_{ji} denoting the cost of selecting possibility j when the true hypothesis is H_i. It is then clear that any pair $(\mathcal{N}, \mathcal{D}_{\mathcal{N}})$ produces an *average cost* which can be written as

$$\mathcal{C}(\mathcal{N}, \mathcal{D}_{\mathcal{N}}) = \mathbb{E}\left[\sum_{n=1}^{\mathcal{N}} c_n\right] + \sum_{j=1}^{L} \sum_{i=\mathcal{U},\mathcal{M}} C_{ji}\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \,\&\, H_i], \tag{2}$$

where $\mathbb{P}[\cdot]$, as we said, denotes probability and $\mathbb{E}[\cdot]$ expectation. In the average cost, which we intend to use as our *performance measure*, we clearly distinguish two parts. The first expresses the average cost for using the attributes and the second the average cost due to our selection strategy. It is obvious that our ultimate goal is to produce the pair $(\mathcal{N}, \mathcal{D}_{\mathcal{N}})$ that will *minimize the average cost*.

In order to understand the difference of our scheme with the existing techniques, we recall that in a nonsequential setup we first use *all* attributes and then apply a selection strategy $\mathcal{D}$ which employs the whole set of comparison outcomes $\{\xi_1, \dots, \xi_K\}$. It is therefore evident that for such a case, the average cost becomes

$$\mathcal{C}(\mathcal{D}) = \sum_{n=1}^{K} c_n + \sum_{j=1}^{L} \sum_{i=\mathcal{U},\mathcal{M}} C_{ji}\mathbb{P}[\mathcal{D} = j \,\&\, H_i],$$

with the first part being now outside the expectation since it is completely deterministic and common to all schemes. The only unknown here is the selection strategy $\mathcal{D}$. If we desire to minimize the average cost in order to optimize $\mathcal{D}$ then it is sufficient to consider only the second part (see Verykios and Moustakides [2004]). In the approach we propose here we clearly need to take into account both parts of the average cost, since we are allowed to stop at any stage and make a selection before exhausting all attributes.

Our intention is to minimize the average cost defined in Eq. (2) in two steps. First, for any given s.t. $\mathcal{N}$, we are going to specify the optimum selection strategy $\mathcal{D}_{\mathcal{N}}$. Since the resulting cost will be a function only of $\mathcal{N}$, we will then optimize with respect to $\mathcal{N}$. Before proceeding with the details of our analysis, we need to introduce a number of definitions and background results.

LEMMA 1. *Assume that we have performed n attribute comparisons with corresponding outcomes $\{\xi_1, \dots, \xi_n\}$, then the joint probability of this event is equal to*

$$\mathbb{P}[\xi_1, \dots, \xi_n] = p \prod_{k=1}^{n} p_k^{\mathcal{M}}(\xi_k) + (1-p) \prod_{k=1}^{n} p_k^{\mathcal{U}}(\xi_k). \tag{3}$$

PROOF. The proof is a direct application of the theorem of total probability. Specifically, we have

$$\begin{aligned}\mathbb{P}[\xi_1, \dots, \xi_n] &= \mathbb{P}[\xi_1, \dots, \xi_n \,\&\, H_{\mathcal{M}}] + \mathbb{P}[\xi_1, \dots, \xi_n \,\&\, H_{\mathcal{U}}] \\ &= \mathbb{P}[\xi_1, \dots, \xi_n | H_{\mathcal{M}}]\mathbb{P}[H_{\mathcal{M}}] + \mathbb{P}[\xi_1, \dots, \xi_n | H_{\mathcal{U}}]\mathbb{P}[H_{\mathcal{U}}].\end{aligned} \tag{4}$$

The two conditional probabilities are simply the corresponding products from (1), while for the prior probabilities used our assumption that $\mathbb{P}[H_{\mathcal{M}}] = 1 - \mathbb{P}[H_{\mathcal{U}}] = p$. □

Let us now consider the *posterior probability π_n that the true hypothesis is* $H_{\mathcal{M}}$ conditioned on the event that the available information is $\{\xi_1, \dots, \xi_n\}$. In other words, we are interested in $\pi_n = \mathbb{P}[H_{\mathcal{M}} | \xi_1, \dots, \xi_n]$. Using the Bayes rule we can write

$$\begin{aligned}\pi_n = \mathbb{P}[H_{\mathcal{M}} | \xi_1, \dots, \xi_n] &= \frac{\mathbb{P}[\xi_1, \dots, \xi_n | H_{\mathcal{M}}]\mathbb{P}[H_{\mathcal{M}}]}{\mathbb{P}[\xi_1, \dots, \xi_n]} \\ &= \frac{p \prod_{k=1}^{n} p_k^{\mathcal{M}}(\xi_k)}{p \prod_{k=1}^{n} p_k^{\mathcal{M}}(\xi_k) + (1-p) \prod_{k=1}^{n} p_k^{\mathcal{U}}(\xi_k)}.\end{aligned} \tag{5}$$

What is interesting here is the fact that we can find a convenient recurrence formula for the computation of π_n. This is given in the following lemma.

LEMMA 2. *Let π_{n-1} denote the posterior probability at stage $n-1$ and suppose that at stage n the nth attribute comparison generates the outcome ξ_n, then*

$$\pi_n = \frac{\pi_{n-1} p_n^{\mathcal{M}}(\xi_n)}{\pi_{n-1} p_n^{\mathcal{M}}(\xi_n) + (1-\pi_{n-1}) p_n^{\mathcal{U}}(\xi_n)};\ \pi_0 = p. \tag{6}$$

PROOF. From Lemma 1, Eq. (5), we can easily show that

$$\frac{\pi_n}{1-\pi_n} = \frac{p}{1-p} \frac{\prod_{k=1}^{n} p_k^{\mathcal{M}}(\xi_k)}{\prod_{k=1}^{n} p_k^{\mathcal{U}}(\xi_k)} = \frac{p}{1-p} \frac{\prod_{k=1}^{n-1} p_k^{\mathcal{M}}(\xi_k)}{\prod_{k=1}^{n-1} p_k^{\mathcal{U}}(\xi_k)} \frac{p_n^{\mathcal{M}}(\xi_n)}{p_n^{\mathcal{U}}(\xi_n)} = \frac{\pi_{n-1}}{1-\pi_{n-1}} \frac{p_n^{\mathcal{M}}(\xi_n)}{p_n^{\mathcal{U}}(\xi_n)}. \tag{7}$$

The desired formula is then obtained by solving for π_n. □

Since π_n depends only on its previous value and the new acquired information ξ_n, it is clear that $\{\pi_n\}$ forms a Markov process. The next lemma provides another useful identity that we are going to need in our analysis.

LEMMA 3. *The joint probability of the set $\{\xi_1, \ldots, \xi_{n+1}\}$ at stage $n + 1$ conditioned on the event that at stage n we have $\{\xi_1, \ldots, \xi_n\}$ satisfies the relation*

$$\mathbb{P}[\xi_1, \ldots, \xi_{n+1} | \xi_1, \ldots, \xi_n] = \frac{\mathbb{P}[\xi_1, \ldots, \xi_{n+1}]}{\mathbb{P}[\xi_1, \ldots, \xi_n]} = \pi_n p_{n+1}^{\mathcal{M}}(\xi_{n+1}) + (1 - \pi_n) p_{n+1}^{\mathcal{U}}(\xi_{n+1}).$$

PROOF. The proof is straightforward once we substitute the numerator and denominator using Lemma 1 and then apply Eq. (5). □

As we can see, the conditional probability depends on the past *only* through π_n. In other words, the posterior probability summarizes the influence of the past information to future events. In fact, as we are going to establish, π_n is all we need to know in order to define our optimum strategies. Let us now use the previous identities in order to put the average cost under a more suitable form. We first need to introduce some notations and definitions.

From now on, $\mathbb{E}[\cdot]$ denotes expectation with respect to the probability defined in (3). For any event A, we denote its corresponding indicator function with $\mathbb{1}_A$ (in other words $\mathbb{1}_A = 1$ when A occurs, otherwise $\mathbb{1}_A = 0$). This definition suggests that $\mathbb{P}[A] = \mathbb{E}[\mathbb{1}_A]$. For our last definition, we recall that our stopping time $\mathcal{N}$ can take upon the values $\{0, \ldots, K\}$. With this in mind, for any sequence $\{x_n\}$ of random variables, we define

$$x_{\mathcal{N}} = \sum_{n=0}^{K} x_n \mathbb{1}_{\{\mathcal{N}=n\}}. \tag{8}$$

The next lemma provides the necessary elements that will help us rewrite the average cost under its final form.

LEMMA 4. *Consider the probability $\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \& H_i]$, $j = 1, \ldots, L$, $i = \mathcal{M}, \mathcal{U}$, then we can write*

$$\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \& H_{\mathcal{M}}] = \mathbb{E}[\pi_{\mathcal{N}} \mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}}] \mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \& H_{\mathcal{U}}] \quad = \mathbb{E}[(1 - \pi_{\mathcal{N}}) \mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}}]. \tag{9}$$

PROOF. Let us first show that if the first relation is true, so is the second. Indeed, since $\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j] = \mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \& H_{\mathcal{M}}] + \mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \& H_{\mathcal{U}}]$, we have $\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \& H_{\mathcal{U}}] = \mathbb{P}[\mathcal{D}_{\mathcal{N}} = j] - \mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \& H_{\mathcal{M}}]$. Using the indicator function, we can write $\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j] = \mathbb{E}[\mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}}]$, therefore the second equality results immediately from the first by simple substitution. Let us now prove the first equality. We have that

$$\begin{aligned}\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \,\&\, H_{\mathcal{M}}] &= \sum_{n=0}^{K} \mathbb{P}[\mathcal{N} = n \,\&\, \mathcal{D}_n = j \,\&\, H_{\mathcal{M}}] = \sum_{n=0}^{K} p\mathbb{P}[\mathcal{N} = n \,\&\, \mathcal{D}_n = j | H_{\mathcal{M}}] \\ &= \sum_{n=0}^{K} p \sum_{\xi_1,\ldots,\xi_n} \mathbb{1}_{\{\mathcal{N}=n\}} \mathbb{1}_{\{\mathcal{D}_n=j\}} \prod_{k=1}^{n} p_k^{\mathcal{M}}(\xi_k) \\ &= \sum_{n=0}^{K} \sum_{\xi_1,\ldots,\xi_n} \mathbb{P}[\xi_1, \ldots, \xi_n] \mathbb{1}_{\{\mathcal{N}=n\}} \mathbb{1}_{\{\mathcal{D}_n=j\}} \pi_n \\ &= \sum_{n=0}^{K} \mathbb{E}[\mathbb{1}_{\{\mathcal{N}=n\}} \mathbb{1}_{\{\mathcal{D}_n=j\}} \pi_n] = \mathbb{E}\left[\sum_{n=0}^{K} \mathbb{1}_{\{\mathcal{N}=n\}} \mathbb{1}_{\{\mathcal{D}_n=j\}} \pi_n\right] \\ &= \mathbb{E}[\mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}} \pi_{\mathcal{N}}].\end{aligned}$$

For the third equality we used the fact that the sum over $\xi_1, \ldots, \xi_n$ of the product of probabilities times a quantity, is simply the definition of the conditional expectation of the quantity. Since $\mathbb{1}_{\{\mathcal{N}=n\}} \mathbb{1}_{\{\mathcal{D}=j\}}$ is the indicator function of the event $\{\mathcal{N} = n \,\&\, \mathcal{D}_n = j\}$, its conditional expectation is equal to the conditional probability of this event. For the forth equality we used the definition of the posterior probability from (5). For the fifth we used a similar property as in the third equality, that is, the sum over $\xi_1, \ldots, \xi_n$ of the probability times a quantity, is the expectation of the quantity.[2] Finally, for the last equality we applied the definition in (8). This concludes the proof. □

With the help of Lemma 4, the average cost in (2) can be rewritten as

$$\mathcal{C}(\mathcal{N}, \mathcal{D}_{\mathcal{N}}) = \mathbb{E}\left[\sum_{n=1}^{\mathcal{N}} c_n\right] + \mathbb{E}\left[\sum_{j=1}^{L} \{C_{j\mathcal{M}} \pi_{\mathcal{N}} + C_{j\mathcal{U}}(1 - \pi_{\mathcal{N}})\} \mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}}\right]. \tag{10}$$

We are now in a position to find the optimum pair $(\mathcal{N}, \mathcal{D}_{\mathcal{N}})$. We start by optimizing with respect to $\mathcal{D}_{\mathcal{N}}$, a task that turns out to be particularly simple.

Running example. To facilitate the understanding of our methodology we introduce the same example employed in Tepping [1968], for which we present its solution gradually along with the derivation of our results. Our goal is to compare two records and make a decision as to their matching status based on $K = 3$ attributes: surname, first name, and sex. The outcome ξ_n of each attribute comparison is either 0 or 1. The probabilities $p_n^{\mathcal{M}}(\xi_n)$, $p_n^{\mathcal{U}}(\xi_n)$ of the nth attribute deciding 0 or 1 under matched or nonmatched conditions are depicted in Table I.

Regarding the possible decisions, we assume that $L = 3$ with decision $\mathcal{D} = 1$ corresponding to "nonmatched," $\mathcal{D} = 2$ indicating "clerical review," and $\mathcal{D} = 3$ corresponding to "matched." We select the decision costs C_{ij} as follows: $C_{1\mathcal{M}} =$

[2] This is true because π_n and the events $\{\mathcal{N} = n\}$, $\{\mathcal{D}_n = j\}$ depend solely on the random variables $\{\xi_1, \ldots, \xi_n\}$.

Table I. Attributes and Attribute Probabilities for Running Example

Attribute	$p_n^{\mathcal{M}}(1)$	$p_n^{\mathcal{M}}(0)$	$p_n^{\mathcal{U}}(1)$	$p_n^{\mathcal{U}}(0)$
Surname	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

$C_{3\mathcal{U}} = 1$, $C_{2\mathcal{M}} = C_{2\mathcal{U}} = 0.15$, and $C_{3\mathcal{M}} = C_{1\mathcal{U}} = 0$. In other words, the cost of making an incorrect decision is equal to 1, the cost of a correct decision is 0, and finally the cost of asking clerical review is valued 0.15. Note that without the second choice (clerical review) this specific selection of costs yields the decision error probability.

Up to this point the parameters of our approach coincide with the ones of the classical theory [Verykios and Moustakides 2004; Verykios et al. 2003]. The element that discriminates our methodology from the existing one is the fact that we impose cost on the usage of each attribute. One might argue that we could adopt the same idea in the classical methodology as well. This is indeed true, however, in the classical case, the attribute costs have no effect in the definition of the optimum decision scheme since by employing *the entire set* of attributes we simply incur a constant contribution to the average cost (the sum of all attribute costs). In the sequential methodology proposed here, this is clearly not the case. By using a different number $\mathcal{N}$ of attributes in each record pair we contribute with a different portion in the total cost. Consequently, attribute costs play a fundamental role in selecting the right moment to stop and make a decision.

For our example we select the following attribute cost values: $8 \times c$ for surname, $5 \times c$ for first name, and $1 \times c$ for sex. Quantities 8,5,1 roughly reflect the average number of letters existing in each attribute and c the processing cost per letter. For our example we select $c = 0.01$ since this value, as we are going to see, yields simple yet pedagogical results.

4.2 Optimal Selection Strategy

For ϖ in the interval [0, 1], we define the function $g(\varpi)$ as follows

$$g(\varpi) = \min_{1\le j\le L}\{C_{j\mathcal{M}}\varpi + C_{j\mathcal{U}}(1-\varpi)\}. \tag{11}$$

Notice that $g(\varpi)$ is a known, deterministic, continuous, and piece-wise linear. Using $g(\varpi)$ we can now find the optimum selection strategy for any *given* s.t. $\mathcal{N}$. Our first theorem provides the desired result.

THEOREM 1. *Fix the s.t. $\mathcal{N}$ then, for any selection rule $\mathcal{D}_{\mathcal{N}}$ we have that*

$$\sum_{j=1}^{L}\{C_{j\mathcal{M}}\pi_{\mathcal{N}} + C_{j\mathcal{U}}(1-\pi_{\mathcal{N}})\}\mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}} \ge g(\pi_{\mathcal{N}})$$

with equality attained by the following rule

$$\mathcal{D}_{\mathcal{N}}^{o} = \arg\min_{1\le j\le L}\{C_{j\mathcal{M}}\pi_{\mathcal{N}} + C_{j\mathcal{U}}(1-\pi_{\mathcal{N}})\}, \tag{12}$$

which is optimum.

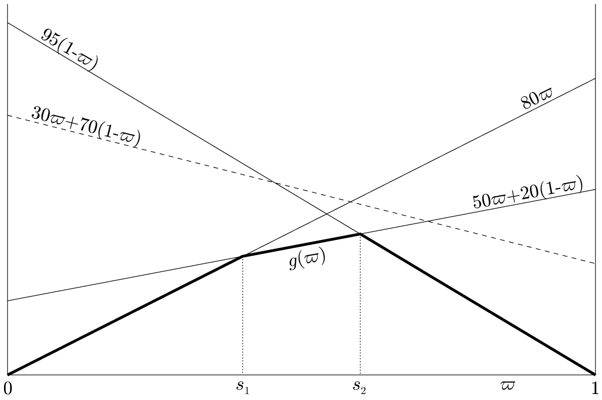

Fig. 2. Formation of function $g(\varpi)$ (thick) for $L = 4$. The third possibility (dashed) is never selected.

PROOF. Since at any stage the selection strategy $\mathcal{D}_{\mathcal{N}}$ can select only one out of the L available possibilities, it is clear that $\sum_{j=1}^{L} \mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}} = 1$ (only one of the terms in the sum becomes 1 while all remaining terms are equal to 0). From $\{C_{j\mathcal{M}}\varpi + C_{j\mathcal{U}}(1-\varpi)\} \geq g(\varpi)$ (remember $g(\varpi)$ is the minimum of these linear functions) and the fact that $\mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}}$ is nonnegative, we conclude that

$$\begin{aligned}\sum_{j=1}^{L}\{C_{j\mathcal{M}}\pi_{\mathcal{N}} + C_{j\mathcal{U}}(1-\pi_{\mathcal{N}})\}\mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}} &\geq g(\pi_{\mathcal{N}})\sum_{j=1}^{N}\mathbb{1}_{\{\mathcal{D}_{\mathcal{N}}=j\}}\\ &= g(\pi_{\mathcal{N}}).\end{aligned}$$

The last quantity is independent from $\mathcal{D}_{\mathcal{N}}$, it thus constitutes a lower bound to *any* selection strategy $\mathcal{D}_{\mathcal{N}}$. We easily observe that this lower bound is attainable only by the rule defined in (12), which is therefore the optimum. □

Figure 2 depicts an example with $L = 4$ possibilities and selection costs $C_{1\mathcal{M}} = 80, C_{2\mathcal{M}} = 50, C_{3\mathcal{M}} = 30, C_{4\mathcal{M}} = 0$; $C_{1\mathcal{U}} = 0, C_{2\mathcal{U}} = 20, C_{3\mathcal{U}} = 70, C_{4\mathcal{U}} = 95$. According to our theory, the line $C_{j\mathcal{M}}\varpi + C_{j\mathcal{U}}(1-\varpi)$ corresponds to selection j. Since we have four possibilities this generates four linear functions with $g(\varpi)$ tracing their minimum. In this figure, the resulting $g(\varpi)$ is marked with a thick line. We can see in this example that $g(\varpi)$ is never equal to the (dashed) line $30\varpi + 70(1-\varpi)$ corresponding to the third possibility. This means that the optimum rule $\mathcal{D}_{\mathcal{N}}^{o}$ completely avoids possibility 3, since there is always an alternative selection with smaller cost.

From Theorem 1 we deduce that $\mathcal{C}(\mathcal{N}, \mathcal{D}_{\mathcal{N}}^{o}) \leq \mathcal{C}(\mathcal{N}, \mathcal{D}_{\mathcal{N}})$ therefore, from now on, for every s.t. $\mathcal{N}$ we are going to use its corresponding optimum selection strategy $\mathcal{D}_{\mathcal{N}}^{o}$. The resulting average cost then becomes

$$\tilde{\mathcal{C}}(\mathcal{N}) = \mathcal{C}(\mathcal{N}, \mathcal{D}_{\mathcal{N}}^{o}) = \min_{\mathcal{D}_{\mathcal{N}}} \mathcal{C}(\mathcal{N}, \mathcal{D}_{\mathcal{N}}) = \mathbb{E}\left[\sum_{n=1}^{\mathcal{N}} c_n + g(\pi_{\mathcal{N}})\right],$$

an expression that depends only on the s.t. $\mathcal{N}$. The goal in our next subsection is to optimize $\tilde{\mathcal{C}}(\mathcal{N})$ with respect to $\mathcal{N}$, that is, to solve the following optimization problem

$$\min_{\mathcal{N}\geq 0} \tilde{\mathcal{C}}(\mathcal{N}) = \min_{\mathcal{N}\geq 0} \mathbb{E}\left[\sum_{n=1}^{\mathcal{N}} c_n + g(\pi_{\mathcal{N}})\right]. \tag{13}$$

This optimization is not as straightforward as the previous one.

4.2.1 *Running Example (continued).* Let us apply the preceding theory to the running example introduced in Section 4.1. Substituting the specific values of the decision costs C_{ij} in (11) we end up with the following $g(\varpi)$ function

$$g(\varpi) = \min\{\varpi, 0.15, 1 - \varpi\}, \tag{14}$$

which can be seen in Figure 4.

4.3 Optimal Stopping Strategy

The optimization in (13) constitutes a classical problem in optimal stopping theory for Markov processes [Shiryayev 1978] and its solution can be immediately obtained by applying the corresponding results. However, in order to make the presentation more intelligible, we decided to follow a less rigorous approach by not making any direct reference to this well-established theory. Unfortunately, such a direction, although more pedagogical, is bound to be analytically lenient, and it is therefore imperative to emphasize that all our results are entirely consistent and fully supported by the optimal stopping theory contained in Shiryayev [1978].

Since every s.t. $\mathcal{N}$ can take upon the values $\{0, 1, \ldots, K\}$, it is evident that the optimum strategy will be characterized by a maximum of $K + 1$ stages. Going from stage 0 to stage K, the optimum scheme must minimize the corresponding average cost. According to the (Bellman) dynamic programming principle, the solution we seek must also be optimum, if instead of the first stage we start from any intermediate stage and continue towards the final one. In other words, if we suppose that we start at stage n having compared the first n attributes with corresponding outcomes $\xi_1, \ldots, \xi_n$, then the optimum strategy *must still be optimum for the remaining stages*. This principle is going to be the basis for deriving our optimum stopping rule.

Let us assume that we are at stage n having already used the first n attributes that generated the outcomes $\xi_1, \ldots, \xi_n$ and led to the summarizing posterior probability π_n. For $n = 0, \ldots, K$, define the *additional-optimum-average-cost* given that $\pi_n = \varpi$ as

$$\mathcal{V}_n(\varpi) = \inf_{\mathcal{N}\geq n} \mathbb{E}\left[\sum_{k=n+1}^{\mathcal{N}} c_k + g(\pi_{\mathcal{N}})|\pi_n = \varpi\right];$$

and for $n = 0, \ldots, K - 1$, the *additional-optimum-average-cost-to-go* as

$$\tilde{\mathcal{V}}_n(\varpi) = \inf_{\mathcal{N}\geq n+1} \mathbb{E}\left[\sum_{k=n+1}^{\mathcal{N}} c_k + g(\pi_{\mathcal{N}})|\pi_n = \varpi\right].$$

Function $\mathcal{V}_n(\varpi)$ expresses the additional optimum average cost we need to pay to complete our task, given that we are at stage n and the posterior probability is $\pi_n = \varpi$. Function $\tilde{\mathcal{V}}_n(\varpi)$ on the other hand expresses the additional optimum average cost we need to pay when we *exclude stopping at n*. According to our definition, $\mathcal{V}_0(p)$ is the solution to our original problem defined in (13). The following theorem relates the two costs and provides a useful backward recursion.

THEOREM 2. *For $n = K - 1, \ldots, 0$, function $\tilde{\mathcal{V}}_n(\varpi)$ is related to $\mathcal{V}_{n+1}(\varpi)$ through the equation*

$$\begin{aligned}\tilde{\mathcal{V}}_n(\varpi) = c_{n+1} + \sum_{\xi_{n+1}} \{\varpi p^{\mathcal{M}}_{n+1}(\xi_{n+1}) + (1-\varpi) p^{\mathcal{U}}_{n+1}(\xi_{n+1})\} \times \\ \mathcal{V}_{n+1}\left(\frac{\varpi p^{\mathcal{M}}_{n+1}(\xi_{n+1})}{\varpi p^{\mathcal{M}}_{n+1}(\xi_{n+1}) + (1-\varpi) p^{\mathcal{U}}_{n+1}(\xi_{n+1})}\right)\end{aligned} \tag{15}$$

where $\mathcal{V}_K(\varpi) = g(\varpi)$. Furthermore $\mathcal{V}_n(\varpi)$ is related to $\tilde{\mathcal{V}}_n(\varpi)$ as follows

$$\mathcal{V}_n(\varpi) = \min\{g(\varpi), \tilde{\mathcal{V}}_n(\varpi)\}. \tag{16}$$

PROOF. When $n = K$ we have exhausted all attributes and we are left with the selection process. With the help of the posterior probability $\pi_K = \varpi$ we select among the L possibilities. Optimal selection has cost $g(\varpi)$, therefore $\mathcal{V}_K(\varpi) = g(\varpi)$.

Assume now that we are at an intermediate stage n with the corresponding posterior probability π_n being equal to ϖ. Let us first verify the validity of Eq. (15). Being at stage n and choosing not to stop means that we will use the $(n + 1)$st attribute and then continue optimally. Comparing the attribute costs c_{n+1} and will produce the outcome ξ_{n+1}, with the help of which we will update the posterior probability from $\pi_n = \varpi$ to π_{n+1}. From this point on (stage $n + 1$) we will continue optimally and, according to our definition, this will have an additional optimal cost $\mathcal{V}_{n+1}(\pi_{n+1})$. Therefore the total cost-to-go is $c_{n+1} + \mathcal{V}_{n+1}(\pi_{n+1})$. Notice, however, that since we are still at stage n we do not yet know what ξ_{n+1} is going to be. Consequently, we need to consider the expected value of the total cost *conditioned on the fact that* $\pi_n = \varpi$. In other words, the optimum-average-cost-to-go is equal to

$$\tilde{\mathcal{V}}_n(\varpi) = c_{n+1} + \mathbb{E}[\mathcal{V}_{n+1}(\pi_{n+1}) | \pi_n = \varpi].$$

From this relation we immediately obtain (15) if we replace π_{n+1} with its equal from Lemma 2; apply Lemma 3 for the conditional probability and then sum over all possible values of ξ_{n+1} in order to compute the conditional expectation.

To verify Eq. (16) we have to proceed as follows. When at stage n and the available information is $\pi_n = \varpi$, we are faced with two options: Either *stop* and select optimally among the L possibilities, or *continue* to the next attribute and from that point on continue again optimally. The first option has cost $g(\varpi)$ whereas the second, as we have seen, $\tilde{\mathcal{V}}_n(\varpi)$. Clearly *we are in favor of the option with the smallest average cost*. This yields (16). □

From the previous proof we also deduce the optimal stopping strategy. According to what we said *we stop at stage n whenever the cost of stopping is smaller than the optimum-average-cost-to-go*, that is, whenever $g(\pi_n) \le \tilde{\mathcal{V}}_n(\pi_n)$. With Theorems 1 and 2 we have completely identified the two optimal strategies for $\mathcal{N}$ and $\mathcal{D}_{\mathcal{N}}$. Let us summarize our optimum scheme.

Offline. We first compute $g(\varpi)$ from (11) and the K functions $\tilde{\mathcal{V}}_n(\varpi)$, $n = K-1, \ldots, 0$, with the help of Eqs. (15) and (16). This computation is performed only once and offline and requires only apriori information.

Sequential. We start by setting $\pi_0 = p$ (our prior probability that $H_{\mathcal{M}}$ is true) and compare $g(\pi_0)$ with $\tilde{\mathcal{V}}_0(\pi_0)$. If $g(\pi_0) \le \tilde{\mathcal{V}}_0(\pi_0)$ then we stop without consulting any attributes and make a selection applying Eq. (12) of Theorem 1 with $\pi_{\mathcal{N}} = \pi_0$; the whole process is then terminated. If $g(\pi_0) > \tilde{\mathcal{V}}_0(\pi_0)$ we go to stage 1 and use the first attribute.[3] In the latter case, the first attribute comparison will generate an outcome ξ_1. This information is consequently used to compute π_1 by applying the updating formula (6) of Lemma 2. We then compare $g(\pi_1)$ with $\tilde{\mathcal{V}}_1(\pi_1)$. If $g(\pi_1) \le \tilde{\mathcal{V}}_1(\pi_1)$ we stop any further attribute comparison and apply the selection process (12) with $\pi_{\mathcal{N}} = \pi_1$; the whole process is then terminated. If $g(\pi_1) > \tilde{\mathcal{V}}_1(\pi_1)$ we continue with the second attribute. In this latter case the comparison of the second attribute produces the outcome ξ_2 which we use to compute π_2 from (6). We then compare $g(\pi_2)$ with $\tilde{\mathcal{V}}_2(\pi_2)$. If $g(\pi_2) \le \tilde{\mathcal{V}}_2(\pi_2)$ we stop any further attribute comparison, we apply the selection process (12) with $\pi_{\mathcal{N}} = \pi_2$ and terminate; etc. These steps are repeated until we either stop and make a selection or exhaust all attributes and then enforce a final selection using (12) with $\pi_{\mathcal{N}} = \pi_K$.

For the implementation of the optimum scheme, we see that the average costs-to-go $\tilde{\mathcal{V}}_n(\varpi)$ play a crucial role. This is the reason why the next section is devoted to uncover a number of important characteristics of these functions that will facilitate their computation and simplify, considerably, the practical application of the optimum scheme.

5. ALTERNATIVE IMPLEMENTATIONS OF THE OPTIMUM TEST

The fact that the maximum number of attributes K and the number of possible values of ξ_n is finite makes the representation and the numerical computation of $\tilde{\mathcal{V}}_n(\varpi)$ particularly simple. Furthermore, we have a number of interesting properties enjoyed by these functions and by $g(\varpi)$ that allow for an alternative and much simpler implementation of the optimum strategies. This alternative implementation methodology will be adopted in our simulations section in order to present a larger and much more realistic application as compared to the small example that runs through our theory sections.

We start with $g(\varpi)$ which is defined in (11) and, as we have seen, appears in the average cost when we use the optimum selection strategy $\mathcal{D}^o_{\mathcal{N}}$ defined in (12). We have the following important property regarding this function.

[3] Of course, under normal conditions we never expect to have $g(\pi_0) \le \tilde{\mathcal{V}}_0(\pi_0)$ since this amounts to trusting the prior probability p more than any information provided by the attribute comparisons. Furthermore, this case results in making always the same selection! Consequently, we anticipate $g(\pi_0) > \tilde{\mathcal{V}}_0(\pi_0)$, which suggests that we should at least use the first attribute.

LEMMA 5. *The function $g(\varpi)$ is concave, continuous, and piece-wise linear, with at most L linear segments. According to the optimal selection rule $\mathcal{D}^o_{\mathcal{N}}$, in each segment we assign only a single selection from the L possibilities and each selection can be assigned to at most one linear segment.*

PROOF. Function $g(\varpi)$ is concave if and only if for every $\varpi_1, \varpi_2, \epsilon \in [0, 1]$ we have $g(\epsilon\varpi_1 + (1-\epsilon)\varpi_2) \geq \epsilon g(\varpi_1) + (1-\epsilon)g(\varpi_2)$. By using the definition of $g(\varpi)$ from (11), it is straightforward to show this inequality. Concavity also assures continuity. Since in convex/concave functions the derivative is monotonous, a line cannot provide two different segments in $g(\varpi)$ because this would correspond to a nonmonotonous derivative. Therefore each line provides at most one segment to $g(\varpi)$ and as we have seen in Figure 2, it is possible for a line not to participate at all in this function. We thus conclude that $g(\varpi)$ is comprised of a number of line segments that cannot exceed L. Furthermore, since each such segment corresponds to a single line and each line represents a specific selection, it becomes clear that the optimum rule assigns in each segment a single selection (the one represented by the corresponding line). Finally, since each line provides at most one segment, each selection is assigned to at most one segment. □

Using Lemma 5 we can propose an alternative realization of the optimum selection rule $\mathcal{D}^o_{\mathcal{N}}$. We note that each line segment is defined by its two endpoints; it is therefore clear that there are $J \leq L+1$ thresholds of the form $0 = s_0 < s_1 < \cdots < s_{J-1} = 1$, with s_{j-1}, s_j denoting the horizontal coordinates of the endpoints of the jth line segment. This also means that the optimum selection rule $\mathcal{D}^o_{\mathcal{N}}$ makes a specific selection every time the posterior probability π_n falls inside the interval $[s_{j-1}, s_j]$. Consequently, instead of (12), we can use the thresholds s_j and simply examine which interval the posterior probability falls into. Figure 3 presents an example with $L = 3$. We can see the corresponding thresholds $0, s_1, s_2, 1$ that form the three consecutive intervals $[0, s_1]$, $[s_1, s_2]$, $[s_2, 1]$, with the optimum selection rule assigning in each interval the selection dictated by the corresponding line segment. Let us now continue by introducing several important characteristics of the functions $\tilde{\mathcal{V}}_n(\varpi)$.

LEMMA 6. *The functions $\tilde{\mathcal{V}}_n(\varpi)$, $n = 0, \ldots, K-1$, are concave, continuous, and piece-wise linear, with a finite number of linear segments.*

PROOF. We first prove that the functions $\tilde{\mathcal{V}}_n(\varpi)$ are concave. For our proof we need the following result: If $A(x)$ is concave in x, then for nonegative a, b, y the function $B(y) = \{ay + b(1-y)\}A(\frac{ay}{ay+b(1-y)})$ is also concave. We recall that a differentiable function $A(x)$ is concave if and only if $A(x) \leq A(x_0) + A'(x_0)(x - x_0)$ (the graph of a concave function is always below the tangent line at any point). To prove our statement it is thus sufficient to show that $B(y) \leq B(y_0) + B'(y_0)(y - y_0)$. After some tedious but straightforward manipulations this inequality can be shown to be true based on the equivalent inequality for $A(x)$.

Let us use this result to show that $\tilde{\mathcal{V}}_{K-1}(\varpi)$ is concave. In (15) the expression under the sum, for every value of ξ_{n+1}, is concave by direct application of

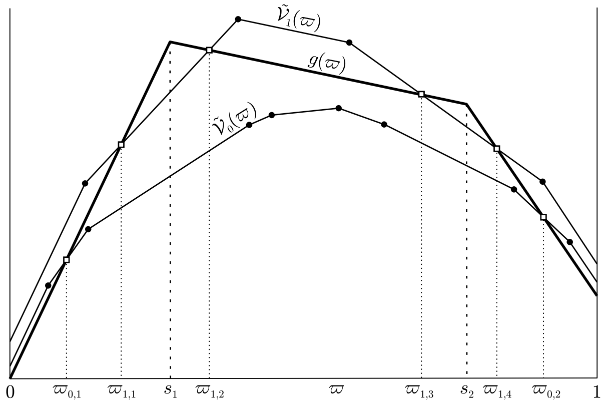

Fig. 3. Typical form of functions $g(\varpi)$ and $\tilde{\mathcal{V}}_n(\varpi)$ for $L = 3$ selection possibilities and $K = 2$ attributes.

our previous result and using the fact that $\mathcal{V}_K(\varpi) = g(\varpi)$ is concave. Since the sum of concave functions is also concave this implies that $\tilde{\mathcal{V}}_{K-1}(\varpi)$ is concave. From this concavity we can also deduce the concavity of $\mathcal{V}_{K-1}(\varpi)$ from (16) because the minimum of two concave functions is also concave. The concavity of $\mathcal{V}_{K-1}(\varpi)$, through (15), assures the concavity of $\tilde{\mathcal{V}}_{K-2}(\varpi)$, etc. We can formally prove our statement for all n using backward induction. We skip the details.

In a similar way we can also prove the piece-wise linear nature of $\tilde{\mathcal{V}}_n(\varpi)$. Again we need to prove an auxiliary result: If $A(x)$ is linear in the interval $x_1 \leq x \leq x_2$ then for nonegative a, b, y the function $B(y) = \{ay + b(1-y)\}A(\frac{ay}{ay+b(1-y)})$ is also linear for $\frac{bx_1}{bx_1+a(1-x_1)} \leq y \leq \frac{bx_2}{bx_2+a(1-x_2)}$. The proof of this statement is straightforward. The two bounds for y assure that $\frac{ay}{ay+b(1-y)} \in [x_1, x_2]$. If $A(x) = c_1x+c_2$ then by direct substitution we can verify that $B(y) = c'_1y+c'_2$. Let us now show that $\tilde{\mathcal{V}}_{K-1}(\varpi)$ is piece-wise linear. In (15) the expression under the sum is piece-wise linear for each value of ξ_{n+1}. This is true by direct application of our previous result and using the fact that $\mathcal{V}_K(\varpi) = g(\varpi)$ is piece-wise linear. Since *finite* sums of piece-wise linear functions produce piece-wise linear functions this means that $\tilde{\mathcal{V}}_{K-1}(\varpi)$ has the desired property. The same property is also passed to $\mathcal{V}_{K-1}(\varpi)$ from (16) because the minimum of two piece-wise linear functions is also piece-wise linear. With the help of (15) the property passes to $\tilde{\mathcal{V}}_{K-2}(\varpi)$, etc. Again we can have a formal proof for all n using backward induction.

Finally for continuity, we have that this property is assured because of the concavity of the functions $\tilde{\mathcal{V}}_n(\varpi)$. □

The fact that $g(\varpi), \tilde{\mathcal{V}}_n(\varpi)$ are piece-wise linear allows for a compact representation of these functions. In particular it is sufficient to keep track of the coordinates of the "corners," that is, the endpoints of their linear segments. Indeed, this is true since any other point can be obtained by simple linear interpolation. For example, in Figure 3, we can completely describe the function $g(\varpi)$ by specifying the four pairs $(0, g(0)), (s_1, g(s_1)), (s_2, g(s_2))$, and $(1, g(1))$. Similarly, for the functions $\tilde{\mathcal{V}}_0(\varpi), \tilde{\mathcal{V}}_1(\varpi)$ we only need the coordinates of the points marked with a dark circle (and the function values at 0 and 1).

Let us now use these characteristics to find an alternative means for describing our optimum stopping rule. We recall that according to the theory we developed in the previous section, we stop at attribute n whenever $g(\pi_n) \leq \tilde{\mathcal{V}}_n(\pi_n)$, otherwise we continue to the next attribute. This rule can be implemented in an alternative way by comparing the posterior probability π_n with thresholds. The following lemma provides the details for this possibility.

LEMMA 7. *There exist an even number I_n of thresholds (depending on the stage n) with $I_n \leq 2L$ of the form $0 = \varpi_{n,0} < \varpi_{n,1} < \cdots < \varpi_{n,(I_n-1)} = 1$ where the intervals $(\varpi_{n,(2i-1)}, \varpi_{n,2i})$ correspond to continuation (to the next stage) and the $[\varpi_{n,2i}, \varpi_{n,2i+1}]$ to stopping. In particular the first and last intervals always correspond to stopping.*

PROOF. Let us first show that $\tilde{\mathcal{V}}_n(0) > g(0)$ and $\tilde{\mathcal{V}}_n(1) > g(1)$ for all $n = K-1, \ldots, 0$. We start with $\tilde{\mathcal{V}}_{K-1}(0)$, we have from (15) that

$$\tilde{\mathcal{V}}_{K-1}(0) = c_K + \sum_{\xi_K} p_K^{\mathcal{U}}(\xi_K)\mathcal{V}_K(0) = c_K + g(0)$$

with the last equality being true because $\mathcal{V}_K(\varpi) = g(\varpi)$ and $\sum_{\xi_K} p_K^{\mathcal{U}}(\xi_K) = 1$. Since the attribute costs c_n were assumed positive we conclude that $\tilde{\mathcal{V}}_{K-1}(0) > g(0)$. This (strict) inequality, when combined with (16), also suggests that $\mathcal{V}_{K-1}(0) = g(0)$. We can now proceed and show our claim for $\tilde{\mathcal{V}}_{K-2}(0)$ in exactly the same way, and more generally for every n using backward induction. Similar proof applies for the second inequality $\tilde{\mathcal{V}}_n(1) > g(1)$.

Let us now investigate the formation of the thresholds $\varpi_{n,i}$. We recall from Lemma 6 that function $\tilde{\mathcal{V}}_n(\varpi)$ is concave. A concave function and a line can intersect at most in two points, therefore each line segment of $g(\varpi)$ can have at most two intersections with $\tilde{\mathcal{V}}_n(\varpi)$. The latter is true for all line segments of $g(\varpi)$ except the ones containing the points 0 and 1 which can intersect $\tilde{\mathcal{V}}_n(\varpi)$ in at most one point (the second is at ∞ because the lines, as we can see in Figure 3, are parallel). Therefore the total number of intersecting points cannot exceed $2 \times (L-2) + 2 = 2L - 2$. If we also include in these points the values 0 and 1, then the total number of points I_n is less or equal than $2L$.

Denote the horizontal coordinates of the intersections as $\varpi_{n,i}$. These quantities constitute our thresholds. Let us order them in increasing order and suppose without loss of generality that $0 = \varpi_{n,0} < \varpi_{n,1} < \cdots < \varpi_{n,(I_n-1)} = 1$ (including 0 and 1). As we have shown $\tilde{\mathcal{V}}_n(0) > g(0)$, this suggests that throughout the whole first interval $[\varpi_{n,0}, \varpi_{n,1}]$ we will have $\tilde{\mathcal{V}}_n(\varpi) > g(\varpi)$ (otherwise, due to continuity, we would have had another intersection point smaller than $\varpi_{n,1}$). In other words, inside the first interval we decide in favor of stopping. In the next interval $[\varpi_{n,1}, \varpi_{n,2}]$, due again to continuity the inequality is reversed to $\tilde{\mathcal{V}}_n(\varpi) < g(\varpi)$, suggesting that this is a continuation interval, and we proceed in this way by switching between stopping and continuation, exactly as described in the lemma. □

With the help of Lemma 7 the decision whether to stop or continue can be made by comparing the posterior probability with the thresholds $\varpi_{n,i}$, $i = 0, \ldots, I_n - 1$. This is entirely equivalent to comparing $g(\pi_n)$ with $\tilde{\mathcal{V}}_n(\pi_n)$. Thus, if $\pi_n \in \cup_l[\varpi_{n,2l}, \varpi_{n,2l+1}]$ we stop, otherwise we go to the $(n+1)$st attribute. If we stop then we proceed to the optimal selection process and compare π_n against the selection thresholds s_j examining which interval $[s_j, s_{j+1}]$ contains the posterior probability. As we have proved, each such interval corresponds to a specific selection. In Figure 3 we can see that for $n = 0$ we have 4 thresholds $0, \varpi_{0,1}, \varpi_{0,2}, 1$, while for $n = 1$ we have 6 (the maximum since $L = 3$) $0, \varpi_{1,1}, \varpi_{1,2}, \varpi_{1,3}, \varpi_{1,4}, 1$. For $n = 0$ if π_0 falls inside the two intervals $[0, \varpi_{0,1}]$, $[\varpi_{0,2}, 1]$ we stop and proceed to the optimal selection process whereas if it falls inside the interval $(\varpi_{0,1}, \varpi_{0,2})$ we continue with the first attribute. Similarly, for $n = 1$ the intervals $[0, \varpi_{1,1}]$, $[\varpi_{1,2}, \varpi_{1,3}]$, $[\varpi_{1,4}, 1]$ are for stopping while the $(\varpi_{1,1}, \varpi_{1,2})$, $(\varpi_{1,3}, \varpi_{1,4})$ for continuation. Finally, for optimal selection we have the three intervals $[0, s_1]$, $[s_1, s_2]$, and $[s_2, 1]$, each one corresponding to a different selection possibility.

5.0.1 *Running Example (continued).* Once the function $g(\varpi)$ is available from (14), we can proceed to the computation of the functions $\mathcal{V}_i(\varpi)$, $i = 0, 1, 2$, recalling that $\mathcal{V}_3(\varpi) = g(\varpi)$. As we are going to detail in Section 5.3, the order by which we compare the attributes plays also an important role in the final average cost. This is another notable difference as compared to the classical methodology where attribute ordering has absolutely no effect in the total cost. For our example we are going to use the attributes in the following order: sex, first name, surname.

For the consecutive computation of the functions $\mathcal{V}_2(\varpi), \mathcal{V}_1(\varpi), \mathcal{V}_0(\varpi)$ we are going to apply (15) and (16). We start with $\mathcal{V}_2(\varpi)$ and use the observation that $\mathcal{V}_3(\varpi) = g(\varpi)$. By applying (15) with $n = 2$ and using the parameters of the third attribute (surname): $c_3 = 8c$ and, from Table I, $p_2^{\mathcal{M}}(1) = 0.90$, $p_2^{\mathcal{U}}(1) = 0.05$, $p_2^{\mathcal{M}}(0) = 0.10$, $p_2^{\mathcal{U}}(0) = 0.95$, we can compute $\tilde{\mathcal{V}}_2(\varpi)$. This function is depicted in Figure 4 and, as we can see, it is also piece-wise linear. Using (16) we obtain $\mathcal{V}_2(\varpi)$. We repeat the same procedure for the computation of $\mathcal{V}_1(\varpi)$. Here we use the parameters of the second attribute (first name): $c_2 = 5c$, $p_1^{\mathcal{M}}(1) = 0.85$, $p_2^{\mathcal{U}}(1) = 0.10$, $p_2^{\mathcal{M}}(0) = 0.15$, $p_2^{\mathcal{U}}(0) = 0.90$ and apply (15) with $n = 1$ to find $\tilde{\mathcal{V}}_1(\varpi)$. This function, like the previous one, can be seen in Figure 4. Using (16) gives rise to $\mathcal{V}_1(\varpi)$. Finally, we need to apply once more the pair of equations (15), (16) in order to compute $\tilde{\mathcal{V}}_0(\varpi)$ and $\mathcal{V}_0(\varpi)$. Here we use $n = 0$ and the parameters of the first attribute (sex) $c_1 = c$, $p_3^{\mathcal{M}}(1) = 0.95$, $p_3^{\mathcal{U}}(1) = 0.45$, $p_3^{\mathcal{M}}(0) = 0.05$, $p_3^{\mathcal{U}}(0) = 0.55$.

In Figure 4 we also indicate the thresholds of each level. Besides the two standard 0,1 thresholds, we observe that $\tilde{\mathcal{V}}_0(\varpi)$ generates two additional ones: $\varpi_{0,1} = 0.069$ and $\varpi_{0,2} = 0.9653$ [the points of intersection with the function $g(\varpi)$]. If the prior probability p is smaller than the first threshold we decide in favor of “nonmatched” whereas if it is larger than the second in favor of “matched.” If p is between the two values then we continue with the first attribute (sex). Note that in this case there is no decision in favor of “clerical review.” It is clear that a prior outside the interval $(\varpi_{0,1}, \varpi_{0,2})$ corresponds to

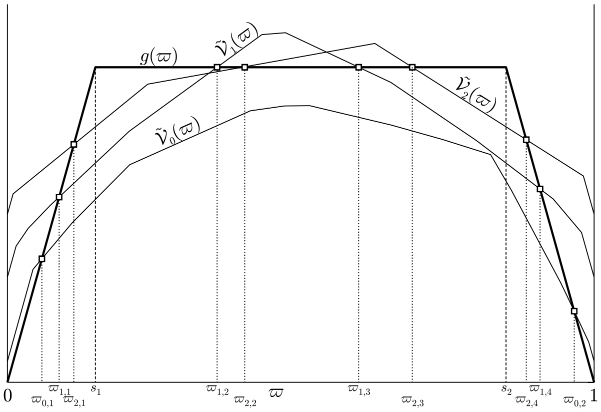

Fig. 4. Form of the functions $g(\varpi)$ and $\tilde{\mathcal{V}}_n(\varpi), n = 0, 1, 2$ and corresponding thresholds for the running example.

the case where we place more trust to the prior knowledge as opposed to any data coming from comparisons. Of course, in a normal situation it is advisable to avoid this situation by selecting the prior probability p inside this interval.

To either stop at the first level and decide or continue with the second attribute we need to use the thresholds coming from $\tilde{\mathcal{V}}_1(\varpi)$. As we can see we have four: $\varpi_{1,1} = 0.111$, $\varpi_{1,2} = 0.4$, $\varpi_{1,3} = 0.5285$, $\varpi_{1,4} = 0.8845$. Using the prior probability p and the outcome ξ_1 of the first attribute comparison, we compute the first posterior probability π_1 according to (6). If π_1 is in the interval $[0, \varpi_{1,1}]$ we stop and decide in favor of "nonmatched"; if it is in the interval $[\varpi_{1,2}, \varpi_{1,3}]$ we stop and decide in favor of "clerical review"; if it is in $[\varpi_{1,4}, 1]$ in favor of "matched." In all other cases, that is, if π_1 falls in $(\varpi_{1,1}, \varpi_{1,2})$ or in $(\varpi_{1,3}, \varpi_{1,4})$ we continue with the second attribute.

If we continue with the second attribute we obtain the comparison outcome ξ_2. We then use this information to form π_2 following (6). Now for this second level we have once more four thresholds $\varpi_{2,1} = 0.0881$, $\varpi_{2,2} = 0.3238$, $\varpi_{2,3} = 0.6353$, $\varpi_{2,4} = 0.9079$ due to the intersections of $\tilde{\mathcal{V}}_2(\varpi)$ with the function $g(\varpi)$. To decide we follow again the same procedure as in the previous case, that is, for $\pi_2 \in [0, \varpi_{21}]$ decision in favor of "nonmatched," for $\pi_2 \in [\varpi_{2,2}, \varpi_{2,3}]$ decision in favor of "clerical review," and for $\pi_2 \in [\varpi_{2,4}, 1]$ decision in favor of "matched." If $\pi_2 \in (\varpi_{2,1}, \varpi_{2,2}) \cup (\varpi_{2,3}, \varpi_{2,4})$ then we continue with the third attribute.

Using the third attribute leads to the generation of ξ_3 which we substitute in (6) to form π_3. Note that at this stage we have completely exhausted all attributes, therefore the only possibility left is to make a final decision. For this we use the two thresholds $s_1 = 0.15$, $s_2 = 0.85$ of the function $g(\varpi)$. If $\pi_3 \in [0, s_1]$ we decide in favor of "nonmatched," if $\pi_3 \in (s_1, s_2)$ in favor of "clerical review," and if $\pi_3 \in [s_2, 1]$ in favor of "matched." Of course the expectation is that in most of the cases we will reach decision in earlier stages thus reducing (in the average) the number of attribute comparisons.

5.1 Tree Representation of Optimum Test

A particularly appealing characteristic of our optimum scheme is our ability to interpret the whole decision mechanism with the help of a *decision tree*. If

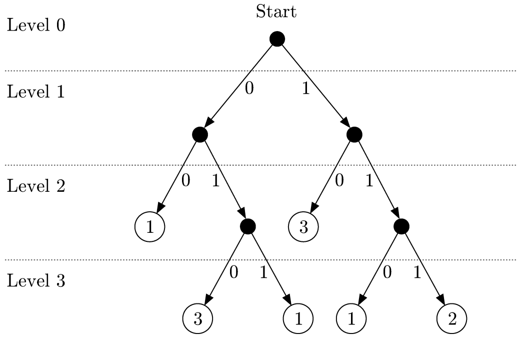

Fig. 5. Example of decision tree with $K = 3$ attributes and $L = 3$ selection possibilities.

the comparison outcomes ξ_n are binary then this tree turns out to be binary as well. Again, for simplicity we limit the presentation to this case.

An example of a tree is depicted in Figure 5 for $K = 3$ and $L = 3$. We distinguish $K + 1$ levels with the first, the 0th level, having a single starting node. All subsequent levels imply the use of the corresponding attribute. We distinguish two types of nodes: dark and white. Dark nodes correspond to continuation (to the next level) and therefore usage of the next attribute, while white nodes correspond to stopping and optimal selection. This means that white nodes are leaves. Each leaf contains the value obtained by applying the optimum selection strategy. Regarding now the dark nodes, since we are in the binary case, outcomes ξ_n can be equal to 0 or 1, therefore from a dark node with a 0 we move to the left and with a 1 to the right.

In a normal situation, the starting node must be dark. This happens when $g(p) > \tilde{\mathcal{V}}_0(p)$ where p is the prior probability $\mathbb{P}[H_{\mathcal{M}}]$ to have a match. If $g(p) \leq \tilde{\mathcal{V}}_0(p)$ then the tree collapses into a single white node with a given selection value in its interior, suggesting that we should *always* make the same selection without consulting any attributes.[4] Finally, we observe that in the last level there are only white nodes (leaves) since we have exhausted all our attributes and we are left with the selection process.

Let us now describe the tree generation mechanism. We recall that we must specify the *order* by which we compare the K attributes. Given the ordering, the attribute probabilities and costs, the prior probability, and the selection costs, we compute the functions $g(\varpi), \tilde{\mathcal{V}}_n(\varpi),\ n = 0, \dots, K-1$, exactly as described in the previous section. From $g(\varpi)$ we then compute the selection thresholds $s_j,\ \ j = 0, \dots, J-1$ and from each function $\tilde{\mathcal{V}}_n(\varpi)$ the thresholds $\varpi_{n,i}, i = 0, \dots, I_n - 1$ for stopping/continuation for the nth attribute. These quantities are needed for the generation of our tree.

During the generation process, to each node (white or dark) we assign a quantity that expresses the posterior probability. Figure 6 can be used to illustrate the generation mechanism. Notice that a dark node at level $n-1$ becomes the parent of two children at level n. If π_{n-1} is the posterior

[4] This case occurs when for example p is very close to 0 or 1, or when the attribute costs c_n are exceedingly higher than the selection costs C_{ji}.

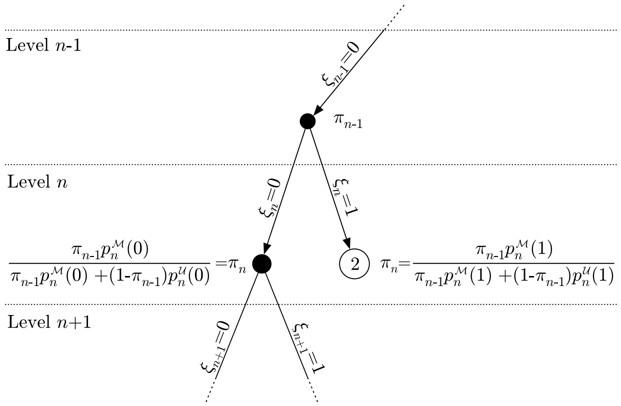
Level n-1
$\xi_{n-1}=0$
π_{n-1}
Level n
$\xi_n=0$
$\xi_n=1$
$\frac{\pi_{n-1}p_n^{\mathcal{M}}(0)}{\pi_{n-1}p_n^{\mathcal{M}}(0)+(1-\pi_{n-1})p_n^{\mathcal{U}}(0)}=\pi_n$
2
$\pi_n=\frac{\pi_{n-1}p_n^{\mathcal{M}}(1)}{\pi_{n-1}p_n^{\mathcal{M}}(1)+(1-\pi_{n-1})p_n^{\mathcal{U}}(1)}$
Level n+1
$\xi_{n+1}=0$
$\xi_{n+1}=1$

Fig. 6. Detail of decision tree generation.

probability assigned to this node, then by moving to the left with $\xi_n = 0$ we assign to the left child the corresponding probability π_n, which is computed using the indicated formula (coming from Lemma 2). A similar procedure holds for the right child. Now we must color the two children. In both cases we compare the computed posterior probability π_n with the thresholds $\varpi_{n,i}$ of this level, and decide whether we should stop or continue. If the decision is "continue" we color the node dark; if the decision is "stop" we color the node white. In the latter case we use π_n to make the optimum selection by comparing the posterior probability with the thresholds s_j. The selection is placed in the interior of the white child, which becomes a leaf. In the example depicted in Figure 6 the left child is colored dark; therefore we need to continue further from this node, whereas the right child's color is white and the optimum selection turned out to be 2. This child becomes a leaf.

We would like to emphasize that the computation of the functions $g(\varpi)$, $\tilde{\mathcal{V}}_n(\varpi)$, the thresholds $\varpi_{n,i}, s_j$, and the posterior probabilities π_n is required *only* in the tree generation phase, which is an offline task. When the actual record-linkage testing begins we simply use the tree (as it appears in Figure 5) *without* the need to reuse these quantities. Guided by the comparison outcomes ξ_n we follow a path in the tree until we reach a leaf which specifies the optimum selection and terminates the testing (between two records). Notice that the gain of our sequential scheme comes from the leaves that appear at intermediate levels (not requiring all attributes to be compared).

5.1.1 *Running Example (continued).* Keeping the same parameters defined previously in the running example and the results we obtained so far regarding the thresholds $\varpi_{i,j}$ and s_i, we can now apply the previous theory in order to build a binary tree representation of the decision mechanism. For this to be possible we need to define the prior probability p. We select for simplicity $p = 0.5$.

In Figure 7 we can see the resulting tree, the values of the posterior probabilities at each node, and the corresponding coloring with the necessary

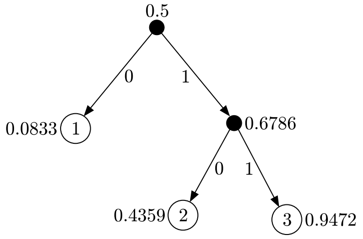

Fig. 7. Decision tree generation for running example.

decision when the color is white. Let us follow the coloring mechanism to understand this particular tree generation case. For the starting point we observe that $p = 0.5$ is between the two thresholds $\varpi_{0,1} = 0.0690$, $\varpi_{0,2} = 0.9653$, suggesting that we must continue with the first attribute. Consequently, the starting node will be colored black. Going to the left with $\xi_1 = 0$ and applying (6) produces the posterior probability $\pi_1 = 0.0833$, whereas going to the right with $\xi_1 = 1$ yields $\pi_1 = 0.6786$. We must now compare each value with the thresholds $\varpi_{1,i}$. Since $\varpi_{1,1} = 0.111$, we observe that $0, 0833 \in [0, \varpi_{1,1}]$, suggesting that here we must stop and decide in favor of 1 (nonmatched). Therefore this node is colored white and the corresponding decision is 1. In the opposite direction, since $\varpi_{1,3} = 0.5285$, $\varpi_{1,4} = 0.8845$, we note that $0.6786 \in (\varpi_{1,3}, \varpi_{1,4})$ which is a continuation region, therefore this node is colored black. Continuing from this last node, again we have two possibilities: left and right. Each direction results in a different posterior probability π_2 which is marked at the side of the corresponding node. These probabilities must be compared with the thresholds of the second level $\varpi_{2,i}$. By direct comparison we can immediately deduce that both cases lead to stopping and decision. The left decides in favor of 2 (clerical review) and the right in favor of 3 (matched).

It is interesting to note that the optimum scheme uses only the first two attributes and never the third. Furthermore the resulting tree has only 3 leaves, as opposed to 8 needed for the classical methodology.

5.2 Blocking Implementation of Optimum Test

The sequential scheme, the way it is presented, suggests a record-by-record testing. One might therefore envisage that there might be a possibility, using blocking type methodology, to outrun it. Next we will show that the existing blocking methods constitute in fact primitive forms of sequential testing, corresponding to an *alternative implementation* of a sequential test. Consequently, our test continues to enjoy the same optimality properties even in this seemingly different class of methodologies.

Instead of applying our test to each database record separately, we can follow an alternative idea. Let us initially apply the *first* stage of our test to *all* records in the database. In other words, compare the first attribute of the query record with the corresponding attribute of each record in the database. For each record this will produce an outcome ξ_n and allow for the computation of the corresponding posterior probability π_1. We can now categorize the records using their posterior probabilities. Each value π_1 either belongs to the stopping region or to the continuation region (defined by the thresholds $\varpi_{1,n}$). We can thus separate the records into two classes, one corresponding to continuation and the other to stopping. To the records that belong to the stopping class we apply the optimal selection strategy and decide among the L possibilities. These records are *excluded from any further testing*. In the remaining records (the ones belonging to the continuation class) we apply the *second* stage of our test. In other words, the second attribute of each record generates an outcome ξ_2 that is used to update the corresponding posterior probability π_2. We can now further categorize these records according to π_2 into stopping and continuation classes. In the first class we apply the optimal selection and then exclude this whole class from any further testing; in the second we continue with the third stage of our test, etc.

From the previous discussion, it becomes clear that it is impossible to find more efficient solutions inside the blocking methodology class than the one offered in this article. This is because blocking simply constitutes a different implementation of the record-by-record sequential scheme.

5.3 Optimum Attribute Ordering

The theory we developed in Section 2 and the useful tree and blocking representations of the optimum scheme detailed in the previous subsections are based on the assumption that we have a *prespecified* attribute ordering. In this subsection we address this final issue.

Given the attribute and selection costs and the prior and the attribute probabilities, the optimum average cost $\mathcal{V}_o(p)$ becomes a function of the *attribute ordering*. To make this clear, let us assume that we have $K = 4$ attributes enumerated as 1,2,3,4. Then the optimum average cost obtained by comparing the attributes in the order 1,2,3,4 is *different*[5] from the optimum average cost we obtain if the comparisons are performed in the order, say, 2,1,4,3. There are $K!$ possible permutations of the attributes and, with the theory developed in Section 2, to each permutation we can assign an optimum sequential test and its corresponding optimum average cost $\mathcal{V}_0(p)$. It is then clear that the ordering that provides the *minimum* average cost value $\mathcal{V}_0(p)$ is the optimum.

Unfortunately, it does not seem possible to perform ordering optimization efficiently, that is, without an exhaustive search through all $K!$ possibilities. Although in most database systems the number of attributes K is limited, examining all permutations might still be prohibitive from a computational point of view. This is because $K!$ increases dramatically with small increases in the value of K. For example, $7! = 5\,040$ while $10! = 3\,628\,800$ and $11! = 39\,916\,800$. Thus a case with 7 attributes might require acceptable computation time to examine all permutations, however, the same task becomes Herculean with 10 or 11 attributes.

[5] Such difference is nonexistent in nonsequential schemes because attribute ordering is immaterial when we first compare all attributes and then make a selection.

For cases where exhaustive search is not practically feasible we propose the following ad hoc rule.

Sort the attributes in increasing order according to the values $c_n[p_n^{\mathcal{M}}(0) + p_n^{\mathcal{U}}(1)]$**.**

Roughly speaking, we should defer usage of attributes that often make errors (expressed by $p_n^{\mathcal{M}}(0) + p_n^{\mathcal{U}}(1)$) and have high cost. This rule requires only order $K \log K$ computations (for sorting) and is therefore very efficient. Furthermore, although it is not guaranteed to yield the optimum ordering in all cases, it was observed through numerous simulations to provide average costs that were extremely close to the minimum. Of course, it is also possible to come up with more sophisticated techniques and improve this rule (by performing for example exhaustive search in subsets of attributes).

5.3.1 *Running Example (continued).* Let us apply the previous observations to the running example. An exhaustive search requires $3! = 6$ permutations of the attributes and computation of the optimum solutions of the corresponding optimal stopping problems. As before, we assume $p = 0.5$. By ranking the solutions according to their corresponding $\mathcal{V}_0(0.5)$ values, the permutation that produces the smallest average cost is the optimum. Using the theory we introduced so far to compute the average cost, we find that the optimum ordering is: sex, first name, surname, with a corresponding average cost: $\mathcal{V}_0(0.5) = 0.13165$.

Our simple ad hoc rule produces the following ordering: sex, surname, first name, with average cost $\mathcal{V}_0(0.5) = 0.13170$ which is extremely close to the optimum. In fact this permutation generates the second best average cost. We should mention that the ad hoc rule does not take into account the prior probability p, as opposed to the result of the exhaustive search which is a function of this parameter.

6. EXTENSIONS AND VARIANTS

There are different possibilities for extensions and variations. We present a characteristic example for each case.

6.1 General Comparison Outcomes

This extension is in fact immediate and requires almost no additional effort. This is because the presentation of our results was made under a general form not limited to the binary case. Let us assume that outcome ξ_n takes values inside a finite set Ξ_n. In other words, we assume that different attributes may produce outcomes with values in different sets. As it turns out the actual values of ξ_n are immaterial since they enter in the computation of the posterior probability only through the probabilities $p_n^{\mathcal{M}}(\xi_n)$, $p_n^{\mathcal{U}}(\xi_n)$. Therefore ξ_n can take any quantitative or qualitative value.

Theorems 1 and 2 that provide the optimum selection and stopping strategies continue to apply unaltered with the only difference being in (15) where the summation is over all possible values of $\xi_{n+1} \in \Xi_{n+1}$. The results of Section 5 are also valid, that is, functions $g(\varpi), \tilde{\mathcal{V}}_n(\varpi)$ are still piece-wise linear, concave, and continuous and for each stage we have thresholds $\varpi_{n,i}$ that define the stopping/continuation intervals.

The only significant difference, as compared to the binary case, occurs in the decision tree implementation of the optimum scheme. As expected, the tree is no longer binary. Indeed, at level $n-1$ we have now $|\Xi_n|$ edges emanating from each dark node, leading to an equal number of children. The posterior probability at each child node is still computed using Lemma 2 and its coloring is performed in exactly the same manner as in the binary case by comparing the posterior probability to the stage thresholds $\varpi_{n,i}$ and to s_j for optimal selection.

6.2 An Important Variant

The following problem defines an interesting variant of our original setup. Suppose that we would like to minimize the average selection cost, as in the nonsequential Bayesian approach [Verykios and Moustakides 2004]

$$\min_{\mathcal{N},\mathcal{D}_{\mathcal{N}}} \sum_{j=1}^{L} \sum_{i=\mathcal{M},\mathcal{U}} C_{ji}\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \,\&\, H_i], \tag{17}$$

but among all sequential strategies (which clearly include all nonsequential tests as a special case). Without imposing any other constraint, let us examine what is the test that minimizes (17). Notice now that since attribute usage has no cost, information accumulation is not only harmless but also helps the selection process if it is used wisely (optimally). It is therefore clear that *the test that minimizes (17) is the classical optimum nonsequential test that uses all attributes* [Verykios and Moustakides 2004]. There is no sequential test that can have better performance! It is only by taking into account the attribute costs that makes a sequential scheme preferable to the optimum nonsequential test.

As we noted earlier, the classical test is undesirable because of its need for all attributes. In order to limit this necessity, we can impose a constraint on the number of attributes to use. We can, for example, attempt to solve (17) under the constraint

$$\mathbb{E}[\mathcal{N}] \leq \mathcal{K}, \tag{18}$$

where $\mathcal{K}$ a constant defined by the user satisfying $1 \leq \mathcal{K} \leq K$ (the expected number of attributes cannot exceed the total number K and we would like to consult at least one attribute). In other words, we do not impose a hard constraint on $\mathcal{N}$ but rather, in the average, we would like it to be no larger than some quantity $\mathcal{K}$. This constraint diverts the optimum tests from the nonsequential class to the sequential one. Of course the crucial question here is: *How much do we lose in performance by constraining our scheme with (18)?* We defer the answer to this question until the next section.

Let us conclude this subsection by briefly presenting the solution to the constrained problem defined by (17) and (18). With the help of the Lagrange multiplier technique, the constrained problem can be reduced into an unconstrained, similar to the one solved in Section 4. Specifically, if $\alpha > 0$ denotes a

Lagrange multiplier, then we can define the average cost of the unconstrained problem

$$\mathcal{C}(\mathcal{N}, \mathcal{D}_{\mathcal{N}}, \alpha) = \alpha\mathbb{E}[\mathcal{N}] + \sum_{j=1}^{L} \sum_{i=\mathcal{M},\mathcal{U}} C_{ji}\mathbb{P}[\mathcal{D}_{\mathcal{N}} = j \,\&\, H_i] \tag{19}$$

which falls under the setup of Section 4 with $c_n = \alpha$. According to the Lagrange multiplier methodology, we must solve the problem assuming α is given. The resulting optimum stopping and decision strategies clearly become a function of this parameter. We must then select a specific Lagrange multiplier $\alpha = \alpha_\star$ for which the corresponding strategies *satisfy the constraint with equality*. One can then show that the stopping and selection rules we obtain for $\alpha = \alpha_\star$ also solve the original constrained optimization problem of (17) and (18).

7. EXPERIMENTS AND EVALUATION

In order to evaluate the proposed methodology, we have made use of the Record Linkage Software [Yancey and Winkler 2002] developed by the Statistical Research Division of the U.S. Bureau of the Census and has been provided to us by William E. Winkler. A general overview of the record-linkage process as it is performed by the Bureau of the Census is as follows. The process begins with two files which must be standardized using the address and name standardizer. The record matching system includes modules to standardize names, both business and personal, and street addresses. The standardized files must then be sorted according to the blocking variables, producing in this way a pair of sorted files. The matching program matches the two sorted files. After each pass through the matching program, record pairs are assigned either to a file of matched records or to a "residual" file of unmatched records that are not involved in the matched pairs. It is usually advisable to run the residual records through the matching program again after adjusting the matching score parameters and sorting and blocking on a different set of variables.

The matching program is distributed as a set of executable files that have been compiled for a windows machine. The program takes as input two files, each one containing 12 attributes. The first file contains 449 records and the second one 392. The input files need to be sorted and standardized. The layout of the input files is described in a parameter file that is provided to the matching program. A subset of the 12 attributes is used for blocking the file. The variables to be used for blocking are also described in the same parameter file, along with the prior probabilities of the 10 attributes to be used for matching. A program runs to compute the distribution of the comparison vectors produced by the linkage phase in the comparison space, so that the initial priors are adequately modified by using the EM algorithm to the data at hand. The matching program runs then and creates two sorted files based on the scores, one for positive scores and another one for negative scores. A script that runs next produces a collective output containing the full range of score values along with the pairs of records that generated these scores.

The 10 attributes to be used for comparison and their corresponding probabilities under the match and nonmatch hypotheses appear in the Table II. Our

Table II. Attributes and Attribute Probabilities for the Experimental Evaluation Example

Attribute	$p_n^{\mathcal{M}}(1)$	$p_n^{\mathcal{M}}(0)$	$p_n^{\mathcal{U}}(1)$	$p_n^{\mathcal{U}}(0)$
Last Name	0.9245	0.0755	0.2035	0.7965
First Name	0.7342	0.2658	0.0019	0.9981
Middle Name	0.7232	0.2768	0.0699	0.9301
Relation	0.5012	0.4988	0.1380	0.8620
Marital Status	0.9198	0.0802	0.3824	0.6176
Sex	0.9796	0.0204	0.4879	0.5121
Race	0.9856	0.0144	0.8618	0.1382
Age	0.8756	0.1244	0.1038	0.8962
House Number	0.9716	0.0284	0.1909	0.8091
Street	0.9018	0.0982	0.2856	0.7144

intention is to solve the variant introduced in Section 6.2. Following our ad hoc rule and since all attributes have common costs $c_n = \alpha$, we only need to order the attributes according to their attribute error probabilities $p_n^{\mathcal{M}}(0)+p_n^{\mathcal{U}}(1)$. This yields the following ordering: *House Number, Age, First Name, Last Name, Middle Name, Street, Marital Status, Sex, Relation, Race*.

We perform an initial blocking of the records in the input files by using two blocking variables, the cluster id in each record and the first character of the last name. This generated 68 blocks in both files. From the overall 68, only 59 blocks match, which means that 9 blocks from both files are not compared. This created a number of 50 records and 20 records from each file correspondingly that escape comparison. Based on the sizes of the matched blocks, we can compute an upper bound for the matched records. Assuming one-to-one matching (which means one record from the first file will match with at most one record from the second) we find, by summing the minima of the sizes of the matched blocks, that we can have at most 334 matches. By summing the products of the sizes of the matched blocks we compute the total number of record comparisons, which turns out to be 3703. These two figures can help us estimate the prior probability as $p = 334/3703 = 0.09$.

Regarding selection possibilities we consider the simple case of $L = 2$ selections (we decide between “Matched” and “Nonmatched”; there is no “Clerical review”). As average selection cost we use the selection error probability, which is obtained by defining $C_{00} = C_{11} = 0$ and $C_{01} = C_{10} = 1$. With this last definition we have specified all parameters of the problem and we are finally in a position to apply our theory.

We would like now to answer the question we deferred until this moment concerning the loss in performance as a result of the constraint in (18). In order to compare the sequential test against the classical that uses all attributes, we are going to solve the problem defined in (19) for different values of the Lagrange multiplier α. Each value of α produces an error probability and a corresponding average number of attributes. By varying α we can make a plot of the error probability as a function of the average number of attributes. We expect this curve to be decreasing since the more attributes we use, the smaller the selection error probability becomes. Furthermore, as the average number of attributes approaches the maximum number K, the selection error

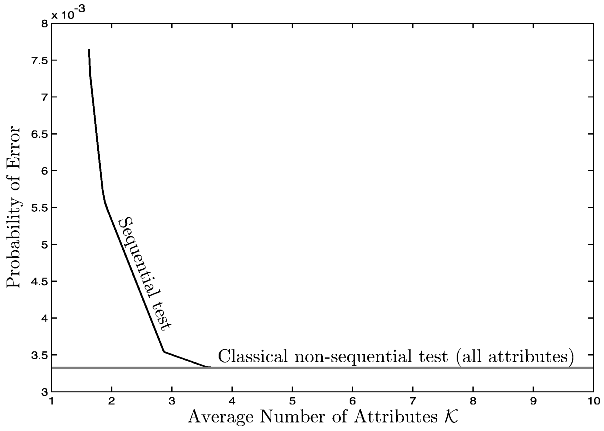

Fig. 8. Probability of error as a function of the expected number of attributes.

probability is expected to approach the error probability floor imposed by the classical nonsequential test that uses all K attributes.

Figure 8 depicts the graph we just described. In the same figure we also display, with a horizontal line, the error floor attained by the classical test. The latter has the value 3.324×10^{-3}. As we emphasized in the previous subsection, this floor cannot be surpassed by any sequential scheme. We observe that the sequential test, when the average number of attributes $\mathcal{K}$ is low (below 2), it exhibits an error probability which is an order of magnitude larger than the classical test. However, performance improves rapidly as we increase the average number. Indeed, at $\mathcal{K} = 2.5$ attributes it reaches the same order of magnitude, whereas at $\mathcal{K} = 3.5$ it becomes practically indistinguishable from the classical test. *The fact that we attain the same minimum error level but at a significantly lower (average) attribute usage is what makes the sequential methodology attractive.* Indeed, in this example by practically making no sacrifice in error performance we can cut down the attribute usage to one-third. If we are willing to make a slightly more significant sacrifice, say raise the error probability from 3.3×10^{-3} to 5.7×10^{-3}, then we can attain this value with an average of just 1.85 attributes instead of 10.

The previous findings constitute theoretical predictions on our scheme's performance, based on the analysis developed in the previous sections. But does the practical application of the sequential test corroborate these analytical figures? To answer this question, we generated the thresholds for the sequential test for an expected number of attributes $\mathcal{K} = 3.6$. Theory predicts that this test has an error equal to 3.333×10^{-3} which is extremely close to the floor value 3.324×10^{-3}. In other words, we expect the sequential and the classical test to have the same performance. Indeed, by running the two tests on the 3703 record comparisons, they both produced 24 errors. We also generated thresholds for the sequential test for an expected number of attributes $\mathcal{K} = 1.85$, which corresponds to a theoretical error probability of 5.7×10^{-3}. We ran the sequential test and the number of errors increased to 65.

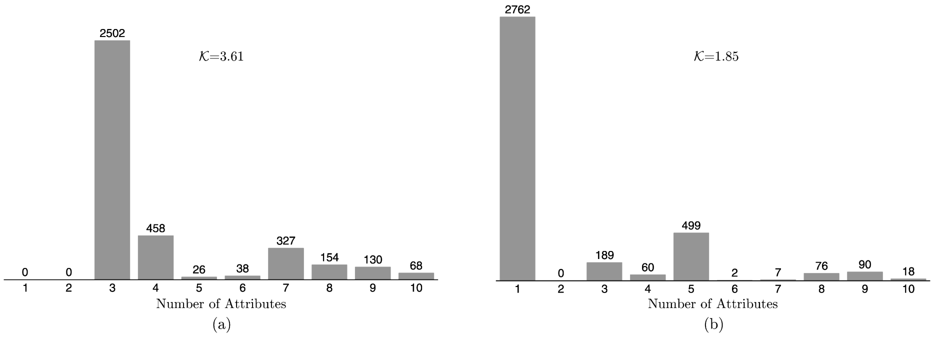

Fig. 9. Frequency of number of attributes needed by the sequential test to reach a final decision.

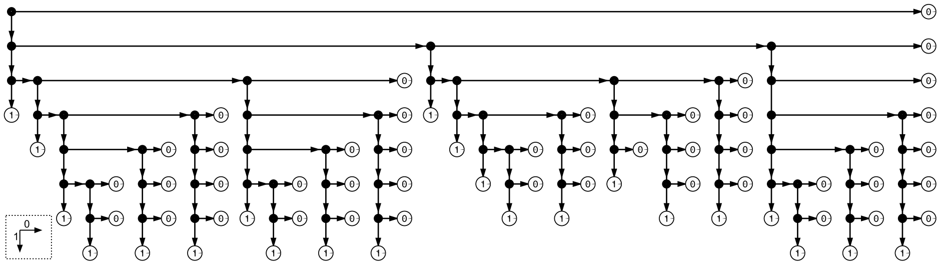

Fig. 10. Binary decision tree implementation of the optimum sequential test for $\mathcal{K}$ = 1.85.

In Figure 9, using bar-charts, we show how often the sequential test needs 1,2,. . . ,10 attributes to reach a final decision. Subfigure (a) refers to $\mathcal{K}$ = 3.61 and (b) to $\mathcal{K}$ = 1.85. The equivalent bar-chart graph for the classical test is a single bar at 10, with frequency 3703. In (a) most of the final decisions (2502) are reached by using the first 3 attributes. We can also see that the sequential test resorted only 68 times to the complete set of attributes. If we allow ourselves the benefit of making more errors (65 instead of 24) then, in case (b), most of the decisions are reached with a *single* attribute (these are "nonmatched" cases which constitute the majority of the record comparisons). Here the test used all attributes only 18 times.

Figure 10 depicts the binary decision tree representation of the optimum sequential test for the case $\mathcal{K}$ = 1.85. We have selected this case because its binary tree is much simpler (and consequently more legible) than the corresponding tree for $\mathcal{K}$ = 3.61. We have slightly changed the tree layout as compared to Section 5.1 to accommodate the limitations imposed by the article format. Instead of going left-right, now with ξ_n = 1 we go down while with ξ_n = 0 we go right. A white leaf with a 1 suggests a decision in favor of "matched" and with a 0 in favor of "nonmatched." We observe the top horizontal edge which corresponds to a decision in favor of "nonmatched" every time the first attribute outputs a 0. According to Figure 9(b) this happens 2762 times. There is also the left most vertical line which is comprised of three consecutive edges. This suggests that whenever the first three attributes output three consecutive 1s we decide in favor of a "matched." Again as we can see from Figure 9(b) this happens 189 times (there is no other combination of the first 3 attributes that leads to a decision). It is also worth counting the total number of leaves in the tree, which is 66. This means that there are only 66 different attribute combinations that arise in the sequential decision mechanism as compared to the $2^{10} = 1024$ that can be encountered in the classical test.

Let us now examine whether the simulation results are consistent with the theoretical findings regarding probability of error and average number of attributes. In Figure 9, by multiplying the frequencies with the corresponding number of attributes and then adding the products, we compute the total number of attribute comparisons used by the sequential test to complete all 3703 cases. In (a) this number is $15,067$ and in (b) $7,723$. The corresponding number for the classical test is $3703 \times 10 = 37,030$. Consequently, we have a 59.3% reduction in number of attribute comparisons in (a) and 79.2% in (b). We can also compute the arithmetic average number of attributes used by the sequential test. We have $\hat{\mathcal{K}} = 15,067/3703 = 4.07$ for (a) and $\hat{\mathcal{K}} = 7,723/3703 = 2.09$ for (b) which must be compared with the theoretical values $\mathcal{K} = 3.61$ and 1.85, respectively. We note a relatively close agreement.

By forming the ratio $P_e = 24/3703 = 6.48 \times 10^{-3}$ we compute the *frequency of errors* which should normally agree with the theoretically determined error probability 3.324×10^{-3}. As we can see, there is a significant difference. We recall, however, that P_e is simply an *estimate* for the actual error probability and since estimates are random variables, P_e has a variance. One can show that with 3703 record comparisons P_e can differ from the theoretical probability[6] even by 100%. In fact, we need 10 times more record comparisons in order for P_e to start being close to the error probability. The same reasoning when applied to the arithmetic average number of attributes $\hat{\mathcal{K}}$ results in a qualitatively different conclusion. Specifically, with 3703 record comparisons, statistical analysis shows that $\hat{\mathcal{K}}$ should deviate from $\mathcal{K}$ by a (practically) *tolerable* 10%. The two computed $\hat{\mathcal{K}}$ values are indeed consistent with this claim since they differ from their theoretical counterparts by approximately this percentage.

8. CONCLUSION

We have presented a sequential test for solving the record-linkage problem. Instead of first comparing all attributes and then making a decision as is the case in the existing methodology, we propose the gradual assessment of attributes and the possibility to stop and decide at any intermediate step. Based on a well-defined performance measure, we were able to optimize the proposed decision scheme by finding interesting tree-like and blocking type implementations. The strong point of the sequential methodology lies in the fact that we can achieve practically the same performance as the classical test, but by using, on average, a significantly lower number of attributes.

[6] We basically refer to the 99% confidence interval.

REFERENCES

ANANTHAKRISHNA, R., CHAUDHURI, S., AND GANTI, V. 2002. Eliminating fuzzy duplicates in data warehouses. In *Proceedings of the 28th International Conference on Very Large Databases (VLDB'02)*.

BHATTACHARYA, I. AND GETOOR, L. 2005. Entity resolution in graph data. Tech. rep. CS-TR-4758, Computer Science Department, University of Maryland.

BILENKO, M., MOONEY, R. J., COHEN, W. W., RAVIKUMAR, P., AND FIENBERG, S. E. 2003. Adaptive name matching in information integration. *IEEE Intell. Syst. 18*, 5, 16–23.

CHAUDHURI, S., GANJAM, K., GANTI, V., AND MOTWANI, R. 2003. Robust and efficient fuzzy match for online data cleaning. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'03)*. 313–324.

COHEN, W. W. 2000. Data integration using similarity joins and a word-based information representation language. *ACM Trans. Inform. Syst. 18*, 3, 288–321.

COHEN, W. W. AND RICHMAN, J. 2002. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*.

DEMPSTER, A. P., LAIRD, N. M., AND RUBIN, D. B. 1977. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statist. Soc. B*, 39, 1–38.

ELMAGARMID, A. K., IPEIROTIS, P. G., AND VERYKIOS, V. S. 2007. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng. 19*, 1, 1–16.

FELLEGI, I. P. AND SUNTER, A. B. 1969. A theory for record linkage. *J. Amer. Statist. Assoc. 64*, 328, 1183–1210.

GALHARDAS, H., FLORESCU, D., SHASHA, D., SIMON, E., AND SAITA, C. A. 2001. Declarative data cleaning: Language, model, and algorithms. In *Proceedings of the 27th International Conference on Very Large Databases (VLDB'01)*. 371–380.

GUHA, S., KOUDAS, N., MARATHE, A., AND SRIVASTAVA, D. 2004. Merging the results of approximate match operations. In *Proceedings of the 30th International Conference on Very Large Databases (VLDB'04)*. 636–647.

HERNÁNDEZ, M. A. AND STOLFO, S. J. 1995. The merge/purge problem for large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'95)*. 127–138.

JOACHIMS, T. 1999. Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, Eds. MIT Press.

MCCALLUM, A. AND WELLNER, B. 2004. Conditional models of identity uncertainty with application to noun coreference. In *Proceedings of the Conference on Advances in Neural Information Processing Systems (NIPS'04)*.

MONGE, A. E. AND ELKAN, C. P. 1996. The field matching problem: Algorithms and applications. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*. 267–270.

NEWCOMBE, H. B. AND KENNEDY, J. M. 1962. Record linkage: Making maximum use of the discriminating power of identifying information. *Comm. ACM 5*, 11, 563–566.

NEWCOMBE, H. B., KENNEDY, J. M., AXFORD, S., AND JAMES, A. 1959. Automatic linkage of vital records. *Sci. 130*, 3381, 954–959.

RAVIKUMAR, P. AND COHEN, W. W. 2004. A hierarchical graphical model for record linkage. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI'04)*.

SARAWAGI, S. AND BHAMIDIPATY, A. 2002. Interactive deduplication using active learning. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*. 269–278.

SHIRYAYEV, A. 1978. *Optimal Stopping Rules*. Springer.

SINGLA, P. AND DOMINGOS, P. 2004. Multi-Relational record linkage. In *Proceedings of the ACM SIKKDD International Conference on Knowledge Discovery 2nd Data Mining (KDD'04): Workshop on Multi-Relational Data Mining*. 31–48.

TEJADA, S., KNOBLOCK, C. A., AND MINTON, S. 2002. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*.

TEPPING, B. J. 1968. A model for optimum linkage of records. *J. Amer. Statist. Assoc. 63*, 324, 1321–1332.

VERYKIOS, V. S. AND MOUSTAKIDES, G. V. 2004. A generalized cost optimal decision model for record matching. In *Proceedings of the International Workshop on Information Quality in Information Systems*. 20–26.

VERYKIOS, V. S., MOUSTAKIDES, G. V., AND ELFEKY, M. G. 2003. A bayesian decision model for cost optimal record matching. *VLDB J. 12*, 1, 28–40.

WINKLER, W. E. 1993. Improved decision rules in the felligi-sunter model of record linkage. Tech. rep., Statistical Research Report Series RR93/12, U.S. Bureau of the Census, Washington, D.C.

WINKLER, W. E. 1995. Matching and record linkage. In *Business Survey Methods*. Wiley, 355–384.

YANCEY, W. E. AND WINKLER, W. E. 2002. Record linkage software. User documentation. Tech. rep., Statistical Research Report Series, U.S. Bureau of the Census, Washington, D.C.

Received June 2007; revised September 2008; accepted September 2008