algorithm to obtain
he two components
he SEM algorithm
ndard errors of the

hods for estimating
ssly optimistic."
thods of Belin and
tions where (1) the
s, U, are somewhat
re of the conditional
d Rubin method is
d via a 1–1 matching
ists, and agriculture
ist weighting curves
ethods of Belin and
ing curves associated

ferent lists, or
ls or learn to override

gave the main decision
nd non-matches; gave
are computed under a
typographical error can
rs, we will more fully
bilities) in practice, the
r parameter estimation
coded fields to attempt
cking methods to bring
where there are large
A and B, and methods
narily very minor ones)

9 Estimating the Parameters of the Fellegi–Sunter Record Linkage Model

In this chapter, we discuss several schemes for estimating the parameters (i.e., the m-and u-probabilities) of the Fellegi–Sunter model discussed in Chapter 8.

9.1. Basic Estimation of Parameters Under Simple Agreement/Disagreement Patterns

For each $\gamma \in \Gamma$, Fellegi and Sunter [1969] considered

$$P(\gamma) = P(\gamma \mid r \in M)P(r \in M) + P(\gamma \mid r \in U)P(r \in U)$$

and noted that the proportion of record pairs, r, having each possible agreement/disagreement pattern, $\gamma \in \Gamma$, could be computed directly from the available data. For example, if $\gamma = (\gamma_1, \gamma_2, \gamma_3)$ consists of a simple agree/disagree (zero/one) pattern associated with three variables, then a typical value for γ would be (1, 1, 0). Then, by our usual conditional independence assumption, there exist vector constants (marginal probabilities) $m = (m_1, m_2, \ldots, m_n)$ and $u = (u_1, u_2, \ldots, u_n)$ such that, for all 2^n possible values of$(\gamma_1, \gamma_2, \ldots, \gamma_n)$

$$P[(\gamma_1, \gamma_2, \ldots, \gamma_n)\,|\,r \in M] = \prod_{i=1}^{n} m_i^{\gamma_i}(1 - m_i)^{1-\gamma_i}$$

and

$$P[(\gamma_1, \gamma_2, \ldots, \gamma_n)\,|\,r \in U] = \prod_{i=1}^{n} u_i^{\gamma_i}(1 - u_i)^{1-\gamma_i}$$

For the case in which $n \geq 3$, Fellegi and Sunter [1969] showed how to use the equations above to find solutions for the $2n + 1$ independent parameters – $m_1, m_2, \ldots, m_n, u_1, u_2, \ldots, u_n$, and *P[M]*. (We obtain *P[U]* as $P[U] = 1 - P[M]$.) The reader can obtain further details from Fellegi and Sunter [1969].

9.2. Parameter Estimates Obtained via Frequency-Based Matching[1]

If the distribution of the attribute values for a field is not uniform, then a value-specific (frequency-based), or outcome-specific, weight can be introduced. Frequency-based weights are useful because they can account for the fact that a specific pair of surnames such as (Zabrinsky, Zabrinsky) occurs less often in the United States than a pair of surnames such as (Smith, Smith). This is useful because names such as Smith and Jones that are common in the general population of the United States may not be as effective as a relatively rare name such as Zabrinsky in distinguishing matches.

Moreover, surnames such as Martinez and Garcia have more distinguishing power in Minneapolis than in Los Angeles because Hispanic surnames are so much more common in Los Angeles.

How can such phenomena be incorporated into our Fellegi–Sunter models? The answer is by modifying the model to give a larger agreement weight to infrequently occurring surnames. This can be justified further by noting that if γ represents the surnames on a pair of records, r, then

$$P(r \in M \mid \textit{agreement on Zabrinsky}) > P(r \in M \mid \textit{agreement on Smith})$$

which implies that

$$P(r \in U \mid \textit{agreement on Smith}) > P(r \in U \mid \textit{agreement on Zabrinsky})\,.\,.$$

Using the last two inequalities in conjunction with Bayes' Theorem, we can show that

$$\frac{P(\textit{agreement on Zabrinsky} \mid r \in M)}{P(\textit{agreement on Zabrinsky} \mid r \in U)} > \frac{P(\textit{agreement on Smith} \mid r \in M)}{P(\textit{agreement on Smith} \mid r \in U)}.$$

Fellegi and Sunter [1969, pp. 1192–1194] propose a solution to this problem. To simplify their ideas, we begin by assuming that neither file A nor file B contains any duplicate records. We also assume that the true frequencies of specific fields (e.g., the surname of an individual) in files A and B, respectively, are

$$f_1, f_2, \ldots, f_m$$

where the number of records in file A is $N_A = \sum_{j=1}^{m} f_j$, and

$$g_1, g_2, \ldots, g_m$$

[1] For further details on this topic, the interested reader should see Section 3.3.1 of Fellegi and Sunter [1969] or Winkler [1989b].

ıed via
1

for a field is not uniform, then a
e-specific, weight can be introduced.
e they can account for the fact that
insky, Zabrinsky) occurs less often
ıes such as (Smith, Smith). This is
ınes that are common in the general
as effective as a relatively rare name
s.
nd Garcia have more distinguishing
s because Hispanic surnames are so

ted into our Fellegi–Sunter models?
give a larger agreement weight to
be justified further by noting that if
ırds, r, then

$P(r \in M \mid agreement\ on\ Smith)$

$\in U \mid agreement\ on\ Zabrinsky)$.

ction with Bayes' Theorem, we can

$$\frac{P(agreement\ on\ Smith \mid r \in M)}{P(agreement\ on\ Smith \mid r \in U)}.$$

propose a solution to this problem. To
; that neither file A nor file B contains
t the true frequencies of specific fields
s A and B, respectively, are

$., f_m$

$V_A = \sum_{j=1}^{m} f_j$, and

$., g_m$

l reader should see Section 3.3.1 of Fellegi

where the number of records in file B is $N_B = \sum_{j=1}^{m} g_j$. If the mth surname or string, say "Smith", occurs f_m times in file A and g_m times in file B, then "Smith" occurs $f_m g_m$ times in the pairs of records that constitute $A \times B$. The corresponding true frequencies in the set of matching pairs, M, are similarly assumed to be

$$h_1, h_2, \ldots, h_m$$

where the number of records in file M is $N_M = \sum_{j=1}^{m} h_j$. We note that for $j = 1, 2, \ldots, m$, we have $h_j \leq \min(f_j, g_j)$. For some applications, we assume that

- $h_j = \min(f_j, g_j)$,
- $P[agreement\ on\ string\ j \mid r \in M] = \frac{h_j}{N_M}$, and
- $P[agreement\ on\ string\ j \mid r \in U] = \frac{f_j \cdot g_j - h_j}{N_A \cdot N_B - N_M}$.

In practice, we must use observed or synthetic values in the actual files being matched because we have no way of knowing the true values. Practitioners have used a variety of schemes to construct the frequencies, h_j. These depend upon how typographical errors are modeled and what simplifying assumptions are made. (For example, a surname such as "Smith" might be recorded as "Snith" due to a typographical error.) The typographical errors then distort the frequency counts in the observed files. Fellegi and Holt [1969] presented a method for dealing with some typographical errors.

In order to obtain numerically stable estimates of the simple agree/disagree probabilities, we use the EM algorithm

Thibaudeau [1989] and Winkler [1989c, 1992] have used the EM algorithm in a variety of record linkage situations. In each, it converged rapidly to unique limiting solutions over different starting points. The major difficulty with the EM algorithm or similar procedures is that it may produce solutions that partition $A \times B$ into two sets that differ substantially from the desired sets of true matches, M, and true non-matches, U. We describe the specifics of the EM algorithm in Section 9.4, later in this chapter. Winkler [1989b] provided a slight generalization of the Fellegi–Sunter method for frequency-based matching. Specifically, Winkler showed that a "surrogate" typographical error rate could be computed using the EM algorithm and that the frequency-based weights could be "scaled" to the EM-based weights. The reader should see Winkler's paper for more details.

9.2.1. *Data from Prior Studies*

In some applications, the frequency tables are created "on-the-fly" – i.e., from the data of the current application. Alternatively, we could create them using a large reference file of names constructed from one or more previous studies. The advantage of "on-the-fly" tables is that they can use different relative frequencies in different geographic regions – for instance, Hispanic surnames in Los Angeles, Houston, or Miami and French surnames in Montreal. The disadvantage of "on-the-fly" tables is that they must be based on data files that cover a large percentage of the target population. If the data files contain samples from a population, then the frequency weights should reflect the appropriate population frequencies. For instance, if two lists of companies in a city are used and "George Jones, Inc." occurs once on each list, then a pair should not be designated as a match using name information only. Corroborating information, such as the business's address, should also be used because the name "George Jones, Inc." may not uniquely identify the establishment. Moreover, the population should be the population of interest to the list-builder; for example, if The National Council of La Raza is building a list of registered Hispanic voters in California, then the frequencies should represent the Hispanic subpopulation of California rather than the frequencies in the general population.

This use of frequency tables in conjunction with the Fellegi–Sunter model has often worked well in practice, especially when the typographical error rate is constant over all of the possible values of the surname. Moreover, we note that frequency-based matching could be applied to other fields in the database besides surname. While this value-specific approach can be used for any matching element, strong assumptions must be made about the independence between agreements on specific values of one element versus agreement on other fields.

Three other circumstances can adversely affect this frequency-based matching modification:

- The two files A and B are samples.
- The two files A and B do not have much overlap.
- The corresponding fields in files A and B have high rates of typographical error.

Example 9.1: Matching marriage and birth records in British Columbia

When matching records on marriages and births that occurred in British Columbia during 1946–1955, Newcombe et al. [1959] considered as one factor the frequencies of the husbands' and brides' last names. A positive value (or weight) was added to the score if the names were in agreement in the two databases. The value added depended on how rare or common the last names were in the file as a whole. Other data fields, such as age and place of birth, were treated similarly.

9.3. Parameter Estimates Obtained Using Data from Current Files

Before discussing the EM algorithm, we describe an alternative scheme for computing the u- and m-probabilities "on-the-fly."

Montreal. The disadvantage of
on data files that cover a large
ata files contain samples from a
reflect the appropriate population
ies in a city are used and "George
pair should not be designated as
orating information, such as the
se the name "George Jones, Inc."
Moreover, the population should
er; for example, if The National
red Hispanic voters in California,
panic subpopulation of California
lation.
n with the Fellegi–Sunter model
ly when the typographical error
es of the surname. Moreover, we
e applied to other fields in the
pecific approach can be used for
st be made about the independence
ne element versus agreement on

ect this frequency-based matching

verlap.
have high rates of typographical

cords in British Columbia

s that occurred in British Columbia
9] considered as one factor the
ames. A positive value (or weight)
a agreement in the two databases.
ommon the last names were in the
ge and place of birth, were treated

ed Using Data

lescribe an alternative scheme for
e-fly."

9.3.1. u-Probabilities

Assume that File A has 1,000 records and that File B also has 1,000 records. Then, at most 1,000 of the 1,000,000 record pairs in $A \times B = M \cup U$ can be matches. This suggests that the u-probabilities can be reasonably approximated by random agreement weights. We can approximate$P\,[\textit{agree on first name} \mid r \in U]$, for example, by counting the number of pairs in $A \times B$ that agree on the first name and dividing the result by the number of pairs in $A \times B$ because almost all of the pairs in $A \times B$ are in U.

Alternatively, we may only look at certain subsets of $A \times B$. For example, we might only consider pairs whose Zip Codes are identical. This approach, known as *blocking*[2], usually leads to a substantial reduction in the number of pairs that need to be considered. When blocking is used, the u-probabilities can be computed as

$$P\,[\textit{agree on first name} \mid r \in U \textit{ and there is agreement on the blockings criteria}].$$

Fellegi and Sunter [1969] provide details on the adjustments that need to be made to the matching rules when blocking is used.

9.3.2. m-Probabilities

Assume that we have used the initial guess of m-probabilities given in the previous section and performed matching. We could draw a sample of pairs, perform follow-up (manual review, etc.) to determine true match status, and re-estimate the m-probabilities based on the sample for which we know the truth. Newcombe [1959] suggested this type of iterative-refinement procedure. It typically yields good estimates of the m-probabilities that work well in the matching decision rules. Winkler [1990] compared the m-probabilities obtained via the iterative-refinement procedure with m-probabilities obtained via several other methods. Surprisingly, with certain high-quality files, EM-based estimates of m-probabilities that do not require any training data or follow-up outperformed the m-probabilities obtained via the iterative refinement procedure.

The iterative-refinement methodology for estimating the m- and u-probabilities is known to work well in practice and is used in several commercial software packages. It can work well with messy, inconsistent data. This type of data is typically encountered in business lists, agriculture lists, and some administrative lists.

9.4. Parameter Estimates Obtained via the EM Algorithm

We begin by assuming that the cross-product space $A \times B = M \cup U$ consists of N pairs of records where each record has n fields. We define

$$\gamma_i^j = \begin{cases} 1 & \textit{if field } i \textit{ is identical on both of the records of record pair } r_j \\ 0 & \textit{otherwise} \end{cases}$$

[2] Blocking is the subject of Chapter 12.

where i = 1, 2, . . . , n and j = 1, 2, . . . , N. We also define

$$\gamma^j = \left\{\gamma_1^j, \gamma_2^j, \ldots, \gamma_n^j\right\}$$

and

$$\gamma = \left\{\gamma^1, \gamma^2, \ldots, \gamma^N\right\}.$$

Furthermore, we define the components of the m- and u-probabilities

$$m = \{m_1, m_2, \ldots, m_n\} \text{ and } u = \{u_1, u_2, \ldots, u_n\}$$

as

$$m_i = P\left[\gamma_i^j = 1 \mid r_j \in M\right] \text{ and } u_i = P\left[\gamma_i^j = 1 \mid r_j \in U\right]$$

for a randomly selected pair of records r_j.

We next define p as the proportion of record pairs that match:

$$p = \frac{Number\ of\ Pairs\ of\ Records\ in\ set\ M}{N}.$$

The *N* record pairs of interest are distributed according to a finite mixture model with the unknown parameters $\Phi = (m, u, p)$. The goal is to use an EM algorithm to estimate these unknown parameters, especially the vector m.

We next consider the complete data vector $\langle g, \gamma \rangle$ where $g = (g_1, g_2, \ldots, g_N)$ is a vector of indicator functions with

$$g_j = \begin{cases} 1 & \text{if } r_j \in M \\ 0 & \text{if } r_j \in U. \end{cases}$$

Then the complete data likelihood function is

$$f(g, \gamma \mid m, u, p) = \prod_{j=1}^{N} \left(p \cdot P\left[\gamma^j \mid r_j \in M\right]\right)^{g_j} \left((1-p)P\left[\gamma^j \mid r_j \in U\right]\right)^{1-g_j},$$

and the complete data log-likelihood is

$$\ln\left(f(g, \gamma \mid m, u, p)\right) = \sum_{j=1}^{N} g_j \cdot \ln\left(p \cdot P[\gamma^j \mid r_j \in M]\right) + \sum_{j=1}^{N} (1-g_j) \cdot \ln\left((1-p) \cdot P\left[\gamma^j \mid r_j \in U\right]\right).$$

We assume (e.g., by invoking the conditional independence assumption) that

$$P[\gamma^j \mid r_j \in M] = \prod_{i=1}^{n} m_i^{\gamma_i^j} (1-m_i)^{1-\gamma_i^j}$$

and

The first ste
initial estima
previous simi
previous mat
not particular
guessed. Jaro
the correspon
for each ~~com~~

Alternative
estimates.

The imple
implementati
(M) step unt
of precision.

For the ex

ĝ

For the m
maximizatic
Setting the
and solving

and

where f (γ
Finally, th
matched p

We also define

$$, \ldots, \gamma_n^j \}$$

$$\ldots, \gamma^N \}.$$

ɔf the m- and u-probabilities

$$\text{ɪd } u = \{u_1, u_2, \ldots, u_n\}$$

$$\text{ɑd } u_i = P\lfloor \gamma_i^j = 1 \mid r_j \in U \rfloor$$

j.

record pairs that match:

$$\frac{\textit{of Records in set M}}{N}.$$

uted according to a finite mixture model ., p). The goal is to use an EM algorithm ɜspecially the vector m. vector $\langle g, \gamma \rangle$ where $g = (g_1, g_2, \ldots, g_N)$

$$\text{if } r_j \in M$$
$$\text{if } r_j \in U.$$

tion is

$$\in M])^{g_j} \left((1-p)P\left[\gamma^j \mid r_j \in U\right]\right)^{1-g_j},$$

;

$$\in M]) + \sum_{j=1}^{N} (1-g_j) \cdot \ln\left((1-p) \cdot P\left[\gamma^j \mid r_j \in U\right]\right).$$

ɪditional independence assumption) that

$$: \prod_{i=1}^{n} m_i^{\gamma_i^j} (1-m_i)^{1-\gamma_i^j}$$

and

$$P[\gamma^j \mid r_j \in U] = \prod_{i=1}^{n} u_i^{\gamma_i^j} (1-u_i)^{1-\gamma_i^j}.$$

The first step in the implementation of the EM algorithm is to compute initial estimates of the unknown parameters m, u, and p. An analyst with previous similar experience might begin by using the parameters from a similar, previous matching project as the initial parameter estimates. The algorithm is not particularly sensitive to starting values and the initial estimates of m can be guessed. Jaro [1989] states that the initial estimates of m should be greater than the corresponding initial estimates of u. Jaro [1989] used an initial estimate of .9 for each component of m in his Tampa study.

Alternatively, the analyst might use the EM algorithm itself to obtain initial estimates.

The implementation of the EM algorithm from here on involves repeated implementations of the expectation (E) step, followed in turn by the maximization (M) step until the algorithm has produced estimates that attain the desired level of precision.

For the expectation step, we replace the indicator function g_j by $\hat{g}_j$ where

$$\hat{g}_j = \frac{\hat{p} \prod_{i=1}^{n} \hat{m}_i^{\gamma_i^j} (1-\hat{m}_i)^{1-\gamma_i^j}}{\hat{p} \prod_{i=1}^{n} \hat{m}_i^{\gamma_i^j} (1-\hat{m}_i)^{1-\gamma_i^j} + (1-\hat{p}) \prod_{i=1}^{n} \hat{u}_i^{\gamma_i^j} (1-\hat{u}_i)^{1-\gamma_i^j}}.$$

For the maximization step, we partition the problem into three distinct maximization problems: one for p, and one for each of the vectors m and u. Setting the partial derivatives of the complete data log-likelihood equal to zero and solving for $\hat{m}_i$ and $\hat{u}_i$, we obtain, respectively,

$$\hat{m}_i = \frac{\sum_{j=1}^{N} \hat{g}_j \cdot \gamma_i^j}{\sum_{j=1}^{N} \hat{g}_j} = \frac{\sum_{j=1}^{2^n} \hat{g}_j \cdot \gamma_i^j \cdot f\left(\gamma^j\right)}{\sum_{j=1}^{2^n} \hat{g}_j \cdot f\left(\gamma^j\right)}$$

and

$$\hat{u}_i = \frac{\sum_{j=1}^{N} \left(1-\hat{g}_j\right) \cdot \gamma_i^j}{\sum_{j=1}^{N} \left(1-\hat{g}_j\right)} = \frac{\sum_{j=1}^{2^n} \left(1-\hat{g}_j\right) \cdot \gamma_i^j \cdot f\left(\gamma^j\right)}{\sum_{j=1}^{2^n} \left(1-\hat{g}_j\right) \cdot f\left(\gamma^j\right)}$$

where $f\left(\gamma^j\right)$ is the number of times the pattern γ^j occurs in the N pairs of records. Finally, the solution for p results in the following estimate of the proportion of matched pairs:

$$\hat{p} = \frac{\sum_{j=1}^{N} \hat{g}_j}{N} = \frac{\sum_{j=1}^{2^n} \hat{g}_j \cdot f\left(\gamma^j\right)}{\sum_{j=1}^{2^n} f\left(\gamma^j\right)}.$$

Winkler [1998 – lecture notes] has additional advice for the analyst using the EM algorithm to obtain estimates of the m- and u-probabilities. The two examples that follow illustrate the methodology.

Example 9.1: Using the EM algorithm

We have two pairs of files A_1 and B_1 whose data fields include first name, surname, age, house number, and street name. We use the following initial m- and u-probabilities to get our EM algorithm started (Table 9.1).

We run the EM algorithm and, after a number of iterations, we converge to the following estimated probabilities (Table 9.2).

For this pair of files, we say that the distinguishing power of the individual fields is very good because the m-probabilities are close to one and the u-probabilities are close to zero. Moreover, the agreement weight on first name, for example, is

$$\log_2\left(\frac{P[agree\ on\ first\ name \mid r \in M]}{P[agree\ on\ first\ name \mid r \in U]}\right) = \log_2\left(\frac{.99}{.02}\right) = 5.6$$

and the disagreement weight on first name is

$$\log_2\left(\frac{1 - P[agree\ on\ first\ name \mid r \in M]}{1 - P[agree\ on\ first\ name \mid r \in U]}\right) = \log_2\left(\frac{.01}{.98}\right) = -6.6$$

Example 9.2: Another Illustration of the EM Algorithm

Here, we have two other pairs of files A_2 and B_2 whose data fields again include first name, surname, age, house number, and street name. We use the same initial m- and u-probabilities as we did in the previous example to get our EM

TABLE 9.1. Initial probabilities for files A_1 and B_1

Initial m-probabilities	Initial u-probabilities
P[agree on first name|r ∈ M] = 0.9	P[agree on first name|r ∈ U] = 0.1
P[agree on surname|r ∈ M] = 0.9	P[agree on surname|r ∈ U] = 0.1
P[agree on age|r ∈ M] = 0.9	P[agree on age|r ∈ U] = 0.1
P[agree on house number|r ∈ M] = 0.8	P[agree on house number|r ∈ U] = 0.2
P[agree on street name|r ∈ M] = 0.8	P[agree on street name|r ∈ U] = 0.2

TABLE 9.2. Final estimated probabilities for files A_1 and B_1

Final estimated m-probabilities	Final estimated u-probabilities
P[agree on first name|r ∈ M] = 0.99	P[agree on first name|r ∈ U] = 0.02
P[agree on surname|r ∈ M] = 0.92	P[agree on surname|r ∈ U] = 0.08
P[agree on age|r ∈ M] = 0.90	P[agree on age|r ∈ U] = 0.02
P[agree on house number|r ∈ M] = 0.95	P[agree on house number|r ∈ U] = 0.05
P[agree on street name|r ∈ M] = 0.95	P[agree on street name|r ∈ U] = 0.20

TABLE 9.3. Final estimated probabilities for files A_2 and B_2

Final estimated m-probabilities	Final estimated u-probabilities
P[agree on first name|r ∈ M] = 0.85	P[agree on first name|r ∈ U] = 0.03
P[agree on last name|r ∈ M] = 0.85	P[agree on last name|r ∈ U] = 0.10
P[agree on age|r ∈ M] = 0.60	P[agree on age|r ∈ U] = 0.01
P[agree on house number|r ∈ M] = 0.45	P[agree on house number|r ∈ U] = 0.01
P[agree on street name|r ∈ M] = 0.55	P[agree on street name|r ∈ U] = 0.05

algorithm started. We run the EM algorithm and, after a number of iterations, we converge to the following estimated probabilities (Table 9.3).

On the second pair of files, the distinguishing power of the individual fields is not quite as good as that of the first pair of files. In this example, the agreement weight on first name is

$$\log_2\left(\frac{P[agree\ on\ first\ name\mid r\in M]}{P[agree\ on\ first\ name\mid r\in U]}\right)=\log_2\left(\frac{.85}{.03}\right)=4.8$$

and the disagreement weight on first name is

$$\log_2\left(\frac{1-P[agree\ on\ first\ name\mid r\in M]}{1-P[agree\ on\ first\ name\mid r\in U]}\right)=\log_2\left(\frac{.15}{.97}\right)=-2.7$$

Finally, we note that when the individual agreement weight is higher, it has more of a tendency to raise the total agreement weight (sum of the weights over all the fields) and gives a higher probability that a pair is a match. When a disagreement weight takes a lower negative value, it has more of a tendency to lower the total agreement weight and gives a lower probability that the pair is a match.

9.5. Advantages and Disadvantages of Using the EM Algorithm to Estimate *m*- and *u*-probabilities

In general, the EM algorithm gives us an excellent method of determining the m- and u-probabilities automatically.

9.5.1. First Advantage – Dealing with Minor Typographical Errors and Other Variations

The EM algorithm does well at obtaining probabilities that take into account the effect of minor typographical errors (e.g., Roberta versus Roburta) and other variations (e.g., first name versus nickname – Roberta versus Bobbie) in the data.

If there were no typographical error, then each matched pair would necessarily agree on its associated identifier (i.e., matching variable) with probability one. For instance, $P[agree\ first\ name \mid r \in M] = 1$ and $P[agree\ last\ name \mid r \in M] = 1$. Because there is typographical variation, it is not unusual for $P[agree\ first\ name \mid r \in M] = .9$. In other words, 10% of the truly matched pairs have a typographical variation in an identifier that will not produce an exact match. A human being can easily realize that such pairs are matches. We would like the computer to classify them as matches, too.

Another variation can occur. With one pair of files, A_1 and B_1, we might have $P[agree\ first\ name \mid r \in M] = .9$ and with another we might have $P[agree\ first\ name \mid r \in M] = .8$. We can think of the typographical variation rate (or typographical error rate) as being higher in the second file than in the first. As we saw in Examples 9.1 and 9.2, even when we start with different initial probabilities, the EM algorithm can give us "good" final estimates.

9.5.2. Second Advantage – Getting a Good Starting Point

Another significant advantage of the EM is that it can be used as an exploratory tool for getting good initial estimates of the m- and u-probabilities in many types of files, particularly those having substantial typographical variation.

9.5.3. Third Advantage – Distinguishing Power of m- and u- Probabilities

The EM algorithm is very good at determining the absolute distinguishing power of the m- and u-probabilities for each field and also for determining the relative distinguishing power of fields.

For instance, with individuals, names typically have better distinguishing power than addresses because there may be many individuals at the same address.

With businesses, the opposite is true. Business addresses are usually house-number-street-name types because they represent physical locations of the business entities. Typically, only one business is at a given location. Because many of the businesses have name variants – such as "John K Smith Co" on one list and "'J K S Inc" on another list – it is sometimes difficult to link business records by the name of the business. On the other hand, especially for small businesses, it is not unusual to find the address of (1) the company's accountant, (2) the place of incorporation, or (3) the owner's residence listed as the company's address. Further, a business may have multiple locations.

The EM algorithm will automatically adjust the m- and u-probabilities for both types of matching situations and will usually assign the more useful field more distinguishing power.

en each matched pair would fier (i.e., matching variable) e *first name* | $r \in M$] = 1 and re is typographical variation, it M] = .9. In other words, 10% hical variation in an identifier an being can easily realize that computer to classify them as

pair of files, A_1 and B_1, we .9 and with another we might can think of the typographical as being higher in the second les 9.1 and 9.2, even when we M algorithm can give us "good"

g a Good Starting Point

at it can be used as an exploratory - and u-probabilities in many types typographical variation.

uishing Power

g the absolute distinguishing power nd also for determining the relative

ypically have better distinguishing any individuals at the same address. siness addresses are usually house- present physical locations of the ess is at a given location. Because its – such as "John K Smith Co" er list – it is sometimes difficult the business. On the other hand, unusual to find the address of (1) f incorporation, or (3) the owner's urther, a business may have multiple

djust the m- and u-probabilities for usually assign the more useful field

9.5.4. *The Main Disadvantages of Using the EM Algorithm*

The EM algorithm does not work well in situations of extreme typographical variation, however.

Some implementations of the EM algorithm may not yield good parameter estimates, with business lists, agriculture lists, and some administrative lists, for example. The main reason is the high-failure rate of name and/or address standardization/parsing software. When name or address standardization fails for a given pair that is truly a match, then we typically cannot use key identifying information (either the name or the address) to determine correctly that the pair is a true match. In this situation, the information associated with the particular match makes the pair look more like a non-match. Note that non-matches typically have identifying information such as name and address that does not agree.

9.6. General Parameter Estimation Using the EM Algorithm

Two difficulties frequently arise in applying EM algorithms of the type described in Section 9.4. First, the independence assumption is often violated (see Smith and Newcombe [1975] or Winkler [1989b]). Second, due to model misspecification, the EM algorithm or other fitting procedures may not naturally partition $A \times B$ into the two desired sets: M (the matches) and U (the non-matches). This section lays out a way to handle these difficulties.

9.6.1. *Overcoming the Shortcomings of the Schemes Described Above*

To account for dependencies between the agreements of different matching fields, an extension of an EM-type algorithm originally proposed by Haberman [1975] and later discussed by Winkler [1989a] can be applied. This extension involves the use of multi-dimensional log-linear models with interaction terms. Because many more parameters can be used within such general interaction models than with independence models, only a small fraction of the possible interaction terms may be used in the model. For example, if there are 10 fields used in the matching, the number of degrees of freedom will only be large enough to include all three-way interaction terms; with fewer matching fields, it may be necessary to restrict the model to various subsets of the three-way interaction terms. For more details on this, see Bishop, Fienberg, and Holland [1975] or Haberman [1979].

To address the natural partitioning problem, $A \times B$ is partitioned into three sets or classes: C_1, C_2, and C_3. The three-class EM algorithm works best when we are matching persons within households and there are multiple persons per household.

When appropriate, two of the three classes may constitute a partition of either M or U; for example, we might have $M = C_1$ and $U = C_2 \cup C_3$. When both name and address information are used in the matching, the two-class EM algorithm tends to partition $A \times B$ into (1) those agreeing on address information and (2) those disagreeing. If address information associated with many pairs of records is indeterminate (e.g., Rural Route 1 or Highway 65 West), the three-class EM algorithm may yield a good partition because it tends to partition $A \times B$ into (1) matches at the same address, (2) non-matches at the same address, and (3) non-matches at different addresses.

The general EM algorithm is far slower than the independent EM algorithm because the M-step is no longer in closed form. The use of a variant of the Expectation-Conditional Maximization (ECM) algorithm may speed up convergence (see Meng and Rubin [1993] or Winkler [1992]). The difficulty with general EM procedures is that different starting points often result in different limiting solutions. A recommended approach is to use the independent EM algorithm to derive a starting point for the general scheme as this usually produces a unique solution.

9.6.2. Applications

When matching individuals across two household surveys, we typically have (1) fields on individuals such as first name and age and (2) fields on the entire household such as surname, house number, street name, and phone number. A general two-class EM algorithm will partition a set of record pairs according to the matching variables. Since there are more household fields than fields on individuals, the household fields usually overwhelm person variables in the two-class EM scheme and partition the set of pairs into those at the same address (within the same household) and those not at the same address. In this case, better results are usually obtained by using the three-class EM algorithm. As noted above, the three-class EM algorithm tends to partition the record pairs into three classes: matches within households, non-matches within households, and non-matches outside of households. The three-class EM algorithm works by estimating the probabilities associated with each of these three classes and then estimating the probabilities associated with the non-matches by combining the two non-matching classes.

Example 9.3: Using files with good distinguishing information

This example, which comes from Winkler [2000], entails matching two files of census data from Los Angeles. Each file contained approximately 20,000 records, with the smaller file having 19,682 records. The observed counts for $1024 \, (= 2^{10})$ agree/disagree patterns on 10 fields were used to obtain most parameter estimates. These fields were first name, middle initial, house number, street name, unit (or apartment) number, age, gender, relationship to head of household, marital status, and race. Frequency-based weights were created for both surname and first name. The basic weight for last name was created via an ad hoc procedure that attempted to account for (1) typographical error and (2)

ses may constitute a partition of either
C_1 and $U = C_2 \cup C_3$. When both name
natching, the two-class EM algorithm
reeing on address information and (2)
associated with many pairs of records
Highway 65 West), the three-class EM
cause it tends to partition $A \times B$ into
-matches at the same address, and (3)

r than the independent EM algorithm
osed form. The use of a variant of
on (ECM) algorithm may speed up
3] or Winkler [1992]). The difficulty
fferent starting points often result in
led approach is to use the independent
for the general scheme as this usually

household surveys, we typically have
ne and age and (2) fields on the entire
ber, street name, and phone number.
artition a set of record pairs according
are more household fields than fields
lly overwhelm person variables in the
t of pairs into those at the same address
not at the same address. In this case,
sing the three-class EM algorithm. As
hm tends to partition the record pairs
holds, non-matches within households,
The three-class EM algorithm works by
ith each of these three classes and then
ith the non-matches by combining the

nguishing information

ler [2000], entails matching two files
h file contained approximately 20,000
),682 records. The observed counts for
1 10 fields were used to obtain most
irst name, middle initial, house number,
r, age, gender, relationship to head of
quency-based weights were created for
weight for last name was created via an
ount for (1) typographical error and (2)

the number of pairs in the subset of pairs on which the matching was performed. A total of 249,000 pairs of records that agreed on the first character of the last name and the Census block number were considered. As there can be at most 20,000 matches here, it is not computationally practical to consider counts based on all of the 400 million pairs in the product space. Based on prior experience, Winkler was confident that more than 70% of the matches would be in the set of 249,000 pairs. The results are summarized in Table 9.4.

Winkler gave several reasons why frequency-based matching performed better here than basic matching that uses only agree/disagree weights. First, the frequency-based approach designated 808 pairs of records as matches that the basic matcher merely designated as possible matches. These pairs of records were primarily those having (1) rare surnames, (2) rare first names, and (3) a moderate number of disagreements on other fields. Second, the frequency-based approach designated 386 pairs of records as possible matches that the basic matcher designated as non-matches. These pairs of records were primarily those having (1) rare last names and (2) few, if any, agreements on other fields.

Example 9.4 – Using files with poor distinguishing information

This example, also from Winkler [2000], entails matching two files of census data from St. Louis. The larger file contains 13,719 records while the smaller one has 2,777 records. The smaller file was obtained by merging a number of administrative data files. The observed counts for 128 $(= 2^7)$ agree/disagree patterns on seven fields were used to obtain most parameter estimates. These fields were first name, middle initial, address, age, gender, telephone number, and race. Frequency-based weights were created for both last name and first name. The basic weight for last name was created via an ad hoc procedure that attempted to account for (1) typographical error and (2) the number of pairs in the subset of pairs on which the matching was performed. A total of 43,377 pairs of records that agreed on the Soundex code (see Chapter 10) of the last name were considered. The results are summarized in Table 9.5.

Neither matching scheme performed well in this example. Each scheme classified approximately 330 records as matches. The file used to obtain additional information about black males between the ages of 18 and 44 had many missing data fields. The middle initial, telephone number, and race fields were missing on 1,201, 2,153 and 1,091 records, respectively. The age and

TABLE 9.4. Comparing basic estimates to frequency-based estimates (data from Los Angeles)

Basic approach	Frequency-based approach			Total
	Match	Possible match	Non-match	
Match	12,320	128	7	12,455
Possible match	808	2,146	58	3,012
Non-match	8	386	3,821	4,215
Total	13,136	2,660	3,886	19,682

TABLE 9.5. Comparing basic estimates to frequency-based estimates (data from St. Louis)

Basic approach	Frequency-based approach			Total
	Match	Possible match	Non-match	
Match	305	21	2	328
Possible match	15	142	0	157
Non-match	2	106	2,184	2,292
Total	332	269	2,186	2,777

address fields were also frequently incorrect. The frequency-based matching scheme designated 269 record pairs as possible matches versus 157 for the basic matching scheme. These 269 record pairs typically had (1) a relatively rare first name, (2) a relatively rare last name, and (3) few, if any, agreements on other fields.

9.7. Where Are We Now?

In this chapter, we have discussed a variety of schemes for estimating the parameters of the Fellegi–Sunter record linkage model. We also discussed the advantages and disadvantages of these schemes. We concluded the chapter with a pair of examples in which we compared frequency-based record linkage to basic record linkage. In the next four chapters, we discuss a number of techniques that can be used to enhance record linkages.