Copy or Coincidence? A Model for Detecting Social Influence and Duplication Events

Lisa Friedland LFRIEDL@CS.UMASS.EDU
David Jensen JENSEN@CS.UMASS.EDU
School of Computer Science, University of Massachusetts, Amherst, MA 01003 USA

Michael Lavine LAVINE@MATH.UMASS.EDU
Department of Math and Statistics, University of Massachusetts, Amherst, MA 01003 USA

Abstract

In this paper, we analyze the task of inferring rare links between pairs of entities that seem too similar to have occurred by chance. Variations of this task appear in such diverse areas as social network analysis, security, fraud detection, and entity resolution. To address the task in a general form, we propose a simple, flexible mixture model in which most entities are generated independently from a distribution but a small number of pairs are constrained to be similar. We predict the true pairs using a likelihood ratio that trades off the entities' similarity with their rarity. This method always outperforms using only similarity; however, with certain parameter settings, similarity turns out to be surprisingly competitive. Using real data, we apply the model to detect twins given their birth weights and to re-identify cell phone users based on distinctive usage patterns.

1. Introduction

The following tasks come from different domains, but they share a common core:

- Can we infer social ties among people whose Flickr photographs are geographically co-located? (Crandall et al., 2010)
- Can we detect (and block) coalitions of attackers clicking on the same advertisements as part of a fraud scheme? (Metwally et al., 2007)
- Can we identify duplicate records to be merged in a customer database? (Elmagarmid et al., 2007)
- Can we determine with confidence whether a crime scene fingerprint matches one in a database? (Su & Srihari, 2010)

Each task concerns data in which most entities (people or records) are distinct and independent, but certain pairs or small groups are unusually similar. The similarity reflects an unobserved link we would like to detect, such as "these people are acting in coordination" or "these are two traces of the same object."

This class of problems arises in fields such as social network analysis (Adamic & Adar, 2003; Bejder et al., 1998), entity resolution (see Section 3), fraud and plagiarism detection (Friedland & Jensen, 2007; Sorokina et al., 2006), security (Yang et al., 2011) and forensics (Committee on DNA Forensic Science, 1996). From a privacy perspective, we ask the same question with an opposing goal: when is an individual's behavior or attributes distinctive enough to be identifiable across multiple sightings (Whang & Garcia-Molina, 2011; Narayanan & Shmatikov, 2008)? Many of these applications are longstanding, well-studied problems, but each is addressed separately. This motivates us to connect them as instances of a single formal task.

In these problems, the goals are to identify the links and to assess their significance. Intuitively, a pair is more likely to be linked the more the entities are *similar* and the more the entities (or merely their shared aspects) are *rare*. (Pairs can also occur in dense regions, but those pairs will be less distinguishable.) Across the literature, numerous measures of pair strength have been developed. These usually describe the similarity of the entities, and sometimes also their rarity. Some measures are probabilistically based, and almost all are domain-specific.

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP Volume 28.

We, instead, explicitly model how both paired and non-paired entities are generated. With a likelihood ratio that compares the paired and non-paired models, our method takes into account both similarity and rarity. We work with the simplest of systems—continuous data and Gaussian distributions—in order to minimize domain-specific aspects and focus on these questions:

- Supposing we knew everything about a domain, how would this task be solved optimally?
- Do we even need a model, or will a simple distance-only baseline be equally effective? If so, why and under what circumstances? (Section 5.3)
- As we approach realistic scenarios, in which the distance between pairs or the number of pairs is not known (Section 5.4), or in which the form of the model might not fit the data (Section 6), will this method still be feasible?

In Section 2 of this paper, we present a generative model for continuous data in k dimensions, and for inference, a likelihood ratio score ("LR") to compute for every pair. In the synthetic data of Section 5, we find that one key parameter most affects performance: t, which describes how far apart the linked pairs may be. We compare LR to baseline methods that measure only similarity of pairs ("d", for distance), only rarity, or sub-optimal combinations of the two. Surprisingly, we find that d can perform almost as well as LR—that is, rarity doesn't matter—but only for the easiest problems, those with the smallest values of t. By examining the theoretical distributions of positive (i.e., linked) and negative (non-linked) pairs, we are able to explain why this happens.

Moving towards situations where parameters are unknown (and true labels might be unavailable), we examine performance when our estimate $\hat{t}$ mismatches the model and discover it governs the score's balance of similarity vs. rarity. When the optimal t is unknown, the approximation $\frac{P(d|\epsilon)}{P(m|\phi)}$ is a robust alternative. In Section 6 we apply the model to two real data sets constructed to be labeled instances of this task. As we vary $\hat{t}$, the performance trends are comparable to those in synthetic data. We find that both real data sets are in a middle range of difficulty, a range where performance is only moderate, but where LR distinctly outperforms d.

2. Model and Inference

The model below makes the following assumptions, which are reasonable for many applications. First, the number of linked entities is low. Second, the linked entities appear only in disjoint pairs, not larger groups. Third, the non-linked entities—the vast majority—can be modeled as being independently generated from some distribution ϕ. Finally, the pairs can be modeled as being generated jointly in a process θ that involves ϕ but also involves a distribution ϵ keeping pairs close together. We deliberately keep the model simple so that we can study the effects of parameter choices. Yet it is flexible, in that arbitrary domains and distributions could be swapped in with different choices of ϕ and θ; in particular, one could specify an ϵ that makes pairs be far apart or in another specific configuration.

2.1. Generative Process and Task

The output will be n points, $\mathbf{x}_1, \ldots, \mathbf{x}_n$ in $\mathbb{R}^k$, where some pairs are generated together. Let ϕ be the distribution of singleton points. Let θ be the process for generating pairs; within θ, we must specify ϵ, a distribution by which pairs of points are displaced from their common midpoint. Two variables are unobserved: r, the actual number of pairs, and $C = \{c_{ij}\}$, a (binary) adjacency matrix describing which points are in pairs. We control the number of pairs with the variable q, such that the expected number of pairs $E(r) = qn$.

When $c_{ij} = 1$ we say that the points x_i and x_j form a *pair* (or a *link*), or equivalently, that the pair is *positive*; when $c_{ij} = 0$ we say that the points are *singletons* or that the pair is *negative*.

The generative process is as follows. First, choose how many and which points are in pairs.

1. Generate r, the number of pairs: $r \sim \text{Binomial}(n/2, 2q)$. (With this proportion, $r \in [0, n/2]$, and $\text{E}(r) = qn$.)
2. Generate $C = \{c_{ij}\}$ uniformly from among all matrices of r links where no point has > 1 link. Let $a_i \in \{0, 1\}$ indicate the number of links incident to point i in C.

At this stage, for each $\mathbf{x}_i$, we know whether it will be a singleton or part of a pair with $\mathbf{x}_j$.

3. Generate $\mathbf{x}_1, \ldots, \mathbf{x}_n$:
   (a) If $a_i = 0$, then generate $\mathbf{x}_i \sim \phi$.
   (b) For each pair (i, j) for which $c_{ij} = 1$, generate $(\mathbf{x}_i, \mathbf{x}_j) \sim \theta$:
      i. Generate $\mathbf{m}_{ij} \sim \phi$
      ii. Generate displacement vector $\mathbf{d}_{ij} \sim \epsilon$.
      iii. Set $\mathbf{x}_i = \mathbf{m}_{ij} + \mathbf{d}_{ij}$ and $\mathbf{x}_j = \mathbf{m}_{ij} - \mathbf{d}_{ij}$

This is essentially a mixture model for the data: one mixture component is a distribution of points (ϕ), the other is a distribution of pairs (θ). The distributions are connected in that θ uses ϕ: the pairs' midpoints are generated the same way as the singleton points.

2.2. Inference

In this paper, we never explicitly infer r or C. Instead, to make inference efficient, we reason about each possible link as if it were independent of the others. We produce a likelihood ratio for each c_{ij} and evaluate this ranking against the true set $\{c_{ij}\}$. The likelihood ratio (below) is rank-equivalent to the probability of the pair being positive: $P(c_{ij} = 1 \mid \mathbf{x}) = \frac{LR}{1+LR}$.

We approximate, for every pair of points:

$$\frac{P(c_{ij} = 1 \mid \mathbf{x}_1, \ldots, \mathbf{x}_n)}{P(c_{ij} = 0 \mid \mathbf{x}_1, \ldots, \mathbf{x}_n)} \approx \frac{P(c_{ij} = 1 \mid \mathbf{x}_i, \mathbf{x}_j)}{P(c_{ij} = 0 \mid \mathbf{x}_i, \mathbf{x}_j)} \quad (1)$$

$$= \frac{P(\mathbf{x}_i, \mathbf{x}_j \mid c_{ij} = 1)P(c_{ij} = 1)}{P(\mathbf{x}_i, \mathbf{x}_j \mid c_{ij} = 0)P(c_{ij} = 0)} \quad (2)$$

$$= \frac{\frac{1}{2^k} P(\mathbf{m}_{ij} \mid \phi)P(\mathbf{d}_{ij} \mid \epsilon)P(c_{ij} = 1)}{P(\mathbf{x}_i \mid \phi)P(\mathbf{x}_j \mid \phi)P(c_{ij} = 0)} \quad (3)$$

Line (2) is an application of Bayes' Rule. In Line (3), we use Step 3 of the generative model to write out the likelihoods for positive and negative pairs, respectively.

The generative process for positive pairs was described in terms of $\mathbf{m}_{ij}$ and $\mathbf{d}_{ij}$, so the most natural way to write its likelihood function would be $P(\mathbf{m}_{ij}, \mathbf{d}_{ij} \mid c_{ij} = 1) = P(\mathbf{m}_{ij} \mid \phi)P(\mathbf{d}_{ij} \mid \epsilon)$. Since Lines (2) and (3) are written as functions of $(\mathbf{x}_i, \mathbf{x}_j)$, we have to perform a change of variables; the mapping is one-to-one but introduces the constant $\frac{1}{2^k}$ (see Lemma 8.1[1]).

The term for the prior $P(c_{ij} = 1)$ is r divided by the total number of pairs, so $\frac{2r}{n(n-1)}$ when r is known. When r is unknown, we compute the term by summing over possible values[2] of r (Eq. (4)). In Eq. (5), $P(r = k \mid q)$ is expanded using $r \sim \text{Binomial}(n/2, 2q)$. In either case, $P(c_{ij} = 0) = 1 - P(c_{ij} = 1)$.

$$P(c_{ij} = 1 \mid q) = \sum_{k=1}^{n/2} P(r = k \mid q)P(c_{ij} = 1 \mid r = k) \quad (4)$$

$$= \sum_{k=1}^{n/2} \binom{n/2}{k} (2q)^k (1 - 2q)^{n/2-k} \frac{2k}{n(n-1)} \quad (5)$$

2.3. Limitations of this Inference Method

The output of inference is a list of likelihood ratios, one for each potential pair. We can turn this into a discrete set of positive pairs, if desired, by thresholding the scores. One drawback to treating each pair as independent is that, in violation of the generative model, the resulting (thresholded) adjacency matrix $\hat{C}$ may assign points to more than one pair. We could remedy this situation with additional post-processing (instead of or in addition to the thresholding), keeping only the highest-probability links. Alternatively, we could reconsider the model's assumptions: if a point is matched to more than one pair, we may have underestimated ϕ in that region or the points may actually belong to a group of more than two. It could be a strength if the method is able to detect such groups when the generative process only describes pairs.

Another way to avoid assigning any point to more than one pair would be to infer the full C: compute $P(C_l \mid \mathbf{x}_1, \ldots, \mathbf{x}_n)$ for every valid matrix C_l and choose the one with maximum likelihood. This would be computationally challenging: for a typical data set in this paper, there are more than 1.6×10^{16} such matrices.

Another simplification is that we model all negative pairs as if they were formed by singleton points. In truth, of the $\frac{n(n-1)}{2} - r$ negative pairs, $2r(n-r-1)$ of them involve at least one point from a positive pair. As r rises from 1 to $\frac{n}{2}$, the fraction of non-modeled pairs increases from near-0 to near-all of them. In Section 5.4, we discuss how these non-modeled negatives can under certain circumstances affect performance.

3. Related Work

This task differs from clustering in that our expected clusters (links) are tiny and rare; if the data does contain large-scale clusters, they should be modeled in ϕ so that we can recognize deviations from them. The task has more in common with significance testing: we want to distinguish true pairs from singletons that are close together by chance. It can also be seen as an anomaly detection problem (Chandola et al., 2009), not in the generic sense of "outlier detection" but in the sense of "detecting a specific unusual pattern." In that vein it is similar to Eskin's (2000) mixture model of normal and anomalous elements.

One central related task is link prediction in social networks based on shared interests or behavior. Adamic & Adar (2003) develop a score to combine rarity with similarity of shared interests; Liben-Nowell & Kleinberg (2007) compare a variety of distance measures between nodes in an observed network; and Friedland & Jensen (2007) compute the rarity of the shared component of people's job histories. Most similar to our work is a generative model by Crandall et al. (2010) in which pairs of friends travel to locations together.

The other closely related area is entity resolution, or

[1] Section 8 is attached as Supplementary Material.

[2] Note that the summation omits the term $k = 0$. Although our process can generate data sets having $r = 0$, we discard those samples because our performance measure is only defined in the presence of positive pairs.

record matching (Elmagarmid et al., 2007; Winkler, 2006). That literature, while extensive, makes some key assumptions that prevent its methods from being directly transferable here. Generally the duplicates to identify are database records that correspond to the same real-world entity, and the records consist of text fields such as names and addresses. Although numerous text comparison metrics have been developed, little has been done with continuous data. Finally, that work does not restrict links to be rare or disjoint.

One popular text matching function explicitly incorporates rarity: it weights each word (or substring) by its tf $\cdot$ idf measure, then takes the cosine similarity of the resulting vectors (Cohen et al., 2003). Chaudhuri et al. offer a complementary approach in which, regardless of the distance measure, clusters are required to be both close together and in sparse regions (2005).

Much of probabilistic record matching is based on the Fellegi-Sunter model (1969). It ranks pairs by the likelihood ratio $\frac{P(\gamma|c_{ij}=1)}{P(\gamma|c_{ij}=0)}$, where γ is some function of the pair—a "comparison vector." If γ is merely a distance measure, then that model would be like our baseline $LR[d]$ (see Section 5.2). Since typically γ also encodes which particular words match, the resulting score is higher when matching strings are rare. Our likelihood ratio of Eq. (3) could be seen as a general form of the Fellegi-Sunter model, in which γ is the points themselves ($\mathbf{x_i}, \mathbf{x_j}$), and in which $P(\gamma|c_{ij})$ is provided by the generative model rather than estimated from data.

Compared to related tasks, our work's strength is in abstracting away the domain-specific elements, allowing a focus on the problem's more general principles.

4. Evaluation

We evaluate performance by comparing a ranked list of predicted pairs to the set of true pairs, calculating the AUC (area under the ROC curve) of the ranking. We considered other common measures of ranking such as average precision or Hand's H measure (2009), but they were unsuitable because, unlike AUC, they fluctuate when the number of true positives or negatives does. In realistic scenarios it may also be important to focus attention on the very top of the ranked list or on the individual probability estimates. These paths are left to future work.

For present purposes, the ranked list contains all pairs. In larger data sets, efficiency would become a concern, as it is in entity resolution. Existing techniques from that literature address efficiency either by making the score calculation faster or by scoring only those subsets of pairs that are judged similar according to some preliminary measure (Elmagarmid et al., 2007). McCallum et al. (2000) describe a method for continuous data that could be used here: in each dimension, create overlapping bins for the data, and only consider pairs that lie within the same bin in some dimension. For the data sets in this paper and practical values of parameters, applying this method, i.e., filtering out pairs with a high $\mathbf{d_{ij}}$, would probably bring gains in efficiency at little loss to performance.

5. Applying the Model to Synthetic Data

In this section, we study the behavior of the algorithm when the data has been generated by the model. For the following analyses and experiments we set ϕ and ϵ to be radially symmetric normal distributions: $\phi = \text{Normal}(\boldsymbol{\mu}, \sigma^2 I)$, and $\epsilon = \text{Normal}(\mathbf{0}, \nu^2 I)$.

5.1. Simplifying the Score

Starting from Eq. (3), we plug in normal probability density functions for the terms involving ϕ and ϵ:

$$P(\mathbf{m}_{ij} \mid \phi)P(\mathbf{d}_{ij} \mid \epsilon) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^k e^{-\frac{\|\mathbf{m}_{ij}-\boldsymbol{\mu}\|^2}{2\sigma^2}} \left(\frac{1}{\sqrt{2\pi}\nu}\right)^k e^{-\frac{\|\mathbf{d}_{ij}\|^2}{2\nu^2}} \tag{6}$$

$$P(\mathbf{x}_i \mid \phi)P(\mathbf{x}_j \mid \phi) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^k e^{-\frac{\|\mathbf{x}_i-\boldsymbol{\mu}\|^2}{2\sigma^2}} \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^k e^{-\frac{\|\mathbf{x}_j-\boldsymbol{\mu}\|^2}{2\sigma^2}} \tag{7}$$

$$= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2k} e^{-\frac{m^2+d^2}{\sigma^2}}. \tag{8}$$

For Eq. (8), we have defined $m = \|\mathbf{m}_{ij} - \boldsymbol{\mu}\| = \|\frac{(\mathbf{x}_i+\mathbf{x}_j)}{2} - \boldsymbol{\mu}\|$ and $d = \|\mathbf{d}_{ij}\| = \|\frac{(\mathbf{x}_i-\mathbf{x}_j)}{2}\|$ (dropping the subscript ij when it is clear from context) and applied Lemma 8.2.

Substituting the densities back into Eq. (3)'s likelihood ratio gives:

$$\frac{P(c_{ij}=1 \mid \mathbf{x}_i, \mathbf{x}_j)}{P(c_{ij}=0 \mid \mathbf{x}_i, \mathbf{x}_j)} = \frac{\frac{1}{2^k}\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^k e^{-\frac{m^2}{2\sigma^2}} \left(\frac{1}{\sqrt{2\pi}\nu}\right)^k e^{-\frac{d^2}{2\nu^2}} P(c_{ij}=1)}{\left(\frac{1}{\sqrt{2\pi}\sigma}\right)^{2k} e^{-\frac{m^2+d^2}{\sigma^2}} P(c_{ij}=0)} = \left(\frac{\sigma}{2\nu}\right)^k e^{\frac{1}{2}\left(\frac{m^2+2d^2}{\sigma^2}-\frac{d^2}{\nu^2}\right)} \frac{P(c_{ij}=1)}{P(c_{ij}=0)}. \tag{9}$$

The likelihood ratio in Eq. (9) is fairly simple: instead

of depending on the full data vectors $\mathbf{x}_i$ and $\mathbf{x}_j$—$2k$ coordinates in all—it uses just two measures of the pair, m and d.

We assume (for now) that the model parameters are available at inference time. Among them, n and r (or q) affect only $\frac{P(c_{ij}=1)}{P(c_{ij}=0)}$. Changing them affects the individual scores, but not the ranking. We also need σ and ν. However, it turns out we can rewrite the score as a function of their ratio $t = \frac{\nu}{\sigma}$. Eq. (10) shows the final, reparametrized LR as a function of $m' = \frac{m}{\sigma}$, $d' = \frac{d}{\sigma}$, and $t = \frac{\nu}{\sigma}$ without σ:

$$\frac{P(c_{ij}=1 \mid \mathbf{x}_i, \mathbf{x}_j)}{P(c_{ij}=0 \mid \mathbf{x}_i, \mathbf{x}_j)} = \left(\frac{1}{2t}\right)^k e^{\frac{1}{2}\left(m'^2+d'^2\left(2-\frac{1}{t^2}\right)\right)} \frac{P(c_{ij}=1)}{P(c_{ij}=0)}. \tag{10}$$

In the rest of Section 5, we will address (a) how the task's difficulty is affected by model parameters (primarily t, but also the dimensionality k, the number of points n, and the number of pairs r or q); (b) how the score for an individual pair varies as a function of t and its (m', d') values (Section 5.3); and (c) how performance is affected by changing the value $\hat{t}$ used during inference (Section 5.4).

5.2. Performance on Synthetic Data

For synthetic data experiments, given any parameter setting of n, q and t, we generate 100 data sets from the model. Within each data set, we score every pair and evaluate the AUC of the ranked list compared to the true pairs. These experiments use $k = 2$ dimensions and (without loss of generality) $\sigma = 1$.

The likelihood ratio ("LR") of Eqs. (3) and (10) is the Bayes estimate for distinguishing positive from negative pairs, so it should perform close to optimally, depending on how closely the data matches the two modeled classes. We compare it to four baseline methods.

One, d, measures only the similarity of points in a pair: it ranks by d_{ij}, the distance between the points, with smaller distance meaning more likely positive. The second, m, measures only the rarity (i.e., local sparseness) of the pair: it ranks by m_{ij}, the distance from the origin to their midpoint, with higher distance meaning more likely positive. It can be seen from Eq. (10) that using m (or m') is rank-equivalent to using LR if d' is held constant. Likewise, using d (or d') is rank-equivalent to using LR if m' is held constant—provided that $\frac{1}{t^2} > 2$, or $t < 1/\sqrt{2} \approx 0.71$. Generally we will use $t \ll 1$, so this will be the case.

The third baseline, called $LR[d]$, is a likelihood ratio designed to take into account only d, not m. It is computed as $\frac{P(\mathbf{d}|c_{ij}=1)}{P(\mathbf{d}|c_{ij}=0)}$. For the synthetic data, the score is similar to Eq. (10), but the discriminant function in the exponential reduces to $d'^2\left(2-\frac{1}{t^2}\right)$. The fourth baseline, $\frac{P(d|\epsilon)}{P(m|\phi)}$, is an intuitive if naive way to combine the the terms for similarity and for rarity. But it is actually a reasonable approximation to the full LR of Eq. (3) when d is small enough, because in that case $P(m \mid \phi) \approx P(x_i \mid \phi) \approx P(x_j \mid \phi)$ and the terms cancel out. In the synthetic data, this method is rank-equivalent to $\left(m'^2 + d'^2\left(\frac{-1}{t^2}\right)\right)$.

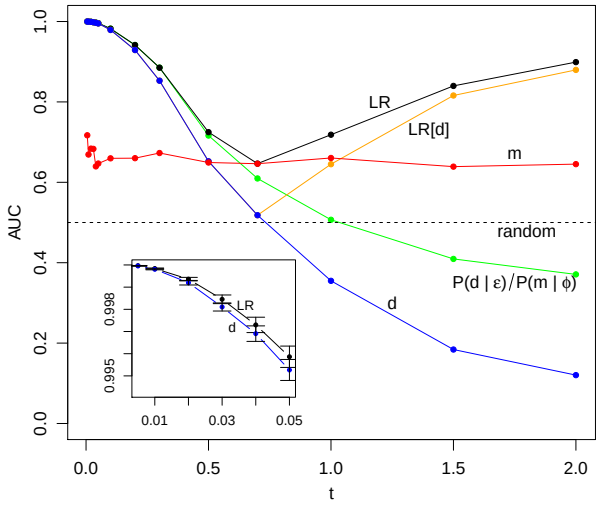

Figure 1. AUC as a function of t, for five methods. Each point is the average of 100 trials. Inset shows a close-up of the smallest values of t, with error bars indicating 95% confidence intervals. In the inset, $P(d \mid \epsilon)/P(m \mid \phi)$ would be visually indistinguishable from LR. Parameters are $n = 200$, $E(r) = 4$, and $\sigma = 1$.

Figure 1 shows performance as we vary t for one setting of (n, q). (Other settings were similar.) The results can be divided into three realms. First, when t is very low (see inset), the AUCs of both LR and d are almost perfect. LR is always above d, but they are nearly indistinguishable. Next, as t approaches $1/\sqrt{2}$, both LR and d drop, and they diverge; at its minimum value, LR matches m, while d is nearly 0.5, or random. When $t > 1/\sqrt{2}$, LR increases again, while d continues to decrease, now ranking pairs in the wrong order. Meanwhile, m is much lower and steady. The third and fourth baselines each partially augment d: $LR[d]$ is identical except that it changes the direction of ranking at $1/\sqrt{2}$, and $\frac{P(d|\epsilon)}{P(m|\phi)}$ incorporates m, so it performs near optimal for low t, but it does not change direction at $1/\sqrt{2}$.

5.3. Understanding Performance

Conceptually, we can explain why $t = 1/\sqrt{2}$ is always a turning point, regardless of the form of ϕ. In each dimension l, $d_l = \frac{x_{il} - x_{jl}}{2}$, so for negative pairs, $\mathrm{E}(d_l|-) = 0$ and $\mathrm{Var}(d_l|-) = \frac{1}{2}\mathrm{Var}(x_l) = \frac{\sigma_l^2}{2}$. For the positive pairs, by definition $\mathrm{Var}(d_l|+) = \nu_l^2 = (t\sigma_l)^2$, so when we set $t = 1/\sqrt{2}$, the positives' $\mathrm{Var}(d_l|+) = \frac{\sigma_l^2}{2}$ matches that of the negatives. In these experiments, not only do the variances of d match at $t = 1/\sqrt{2}$, but since ϕ and ϵ are normals and ϵ is centered at 0, the distributions of d_l are normals, identical for the positive and negative pairs. Therefore d contains no distinguishing information, and LR is only using m. At higher t, the positives become farther apart, on average, than the negatives.

We next examine how the LR score of an individual pair combines the two measures of it, m' and d'. Figure 2 shows that the score increases when m' increases; for the boxes in which $t < 1/\sqrt{2}$, the score increases when d' decreases, and when $t > 1/\sqrt{2}$, the score increases when d' increases, as discussed above. At $t \approx 1/\sqrt{2}$ the contour lines are vertical, which shows visually that the only information is contained in m. Now, consider the smallest setting of t, in which empirically d performs almost as well as LR. The contour lines in the first box are almost horizontal, indicating that d' contains almost all the information (in the LR score, $\frac{d'^2}{t^2} \gg m^2$). This dominance of d' explains why the two methods are almost indistinguishably strong.

Figure 2 becomes more informative once we know not only what score is assigned to a given position, but also the distributions of positive and negative pairs along these axes. It turns out that with normal distributions for ϕ and ϵ in $\mathbb{R}^k$, the distributions of positive and negative pairs have closed forms (full derivations are in Section 8.2). Each distribution is a product of two independent χ_k distributions, one describing m', one describing d':

$$P(m' \mid \phi)P(d' \mid \epsilon) = \left(\frac{1}{t}\right)\chi_k(m')\chi_k\left(\frac{d'}{t}\right) \quad (11)$$

$$P(m' \mid \phi)P(d' \mid \phi) = 2\chi_k(m'\sqrt{2})\chi_k(d'\sqrt{2}). \quad (12)$$

The peak of χ_k is at $\sqrt{k-1}$. Since $k = 2$ here, that peak is at $(1, t)$ for the positive pairs and $(1/\sqrt{2}, 1/\sqrt{2})$ for the negatives. As t changes, the only effect is on the d' dimension of the positives. Visually, it is clear that the distributions are well separated at small t and begin to overlap as t grows. In higher dimensions, the distributions become better separated (see Section 8.3), so the task should become easier as k increases.

5.4. Sensitivity to Parameters and to Assumptions

When n increases or q decreases, intuition suggests that since true pairs are less frequent, the problem get harder. However, since AUC is unaffected by changes to class proportions, a glance at the class distributions of Figure 2 should help solidify the (more relevant) intuition that changing the number of positives or negatives will not affect the separation between the classes. At inference time, if we mis-guess q, the probability estimates for pairs change, but the LR ranking does not.

At data generation time, the situation is more subtle. For a given n, as the number of pairs increases towards $n/2$, the performance of LR can actually decrease—but only for large $t > 1/\sqrt{2}$. This is due to interference of the non-modeled pairs described in Section 2.3: at large t, the positive points no longer resemble the singletons, so the majority of negatives no longer resemble the modeled negatives. However, we observe no such performance effects with smaller t.

In many realistic problem scenarios, we will not know q nor, more importantly, t. Figure 3 shows how performance degrades when using an incorrect value $\hat{t}$ for inference. For LR, $\hat{t}$ determines the balance between d' and m', and the direction of d''s effect. When $\hat{t}$ approaches 0, LR approaches d; when $\hat{t}$ reaches $1/\sqrt{2}$, LR matches m, then continues to drop; and the optimal is in between, at the true t. For $\frac{P(d|\epsilon)}{P(m|\phi)}$, performance is surprisingly robust: when $\hat{t}$ is underestimated, performance drops just like LR's, but when $\hat{t}$ is overestimated, $\frac{P(d|\epsilon)}{P(m|\phi)}$ remains high. This is because $\frac{P(d|\epsilon)}{P(m|\phi)}$ has no turning point in its use of d: as $\hat{t} \to \infty$, $\frac{P(d|\epsilon)}{P(m|\phi)}$ merely puts less weight on d and eventually converges to m. Meanwhile, $LR[d]$ simply matches d, and its AUC flips to $(1 - d)$ when $\hat{t} > 1/\sqrt{2}$.

The implications for data sets with unknown parameters can be summarized as follows. Mis-guessing q does not affect the ranking, and our inference methods seem to work well even when the data contains a large number of pairs, as long as $t < 1/\sqrt{2}$. As long as we know positive pairs are closer together than negative pairs, then when using LR, $\hat{t}$ should always be less than $1/\sqrt{2}$. Finally, mis-guessing t can be harmful, but there are several options for avoiding the performance drop-off: (a) use d, which is parameter-free and often performs well, (b) underestimate t, rather than overestimate it, to ensure performance will not drop below d, or (c) use $\frac{P(d|\epsilon)}{P(m|\phi)}$, which is more robust to overestimates of t.

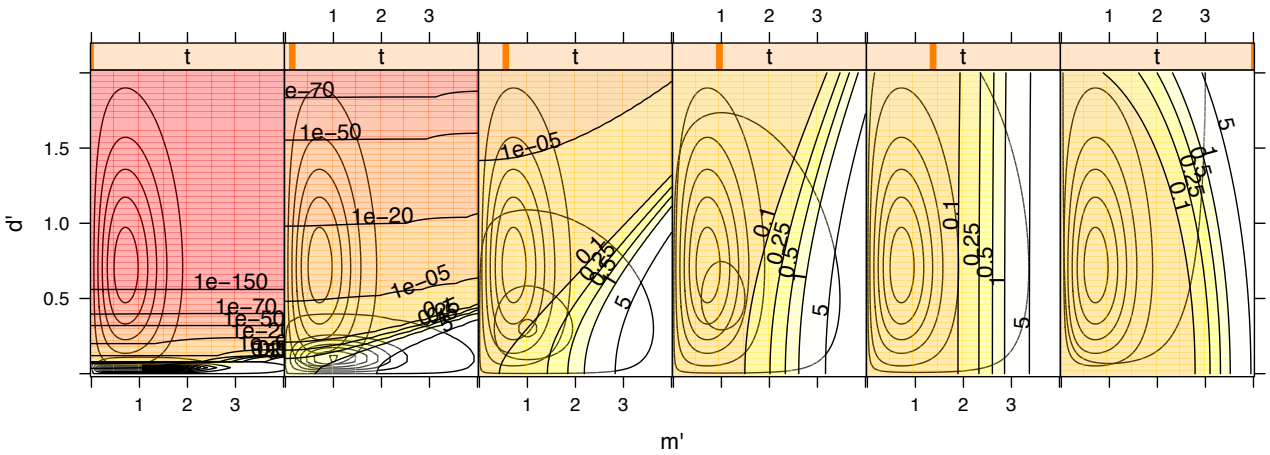

Figure 2. Color and labeled contour lines: likelihood ratio assigned as a function of (m', d') when $n = 25$, $\mathrm{E}(r) = 10$. Higher $P(c_{ij} = 1 \mid m', d')$ is whiter. Within each box: left contour lines: density function for negative pairs; bottom/middle contour lines: density function for positive pairs. Top orange bar: relative values of t across (0.02, 0.1, 0.3, 0.5, 0.7, 2).

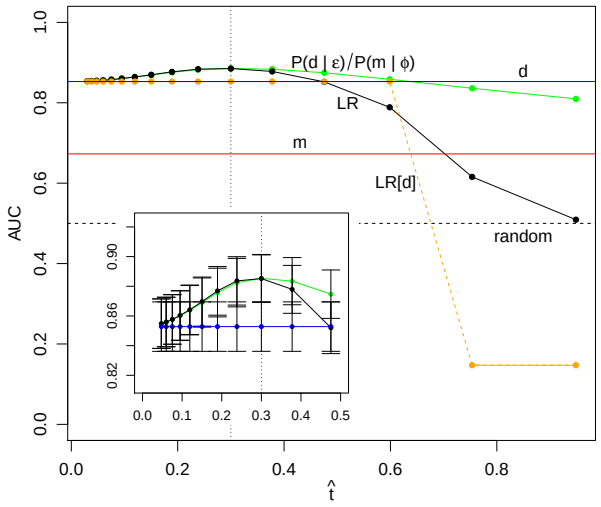

Figure 3. Performance as $\hat{t}$ varies. True parameters are $t = 0.3$ (vertical dotted line), $n = 200$, and $E(r) = 4$.

6. Applying the model to real data

To apply this model to an arbitrary data set in $\mathbb{R}^k$, we need to specify several parameters. The distribution of singletons is straightforward: estimate ϕ (of any desired form) from the entire data set. For positive pairs, we preserve the generative process θ in which $\mathbf{m} \sim \phi$ and $\mathbf{d} \sim \epsilon$. We let ϵ remain a normal, but it should no longer be radially symmetric, since the variables might be at different scales. We define the vector version of $\mathbf{t}$ such that $t_l = \frac{\nu_l}{\hat{\sigma}_l}$ in each dimension l, where $\hat{\sigma}_l$ is the (empirical) estimate of the variance of the negatives. Then we can write $\mathbf{d} \sim \epsilon = \text{Normal}(0, \mathbf{t}'\hat{\Sigma}^{-1}\mathbf{t})$ where $\hat{\Sigma}$ is a diagonal covariance matrix estimated from the data. As before, the key parameter to specify is $\mathbf{t}$, which describes the distance between the positive pairs. That distance will match the negative pairs when $\mathbf{t} = {}^1/\sqrt{2}(1, 1, \ldots, 1)$.

The baseline methods d and m can be generalized as $P(\mathbf{d} \mid \epsilon)$ and $\frac{1}{P(\mathbf{m} \mid \phi)}$, respectively. When all the components of $\mathbf{t}$ are equal, $P(\mathbf{d} \mid \epsilon)$ becomes rank-equivalent to a natural k-dimensional measure, scaled Euclidean distance. The method $LR[d]$ requires an estimate of $P(\mathbf{d} \mid c_{ij} = 0)$; for this, we fit a normal to the set of all pairwise displacement vectors $\mathbf{d}$.

6.1. Data sets

The Matched Multiple Birth Data from the National Center for Health Statistics (2000) contains infant birth and mortality data for all twins and larger multiples born in the U.S. from 1995–2000. In this data, two variables could potentially serve to re-identify paired infants: birthweight (grams) and Apgar score (a 0–10 assessment of newborn baby health). True pairs of twins might be expected to have one baby larger and healthier than the other. Yet tests of a sample of twins show the pairs' values are correlated (with a Pearson correlation of 0.79 for weight, 0.44 for Apgar), so there is at least some signal for the algorithm to work with.

The second data set is derived from the Reality Mining data, cell phone data collected from 94 students and faculty over a nine-month period (Eagle & Pentland, 2006). Our task instances address the question "Is an individual's phone usage pattern distinctive enough to identify them?" We summarize each user's weekly behavior with seven aggregate features: total communication events; number of distinct contacts; number of calls made, received, and missed; number of SMS's received and sent. Each such person-week becomes a point in a data set, and the pairs are defined as instances of the same individual in two different weeks.

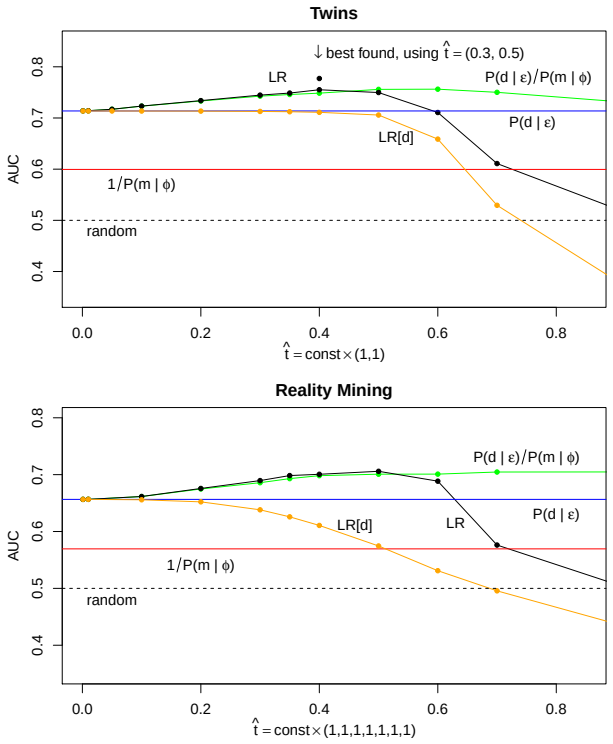

Figure 4. Results (avg. AUC) on real data sets as $\hat{\mathbf{t}}$ varies.

From each data source, we construct 100 labeled instances of the pair detection task. An instance of twins data consists of five pairs of twins and 90 singleton babies. An instance of cell phone data consists of five pairs of person-weeks and 75 singletons. In the experiments below, ϕ is always a normal distribution with diagonal covariance.

6.2. Experiments and Results

Since we know ground truth, we can experiment here with different values of $\hat{\mathbf{t}}$. It has one component for each variable, and for these domains all we know in advance is that pairs should be "close together"—i.e., each component is in the range $(0, 1/\sqrt{2})$. For the two-variable twins data, we explore a grid of possible values. For the seven-variable cell phone data, the exponential state space becomes a problem, so we restrict $\hat{\mathbf{t}}$ to the form $a \cdot (1, 1, \ldots, 1)$ for some constant a.

Figure 4 shows that the methods behave very much the same way on real data as they do on synthetic. As before, Best-$LR > d > m$, and $\frac{P(d|\epsilon)}{P(m|\phi)}$ is an excellent alternative when $\hat{\mathbf{t}}$ is unknown.

The grid search on twins data reveals that when we vary the individual components of $\hat{\mathbf{t}}$, this affects the relative strengths of the variables. For instance, setting $\hat{t}_{weight} = 0.001$ (stringently small) but leaving $\hat{t}_{apgar} = 0.7$ (flexible) is almost equivalent to ranking only by d_{weight}. For a fixed ratio among the components of $\hat{\mathbf{t}}$, the relative strengths of the variables are held constant, and only the balance with m will vary.

As a comparison, we also estimate a best fit $\mathbf{t}$ from a large sample of twins: that $(t_{weight}, t_{apgar}) = (0.33, 0.57)$ is not far from the $\hat{\mathbf{t}} = (0.3, 0.5)$ found by searching. Separate experiments with single variables show that for twins, weight is a strong feature, but Apgar is not. With Reality Mining, the strongest features are number of SMS's sent and number of contacts.

It is not surprising that both these tasks turn out to be difficult given their respective feature sets; in particular, it has been noted that for the Reality Mining data, phone communication is not nearly as consistent as proximity patterns (Eagle et al., 2009). If the trends of Figure 1 generalize to here, then the relatively low AUCs may go hand in hand with the high values of $\hat{\mathbf{t}}$ and the performance boost of LR over $P(\mathbf{d} \mid \epsilon)$.

7. Conclusions

This paper introduces a simple model for the task of distinguishing tightly linked pairs from singleton points, given a mixture of both. This task has not been previously described in a general form, although specific instances have been studied in numerous contexts. From the generative model, we derive a likelihood ratio incorporating both the similarity and rarity of the pairs. A single parameter describing the distances between pairs turns out to govern the task's difficulty; at inference time, this same parameter describes how to trade off a pair's similarity with its rarity. This method always outperforms using only similarity, but in a certain parameter range, similarity turns out to be surprisingly competitive. We discuss how to apply the model to real-world data sets having unknown parameters. In the future, we intend to explore versions of this model for more complex domains.

Acknowledgments

This effort is supported by the National Science Foundation (NSF) under grant 0964094 and by Science Applications International Corporation (SAIC) and DARPA under contract number P010089628. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of NSF, SAIC, DARPA or the U.S. Government.

References

Adamic, L. A. and Adar, E. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, July 2003.

Bejder, L., Fletcher, D., and Bräger, S. A method for testing association patterns of social animals. *Animal Behaviour*, 56(3):719–725, 1998.

Chandola, V., Banerjee, A., and Kumar, V. Anomaly detection: A survey. *ACM Computing Surveys*, 41 (3):1–58, July 2009.

Chaudhuri, S., Ganti, V., and Motwani, R. Robust identification of fuzzy duplicates. In *Proc. 21st Int'l Conf. on Data Engineering (ICDE 2005)*, pp. 865–876. IEEE, April 2005.

Cohen, W. W., Ravikumar, P. D., and Fienberg, S. E. A comparison of string distance metrics for name-matching tasks. In *Proc. IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*, pp. 73–78, 2003.

Committee on DNA Forensic Science: An Update, National Research Council. *The Evaluation of Forensic DNA Evidence*. The National Academies Press, 1996.

Crandall, D. J., Backstrom, L., Cosley, D., Suri, S., Huttenlocher, D., and Kleinberg, J. Inferring social ties from geographic coincidences. *Proceedings of the National Academy of Sciences*, 107(52):22436–22441, December 2010.

Eagle, N. and Pentland, A. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

Eagle, N., Pentland, A. S., and Lazer, D. Inferring friendship network structure by using mobile phone data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, September 2009.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16, January 2007.

Eskin, E. Anomaly detection over noisy data using learned probability distributions. In *Proc. 17th Int'l Conf. on Machine Learning (ICML 2000)*, pp. 255–262, 2000. Morgan Kaufmann.

Fellegi, I. P. and Sunter, A. B. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, December 1969.

Friedland, L. and Jensen, D. Finding tribes: Identifying close-knit individuals from employment patterns. In *Proc. 13th Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2007)*, pp. 290–299, 2007. ACM.

Hand, D. J. Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Machine Learning*, 77(1):103–123, October 2009.

Liben-Nowell, D. and Kleinberg, J. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

McCallum, A., Nigam, K., and Ungar, L. H. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proc. 6th Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2000)*, pp. 169–178, 2000. ACM.

Metwally, A., Agrawal, D., and Abbadi, A. E. Detectives: Detecting coalition hit inflation attacks in advertising networks streams. In *Proc. 16th Int'l Conf. on World Wide Web (WWW 2007)*, pp. 241–250, 2007. ACM.

Narayanan, A. and Shmatikov, V. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, pp. 111–125, 2008. IEEE Computer Society.

National Center for Health Statistics. Matched multiple birth data, 1995–2000. Public-use data file and documentation, 2000. URL `http://ftp.cdc.gov/pub/Health_Statistics/NCHS/Datasets/mmb2/`.

Sorokina, D., Gehrke, J., Warner, S., and Ginsparg, P. Plagiarism detection in arXiv. In *Proc. 6th Int'l Conf. on Data Mining (ICDM 2006)*, pp. 1070–1075, 2006. IEEE Computer Society.

Su, C. and Srihari, S. N. Evaluation of rarity of fingerprints in forensics. In *Advances in Neural Information Processing Systems 23*, pp. 1207–1215, 2010.

Whang, S. and Garcia-Molina, H. Managing information leakage. In *Proc. 5th Biennial Conf. on Innovative Data Systems Research (CIDR 2011)*, pp. 79–84, 2011.

Winkler, W. E. Overview of record linkage and current research directions. Technical report, U.S. Census Bureau, February 2006.

Yang, Z., Wilson, C., Wang, X., Gao, T., Zhao, B. Y., and Dai, Y. Uncovering social network sybils in the wild. In *Proc. Internet Measurement Conf. (IMC 2011)*, pp. 259–268, 2011. ACM.