Using Structured Neural Networks for Record Linkage

Burdette Pixton

Christophe Giraud-Carrier

Record Linkage

- Record Linkage is:
  - the process of identifying similar people
  - a necessary step in exchanging and merging pedigrees

Record Linkage – General Process

- General Process
  - Compare attributes
    - Surname_A vs. Surname_B
    - Use String Metrics (jaro, soundex, etc..)
  - Quantify the comparison (score)
    - Rule-based
    - Use metric score
  - Combine the scores
    - Rule-based
    - Neural Network
  - Compare against a threshold

MAL4:6

- Mining And Linking FOR Successful Information eXchange
  - An automatic approach
  - MAL4:6 uses relationships found in pedigrees
    - Traverses both pedigrees in parallel and measures the similarity of each instance
    - $Individual_A$ vs $Individual_B$ and $Father_A$ vs $Father_B$, etc…

Version 0.1

- Focused on
  - Comparing the attributes
  - Quantifying the comparison
- Naively
  - Combined the scores (Average)
  - Compared against a threshold

Version 0.1

- Similarities are computed using a heterogeneous metric system

Attribute Type	Metric
Gender	Binary Discrimination
Name	Soundex
Location	Jaro
Day	1-norm
Month	Dice
Year	1-norm

Version 0.1 Definitions

- Attributes: A = {A_1,A_2,…A_n}, A_i would be a piece of information (e.g., date of birth)
- For each A_i, sim_{Ai} is the similarity metric associated with A_i
- Let x = < A_1 : a_1^x, A_2 : $a_2^{x,\ldots}$, A_n : a_n^x > denote an individual where a_j^x is the value of A_j for x
  - <firstname: John, lastname: Smith,…>
- Let R= {R_0,R_1,…R_m} be a set of functions that map an individual to one of its relatives
- α_{ij} = {0,1}

$$sim(x,y) = \sum_{i=0}^{p} \sum_{j=1}^{n} \alpha_{ij} sim_{A_j}(A_j^{R_i(x)}, A_j^{R_i(y)})$$

Version 0.1

- Matches:
  - Recall = 94.2%, Precision = 71.8%
- Mismatches
  - Recall = 86.2%, Precision = 98.4%

Version 0.1 Challenges

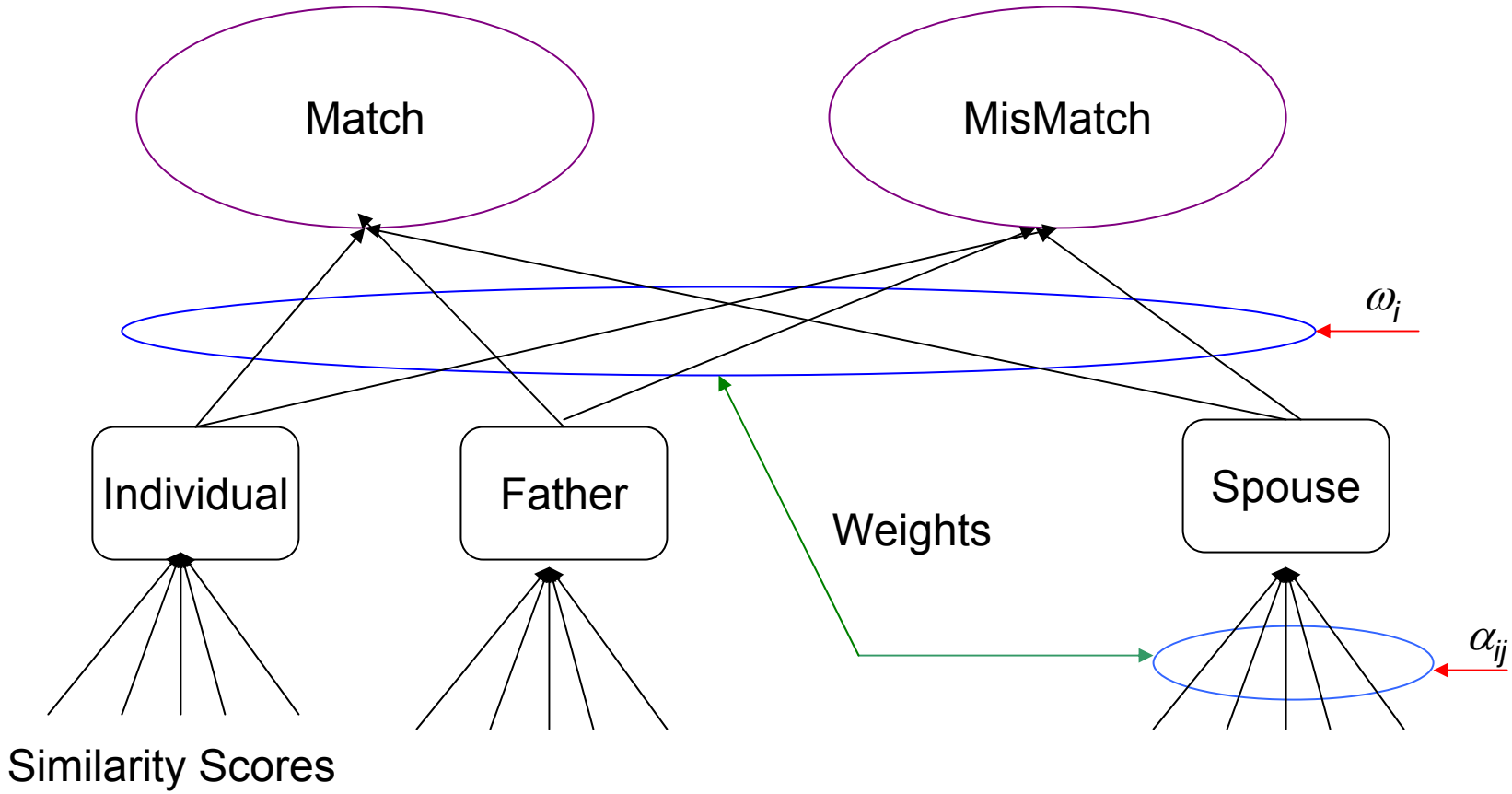
- Each relationship/attribute is treated equally
- Weights
  - Version 0.1 used feature selection instead of continuous weights
  - Weights would allow MAL4:6 to use all of the data in a pedigree to a degree (TBD by MAL4:6)
- Naturally Skewed Data
  - #NonMatches >> #Matches
  - Learners tend to over learn the majority class

Version 1.0 Definitions

- Problem 1: Each relationship/attribute is treated equally
- Attributes: A = {A_1,A_2,…A_n}, A_i would be a piece of information (e.g., date of birth)
- For each A_i, sim_{Ai} is the similarity metric associated with A_i
- Let x = < A_1 : a_1^x, A_2 : $a_2^{x,\ldots}$ A_n : a_n^x > denote an individual where a_j^x is the value of A_j for x
  - <firstname: John, lastname: Smith,…>
- Let R= {R_0,R_1,…R_m} be a set of functions that map an individual to one of its relatives
- ω_i and α_{ij} are continuous

$$sim(x,y) = \sum_{i=0}^{p} \omega_i \sum_{j=1}^{n} \alpha_{ij} sim_{A_j}(A_j^{R_i(x)}, A_j^{R_i(y)})$$

Structured Neural Network Learning Weights (Problem 2)



Blocking/Filtering

- Problem 3: Naturally Skewed Data
- Blocking
  - Typically done on preprocessed data to reduce obvious non-matches
  - Extended Blocking/Filtering
    - Use a series of structured neural networks
    - After each training-testing phase (pass), eliminate “obvious” instances of the majority class

Filtering Definitions

- Let $T = M \cup m$ be the training set, where M is the set of pairs from the majority class and m is the other class
- MATCH(x) is the value of the match output node when x is presented
- MISMATCH(x) for the mismatch output node

Filtering Definitions

- If *q* is a pair to be classified, then its ratio *r* is

$$r = \frac{MATCH(q)}{MISMATCH(q)}$$

- Thresholds

$$\delta_M = \frac{1}{|M|} \sum_{x \in M} \frac{MATCH(x)}{MISMATCH(x)} \qquad \delta_m = \frac{1}{|m|} \sum_{x \in m} \frac{MATCH(x)}{MISMATCH(x)}$$

Filtering Definitions

- If match is the majority class (*M*)
  - An instance is classified as *a match* if $r > \delta_M$
- If mismatch is the majority class (*M*)
  - An instance is classified as *a mismatch* if $r < \delta_M$
- Remaining instances are inputted into a new structured neural network
- When a test instance is classified
  - True/false positive/negative rates are calculated
  - These rates are propagated to future networks
- Each element is classified
  - Elements between the thresholds are classified as *M*
  - Rates from previous networks are computed with current rates to obtain overall performance indicators

Experimental Setup

- Genealogical database from the LDS Church's Family History Department (~5 million individuals)
- ~16,000 labeled data instances
  - Created a training set and test set for distributions of 1:1 and 1:100
  - Pre-blocked (each instance is "close")
  - 1:100 not likely to occur but used for experimental purposes

Balancing the distributions

Original	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5
1:100	1:79.7	1:28.9	1:3.18	---	---
1:1	1:.042	1:4.45	1:2.59	1:1.42	1:2.47

Precision/Recall

	No Filtering	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5
1:100	25.0/ 33.3	70.0/ 33.3	44.4/ 85.7	44.4/ 85.7	--	--
1:1	80.3/ 81.6	91.6/ 85.7	91.4/ 86.7	88.0/ 94.0	88.6/ 93.5	88.9/ 93.8

0.1 vs. 1.0

	Version 0.1	Version 1.0
Distribution	1:3	1:1
Generations	8 (4 up, 4 down)	3 (3 up)
Precision	71.8%	88.9%
Recall	94.6%	93.8%

Future Work

- Structured Neural Networks allow us to look into the “why”
- Compare networks at different distribution layers