Record Linking

Robert Berry
Twitter: @no0p_
August, 2015

Objectives

- Problem definition → Identify
  - Informal problem statement
  - Examples
  - Formal problem statement
- A somewhat general solution → Implementation
  - Logistic regression model
  - Tools: Madlib
  - Evaluation of results
- Practical tips → Successful / Painless Implementation

What is record linking?

- Database Context
  - Two or more records, referencing the same entity, without a key.

- General Statement (Wikipedia)
  - “Record linkage refers to the task of finding records in a data set that refer to the same entity across different data sources. Record linkage is necessary when joining data sets based on entities that may or may not share a common identifier.”

https://en.wikipedia.org/wiki/Record_linkage

Common Experience

Concrete Example: Two Tables

name	address
Roddy Piper	123 Fake St. Portland, OR
Tom Anderson	Wabash and Lake, Chicago
Barry Obama	1600 Pennsylvania, Washington, DC
Ignatius J. Reilly	Baton Rouge
Lennart Sollberg	Stockholm, Sweden
Lady Dulcinea	Spain
Veronica Mars	Neptune, CA
Tricia McMillan	Islington London, UK
sid benegali	Salamanca, Spain
Liz Warren	123 Fake St. Boston, MA

(10 rows)

name	address
Elizabeth Warren	Boston, MA
I. Reilly	New Orleans, LA
sid hamete benegali	Salamanca, Spain
Trillan	London, UK
John Reilly	New Orleans, LA
I. J. Reilly	Chicago, IL
Ig O'Reilly	Houston TX
Roddy Piper	100 Main St. Los Angeles
Dulcinea De Toboso	Toboso, Spain
Ignatius Jacques Reilly	100 12th., Baton Rouge
Ignatius J. Reilly	Louisiana
Barack Obama	1600 Penn. Ave.
Ig Reilly	Baton Rouge
Thomas Anderson	Wabash and Lake, Chicago
Lennart Sollberg	Sweden
Barak Obama	1600 Pennsylvania Ave.
Mars, Veronica	Neptune, CA

(17 rows)

One of the oldest problems facing civilization

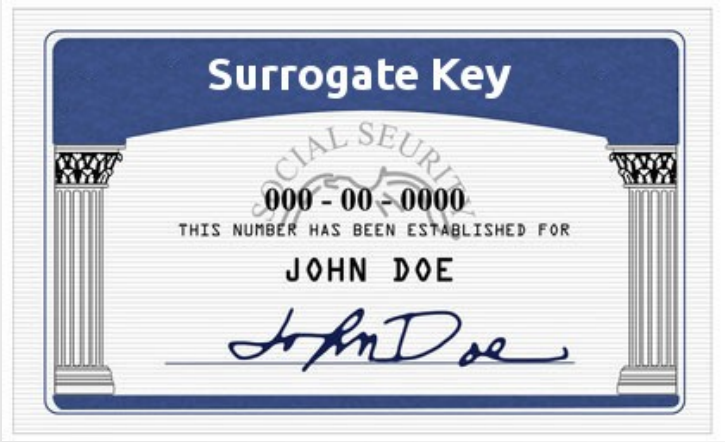
- 1946 article titled "Record Linkage" published in the American Journal of Public Health.
- Census as statecraft
  - Censuses in Egypt are said to have been taken during the early Pharaonic period in 3340 BCE and in 3056 BCE
  - US Census research

How Problem Emerges for Census

- Counting People
- Varied Data Sources:
  - Interviews
  - Mailed Surveys
  - Birth Certificates
  - Marriage Records
  - Death Records

The Many-faced Problem

- Typically Entities in data model
  - Companies
  - Buildings
  - People
  - Documents



- Typically arises from different data sources
  - User entries
  - Transcription errors
  - Divergent data sets (split brain)
  - Application Developers
- No natural Key
- Candidate keys include a set of possible values, including transcription and spelling errors
- Text

Duplicates Case

name	address
I. J. Reilly	Chicago, IL
Roddy Piper	100 Main St. Los Angeles
John Reilly	New Orleans, LA
Tom Anderson	Wabash and Lake, Chicago
Lennart Sollberg	Stockholm, Sweden
Lennart Sollberg	Sweden
sid hamete benegali	Salamanca, Spain
Elizabeth Warren	Boston, MA
Dulcinea De Toboso	Toboso, Spain
Barak Obama	1600 Pennsylvania Ave.
Thomas Anderson	Wabash and Lake, Chicago
Barack Obama	1600 Penn. Ave.
Veronica Mars	Neptune, CA
Ignatius J. Reilly	Louisiana
Tricia McMillan	Islington London, UK
Ig Reilly	Baton Rouge
Ignatius Jacques Reilly	100 12th., Baton Rouge
Ignatius J. Reilly	Baton Rouge
sid benegali	Salamanca, Spain
Liz Warren	123 Fake St. Boston, MA
Roddy Piper	123 Fake St. Portland, OR
Trillan	London, UK
Mars, Veronica	Neptune, CA
I. Reilly	New Orleans, LA
Lady Dulcinea	Spain
Barry Obama	1600 Pennsylvania, Washington, DC
Ig O'Reilly	Houston TX

Name & Conquer

Label the records in table1 as set **A**, and the records in table2 as set **B**.

Define a set of possible matches as **A X B**, such that each pair ($a \in A, b \in B$) is a possible match or non-match.

The set of true pairs is labeled set **T** where ($a = b$), and the set of false pairs is labeled set **F** where ($a \neq b$).

Solve the problem by distinguishing whether a given pair belongs to set **T** or set **F**.

T and **F** are a partition of **A X B**.

Concrete Pairs

a_name	a_address	b_name	b_address	true_pair
Barack Obama	1600 Penn. Ave.	sid benegali	Salamanca, Spain	?
Barack Obama	1600 Penn. Ave.	Tom Anderson	Wabash and Lake, Chicago	?
Barack Obama	1600 Penn. Ave.	Barry Obama	1600 Pennsylvania, Washington, DC	?
Barack Obama	1600 Penn. Ave.	Ignatius J. Reilly	Baton Rouge	?
Barack Obama	1600 Penn. Ave.	Tricia McMillan	Islington London, UK	?
Barack Obama	1600 Penn. Ave.	Roddy Piper	123 Fake St. Portland, OR	?
Barack Obama	1600 Penn. Ave.	Lady Dulcinea	Spain	?
Barack Obama	1600 Penn. Ave.	Lennart Sollberg	Stockholm, Sweden	?
Barack Obama	1600 Penn. Ave.	Liz Warren	123 Fake St. Boston, MA	?
Barack Obama	1600 Penn. Ave.	Veronica Mars	Neptune, CA	?
Barak Obama	1600 Pennsylvania Ave.	Tom Anderson	Wabash and Lake, Chicago	?
Barak Obama	1600 Pennsylvania Ave.	Liz Warren	123 Fake St. Boston, MA	?
Barak Obama	1600 Pennsylvania Ave.	Ignatius J. Reilly	Baton Rouge	?
Barak Obama	1600 Pennsylvania Ave.	sid benegali	Salamanca, Spain	?
Barak Obama	1600 Pennsylvania Ave.	Tricia McMillan	Islington London, UK	?
Barak Obama	1600 Pennsylvania Ave.	Roddy Piper	123 Fake St. Portland, OR	?
Barak Obama	1600 Pennsylvania Ave.	Lennart Sollberg	Stockholm, Sweden	?
Barak Obama	1600 Pennsylvania Ave.	Lady Dulcinea	Spain	?
Barak Obama	1600 Pennsylvania Ave.	Veronica Mars	Neptune, CA	?
Barak Obama	1600 Pennsylvania Ave.	Barry Obama	1600 Pennsylvania, Washington, DC	?
Dulcinea De Toboso	Toboso, Spain	Ignatius J. Reilly	Baton Rouge	?
Dulcinea De Toboso	Toboso, Spain	Liz Warren	123 Fake St. Boston, MA	?
Dulcinea De Toboso	Toboso, Spain	Lady Dulcinea	Spain	?

Binary Classification

- The problem is now stated a binary classification problem which is well understood
- A solution can take the form of a function which returns a probability that a given pair is a true pair.

$$P(truepair) = f(a, b)$$

Need an Automated Solution

$$P(truepair) = f(a, b) \neq$$

Simple Solution: Discriminant Metric

- String distance metrics
  - Levenshtein (contrib/fuzzystrmatch)
  - Tri-gram (contrib/pg_trgm)
  - Jaro-Winkler
    - Returns value between 0 and 1
    - Developed for '95 Census

The Jaro distance d_j of two given strings s_1 and s_2 is

$$d_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases}$$

Where:

- m is the number of *matching characters* (see below);
- t is half the number of *transpositions* (see below).

	D	I	X	O	N
D	1	0	0	0	0
I	0	1	0	0	0
C	0	0	0	0	0
K	0	0	0	0	0
S	0	0	0	0	0
O	0	0	0	1	0
N	0	0	0	0	1
X	0	0	0	0	0

https://en.wikipedia.org/wiki/Jaro–Winkler_distance

Jaro-Winkler Example

a_name	b_name	jaro_winkler
Barack Obama	Barry Obama	0.91
Thomas Anderson	Tom Anderson	0.86
Barack Obama	Lennart Sollberg	0.58
Thomas Anderson	Lennart Sollberg	0.55
Thomas Anderson	Roddy Piper	0.52
Thomas Anderson	Ignatius J. Reilly	0.47
Thomas Anderson	Barry Obama	0.46
Barack Obama	Roddy Piper	0.45
Thomas Anderson	sid benegali	0.45
Barack Obama	Ignatius J. Reilly	0.43
Barack Obama	sid benegali	0.39
Barack Obama	Tom Anderson	0.39

pg_trgm Example

- What is the optimal threshold?
- Only leverages information present in text field

name	address	dist
Ignatius J. Reilly	Louisiana	0.363636
I. J. Reilly	Chicago, IL	0.473684
Ig Reilly	Baton Rouge	0.526316
Ignatius Jacques Reilly	100 12th., Baton Rouge	0.551724
Ig O'Reilly	Houston TX	0.571429
I. Reilly	New Orleans, LA	0.578947
John Reilly	New Orleans, LA	0.636364
Trillan	London, UK	0.96
Roddy Piper	100 Main St. Los Angeles	0.965517
Dulcinea De Toboso	Toboso, Spain	1
Barak Obama	1600 Pennsylvania Ave.	1
Lennart Sollberg	Sweden	1
sid hamete benegali	Salamanca, Spain	1
Barack Obama	1600 Penn. Ave.	1
Elizabeth Warren	Boston, MA	1
Mars, Veronica	Neptune, CA	1
Thomas Anderson	Wabash and Lake, Chicago	1

Black box Logistic Regression

$$P(truepair) = f(a, b)$$

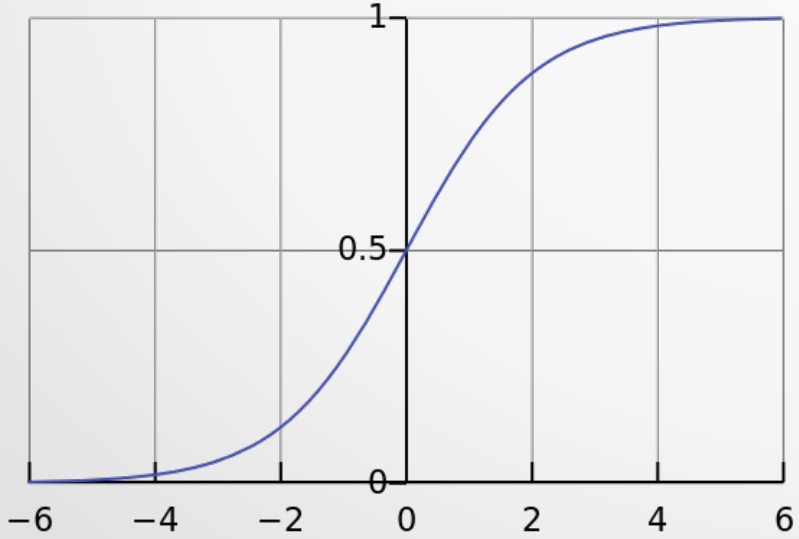
$$P(truepair) = f(x_1, x_2, x_3 \ldots x_n)$$

x_1 = Jaro-Winkler distance of name
x_2 = geospatial distance of addresses
x_3 = Share middle initial

Logistic Regression

$$g(t) = \frac{1}{1 + e^{-t}}$$

$$t = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$



- Binary Classification
- t is a linear function of independent variables
- Returns a probability (number between 0 and 1)
- Widely Implemented
  - Available inside Postgresql through the Madlib extension.
  - http://madlib.net/

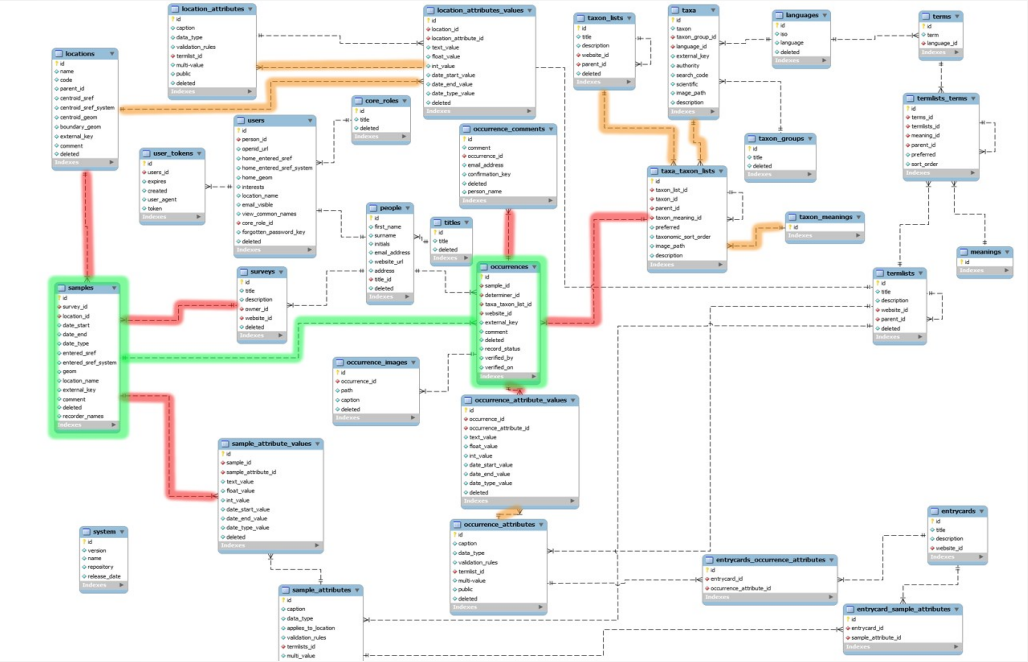
Feature Selection

- Choose a number of independent variables (for pairs)
- Some useful string comparison features

$$t = \beta_0 + \beta_1 x$$

Jaro-Winkler Distance	0-1
Levenshtein Distance	Z
length(Longest Common Substring)	Z
Shared Middle Initial	0/1
Weighted Shared n-grams	R
Shared n-gram features (Suffix/Prefix)	R
Word Shape	0/1
Binned length delta	Z

Pair Features From Relations



Same Company or Organization	1/0
Date of Activity	Z
Average Topic Distance	R
Geospatial Distance	Whatever ST_Distance returns
Same Phone Number	1/0

Installing Madlib

- pgxnclient install madlib
- Source:
  - https://github.com/madlib/madlib
  - Configure; make
  - madpack -p postgres -c user@0.0.0.0/db install

Madlib Modules

- Regression Models
  - Linear
  - logistic
- Tree Models
- Conditional Random Field
- ARIMA
- Latent Dirichlet Allocation
- K-means clustering
- Descriptive Statistics
- Inference
- SVM
- Cardinality Estimators
- Utility Functions
  - Linear algebra operations for arrays as vectors

Create Training and Test Sets

```
robert=# \d modeling.real_humans
        Table "modeling.real_humans"
        Column          |  Type   | Modifiers
------------------------+---------+-----------
 a_id                   | integer |
 b_id                   | integer |
 name_jaro_winkler      | numeric |
 name_trigram_distance  | numeric |
 true_pair              | boolean |
```

- Create a utility to manually add records
- If available, may be possible to match pairs based on a foreign key.
  - E.g. 20% of company_a and company_b records have an IRS TID you can link on, but still have representative typo and alternative spelling examples.

How To Train Your Model

```
SELECT madlib.logregr_train(
  'modeling.real_humans',           -- source table
  'modeling.real_humans_linking',   -- out table
  'true_pair',                      -- dependent variable column name
  'ARRAY[ name_jaro_winkler,
          name_trigram_distance]'   -- independent variable columns
);
```

How To Train Your Model

```
SELECT madlib.logregr_train(
  'modeling.real_humans',           -- source table
  'modeling.real_humans_linking',   -- out table
  'true_pair',                      -- dependent variable column name
  'ARRAY[ name_jaro_winkler,
          name_trigram_distance]'   -- independent variable columns
);
```

$$t = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Making Predictions

```
SELECT madlib.logregr_predict_prob(
         coef,                              -- Column name in model table
         ARRAY[0.91                         -- Name Jaro Winkler
               0.22]                        -- Trigram Distance
       ) as prob
  FROM modeling.real_humans_linking;        -- Model Table
```

Evaluating the Model

		Prediction outcome: p	Prediction outcome: n
actual value	p′	True Positive	False Negative
	n′	False Positive	True Negative

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Results for “banks” with name and address

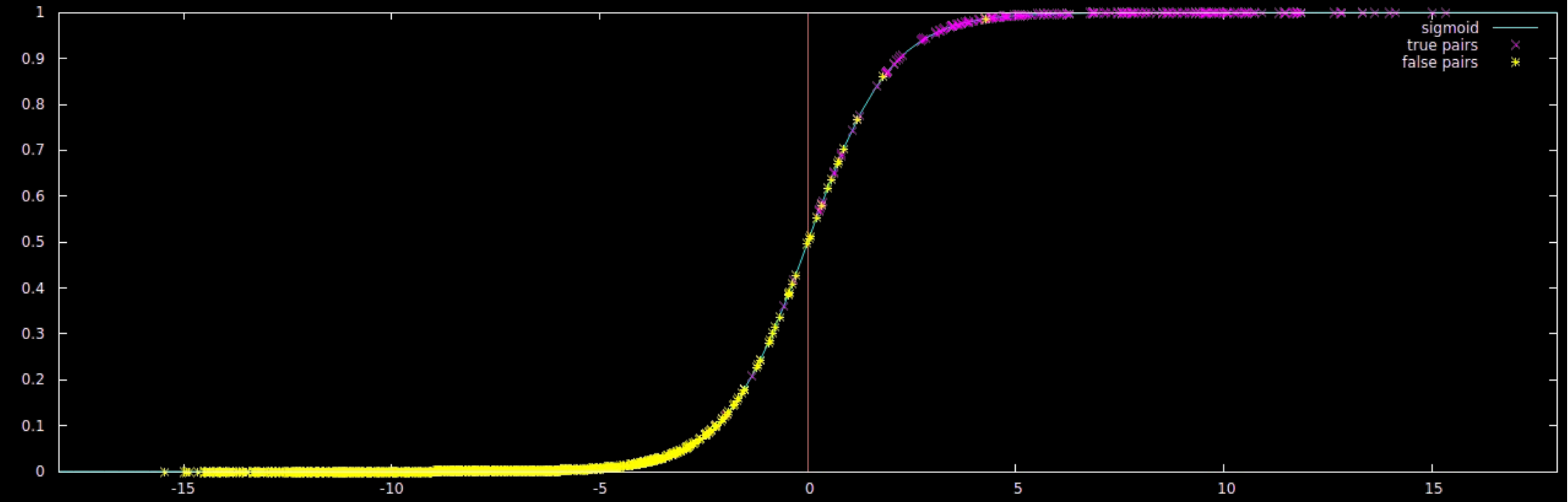
		Prediction outcome: p	Prediction outcome: n
actual value	p′	420	3
	n′	6	1,640

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.995$$

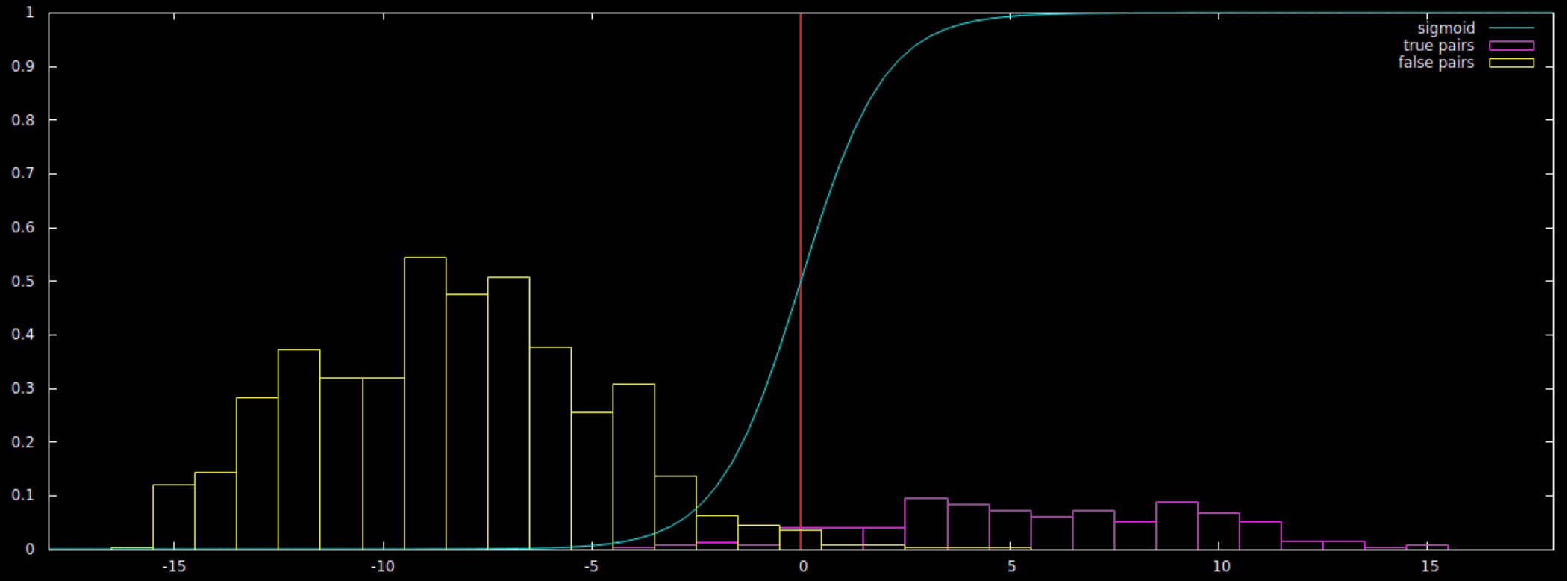
$$Precision = \frac{TP}{TP + FP} = 0.985$$

$$Recall = \frac{TP}{TP + FN} = 0.992$$

Results Points



Results Distribution



Results for Generic Company Model with Names Only

		Prediction outcome: p	Prediction outcome: n
actual value	p′	53	27
	n′	5	3,201

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.99$$

$$Precision = \frac{TP}{TP + FP} = 0.914$$

$$Recall = \frac{TP}{TP + FN} = 0.662$$

Generic Company Predictions

```
A: infineon technologies ag, B: infineon technologies inc -- 0.953189428491674
A: infineon technologies ag, B: infineon technologies austria ag -- 0.82787016683184
A: infineon technologies ag, B: infineon technologies corporation -- 0.750562281738069
A: infineon technologies ag, B: infineon technologies aktiengesellschaft -- 0.58788395802682
```

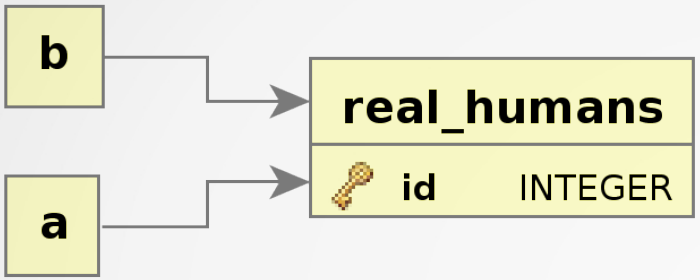
Less Good Generic Company Predictions

```
A: kabushiki kaisha toshiba, B: kabushiki kaisha tohsiba -- 0.434694321755919
A: kabushiki kaisha toshiba, B: kabusshuki kaisha toshiba -- 0.378653943389044
A: kabushiki kaisha toshiba, B: kabushiki kasiha toshiba -- 0.349440007538419
A: kabushiki kaisha toshiba, B: kabushiki kaihsa toshiba -- 0.331319289091398
A: kabushiki kaisha toshiba, B: toshiba kikai kabushiki kaiaha -- 0.368162174220
A: kabushiki kaisha toshiba, B: kabushiki kaisha tec -- 0.575044977255568
A: kabushiki kaisha toshiba, B: kabushiki kaisha tomoku -- 0.346023383356478
A: kabushiki kaisha toshiba, B: kabushiki kaisha topcon -- 0.35648483924697
A: kabushiki kaisha toshiba, B: kaise kabushiki kaisha -- 0.321845187262785
A: kabushiki kaisha toshiba, B: toshiba tec kaubhiki kaisha -- 0.341747682556015
```

Reviewing Model

```
coef                    | {1.38137269782965,-13.6632183471641,-1.76848128264876,0.64766158726839,0.0573998865389614,-0.0312569439210871}
log_likelihood          | -249.350905991252
std_err                 | {0.138741016539159,1.18221393113242,0.150065718029083,0.0648178241900955,0.0100528699209984,0.010743497686263}
z_stats                 | {9.95648390279572,-11.5573146173944,-11.7847120973094,9.99202912718808,5.70980098121681,-2.90938247802235}
p_values                | {2.36271390487209e-23,6.77958569393987e-31,4.68016635121174e-32,1.65165583078925e-23,1.13108361062052e-08,0.003621435
12756168}
odds_ratios             | {3.98036171299827,1.16450046633336e-06,0.170591873027232,1.91106673712689,1.05907923725918,0.969226504220616}
condition_no            | 497.892659837498
num_rows_processed      | 9857
num_missing_rows_skipped | 0
num_iterations          | 11
variance_covariance     | {{0.0192490696703191,-0.152411777926995,-0.0107748318067199,-0.000407861281545256,-0.00037803844307969,-6.80469381585
367e-05},{-0.152411777926995,1.39762977896357,0.0555474037277723,0.000307181489771537,0.000515819222603641,0.00226006285334824},{-0.010774831806
7199,0.0555474037277723,0.0225197197275844,-0.0016087075510683,0.000516958963848291,-0.0010020681965598 9},{-0.000407861281545256,0.0003071814897
71537,-0.0016087075510683,0.00420135033273813,-6.85031548612453e-05,-2.24346225837082e-05},{-0.000378038443079689,0.000515819222603646,0.0005169
58963848291,-6.85031548612453e-05,0.000101060193648514,-5.4170776767714e-05},{-6.8046938158537e-05,0.00226006285334824,-0.00100206819655989,-2.2
4346225837082e-05,-5.4170776767714e-05,0.000115422742534738}}
```

Storing Link Information




```
robert=# \d a
          Table "public.a"
    Column     |  Type   | Modifiers
---------------+---------+-----------
 name          | text    |
 address       | text    |
 real_human_id | integer |
 real_human_pr | numeric |
Foreign-key constraints:
    "a_real_humans_fk" FOREIGN KEY (real_human_id)
```

- Does not require application changes
- Preserves Original Information
- Separates Speculation From Source Data
  - Easy to update
- “Probabilistic Foreign Key” Sounds Cool
- Reasonably Painless Access Pattern
- Duplication and Linking Case
- Pluggable Solutions
- Manual Review

What about that cross product?

- M*N ~ (N^2) pair comparisons to find best match
- Compare only against best candidates
  - Indexable value that can exclude candidates
    - trigram distance
    - Geographic restriction
    - Kmeans cluster id
    - LSH bucket
- Probably OK to have fairly loose comparison set
  - With 100,000,000 records, comparing 100,000,000 * **100** is quite a bit faster than 100,000,000 * 100,000,000.

Practical Tip: like-like comparisons

- Consider linking company pairs.
- Better results if comparing specific subset of companies with targeted features.
  - Banks
  - Blood Banks
- May be worthwhile to first classify records because feasible to easily distinguish between classes.

Practical Tip: Features > 1

- Odds ratios can be difficult to reason about when features are 0 to 1.
- Multiplying by 10 seems to work

Practical Tip: Tuning for Precision

- Precision is TP / (TP + FP)
- May wish to avoid FP
- Can adjust linking operation to only create prob foreign key when P(true pair) > 0.6.

Practical Tip: Organizing

- Leverage Postgresql's data organizing features to organize your training and test data
- Schemas for isolating model tables and training tables
- Materialized views for partitioning examples into training and test sets
  - Tablesample
- Pair surrogate keys allows for generating new features from data relatively painlessly.

Practical Tip: Probabilistic Foreign Key

- Manual review of close calls to minimize human effort
- order by real_humans_pr where real_humans_pr < 0.55

Summary

- Identify Problem as multiple records for same entity without a key
- Set of true pairs in A X B
  - Binary Classification of pairs
- Solve binary classification in Postgresql with Madlib logistic regression module
  - String metric features & features from relations
- Create probabilistic foreign keys
  - Manual review to taste

Final Thoughts

- The world is full of messy data
- May have latent information useful for cleaning the data.

Slides at http://no0p.github.io/record_linking.pdf