Algorithms For Finding The Bad Guys

Richard Minerich, Director of R&D at
@Rickasaurus

BAYARD ROCK

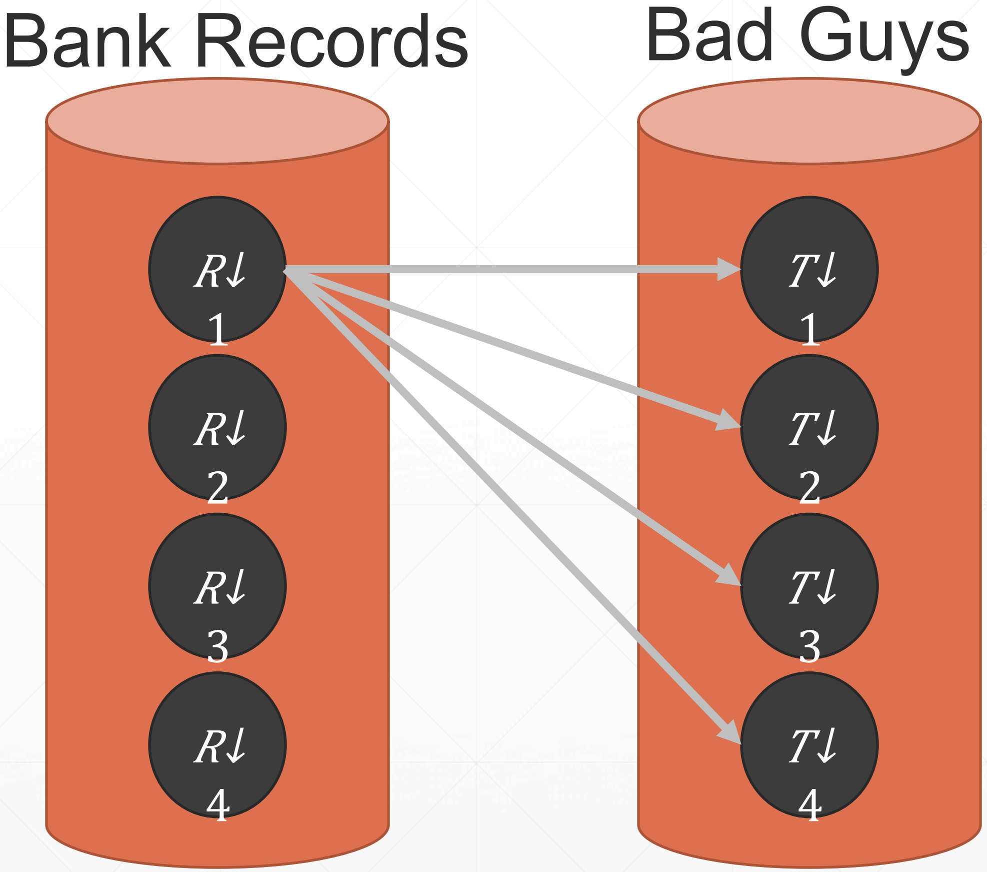
Entity Resolution in Theory

$P(R{\downarrow}n \;\; \mathit{reprents\ the\ same\ entity\ as}\ T{\downarrow}m)$

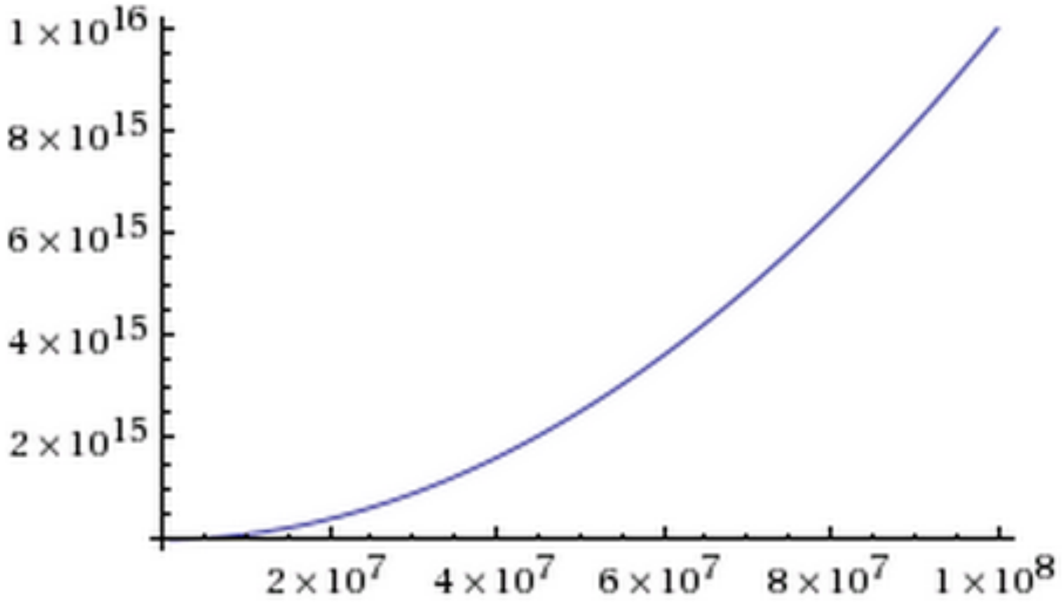
Example Variations:

- Aggregating Products
- Finding Medical Records
- Resolving Paper Authors
- Census
- Finding Bad Guys
- Database Deduping

Different tradeoffs per domain



Too Many Comparisons!



- 100 Million x 100 Million = 10 quadrillion pairs
- 86,400,000 milliseconds per day
- One pair per ms: ~116,000,000 days to compute (~317K years)

How do we beat N*M? Blocking Algorithms.

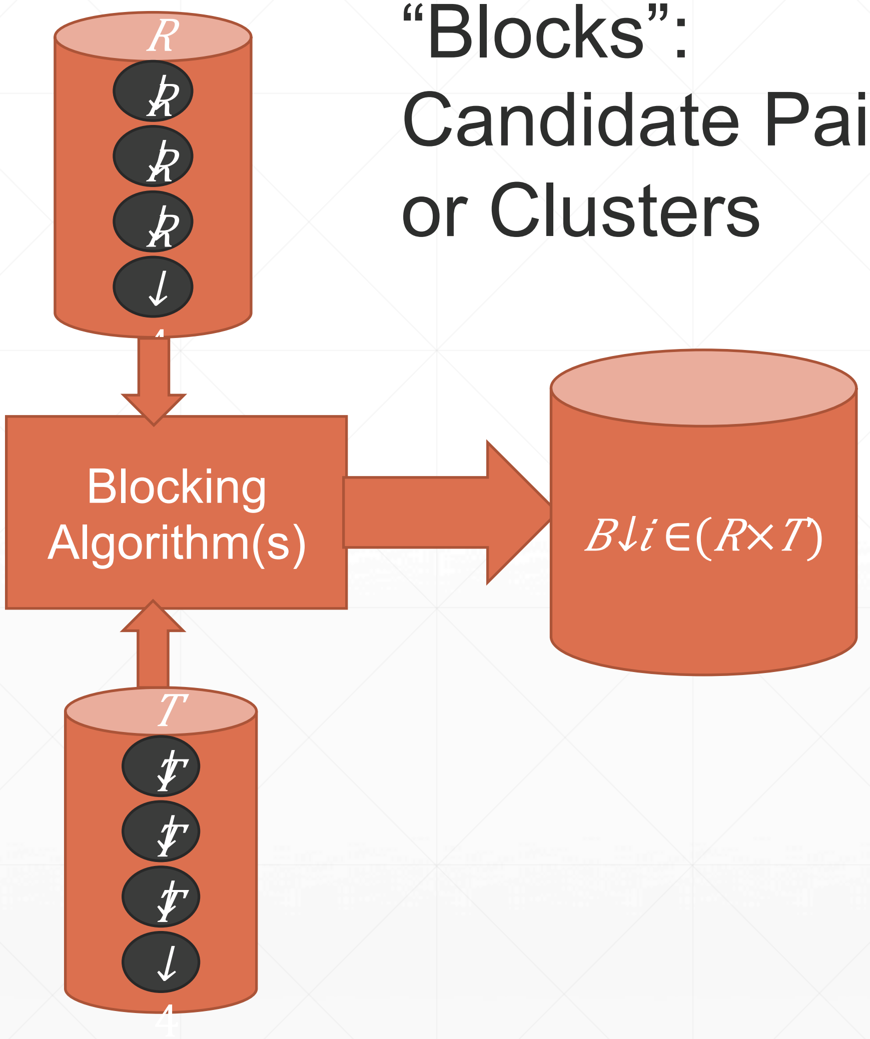
Input:

- Source Records R
- Target Records T

Output:

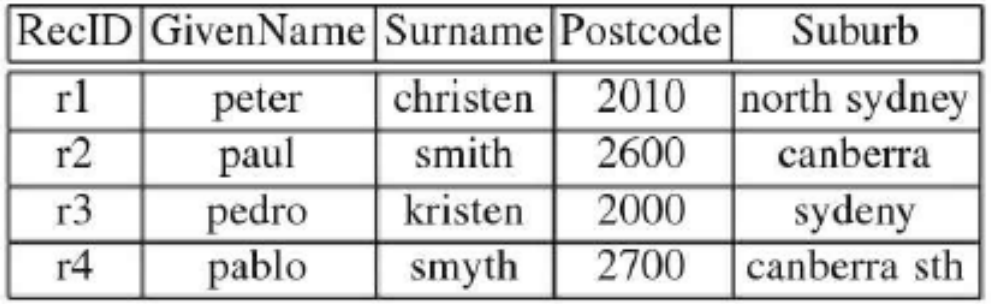
- Blocks of Similar Records

$B_i \in (R \times T)$

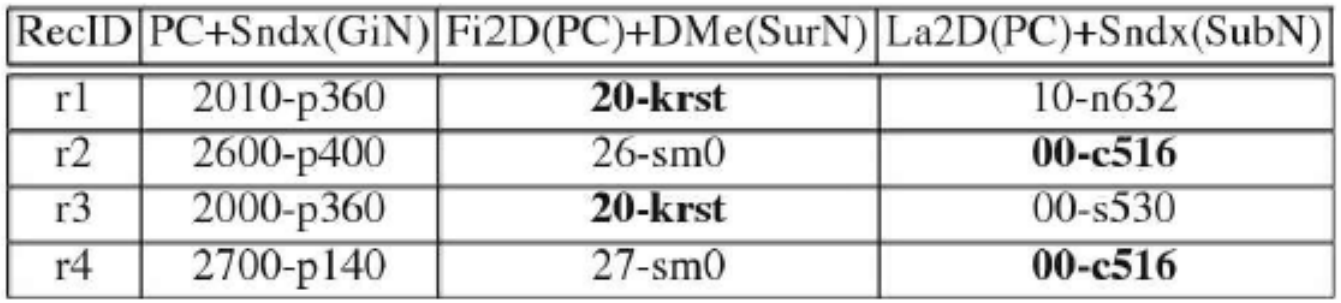


“Blocks”:
Candidate Pairs or Clusters

Simplest: Key-based Blocking



RecID	GivenName	Surname	Postcode	Suburb
r1	peter	christen	2010	north sydney
r2	paul	smith	2600	canberra
r3	pedro	kristen	2000	sydeny
r4	pablo	smyth	2700	canberra sth



RecID	PC+Sndx(GiN)	Fi2D(PC)+DMe(SurN)	La2D(PC)+Sndx(SubN)
r1	2010-p360	**20-krst**	10-n632
r2	2600-p400	26-sm0	**00-c516**
r3	2000-p360	**20-krst**	00-s530
r4	2700-p140	27-sm0	**00-c516**

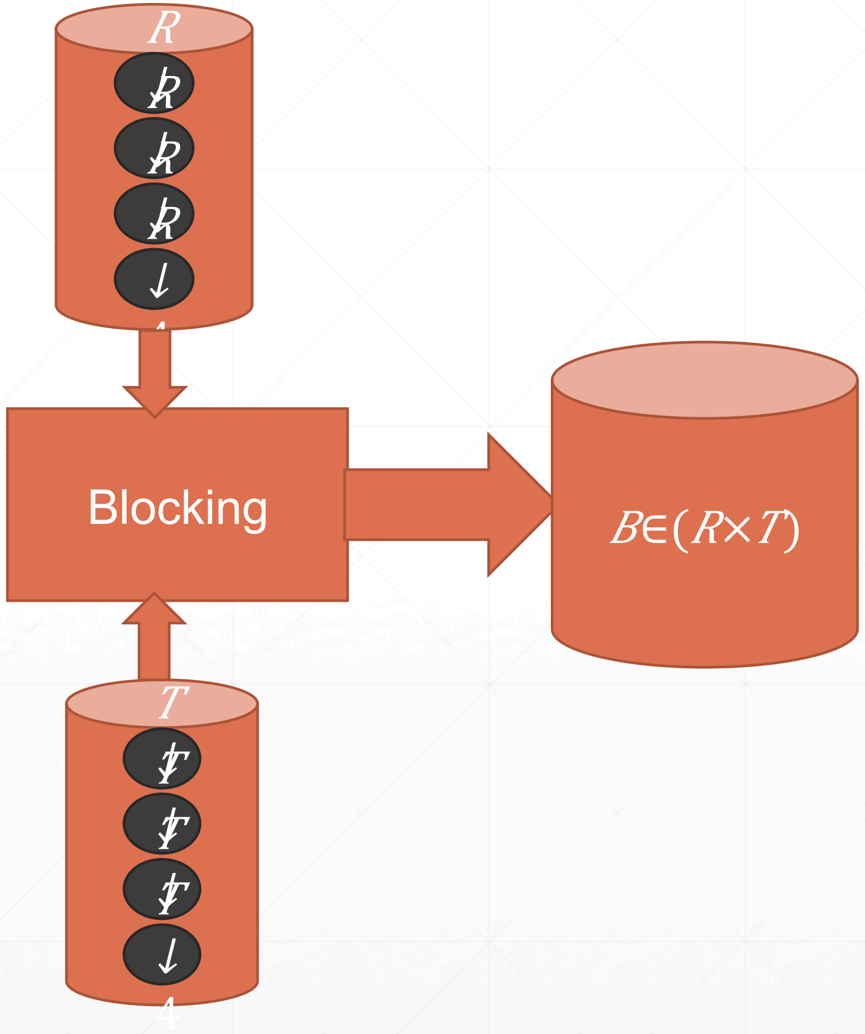
- Nothing's easier than a table lookup!
- Many ways to key, choosing is hard
- Small errors can cause misses
- What about missing data?

Table: Peter Christen - Data Matching 2012

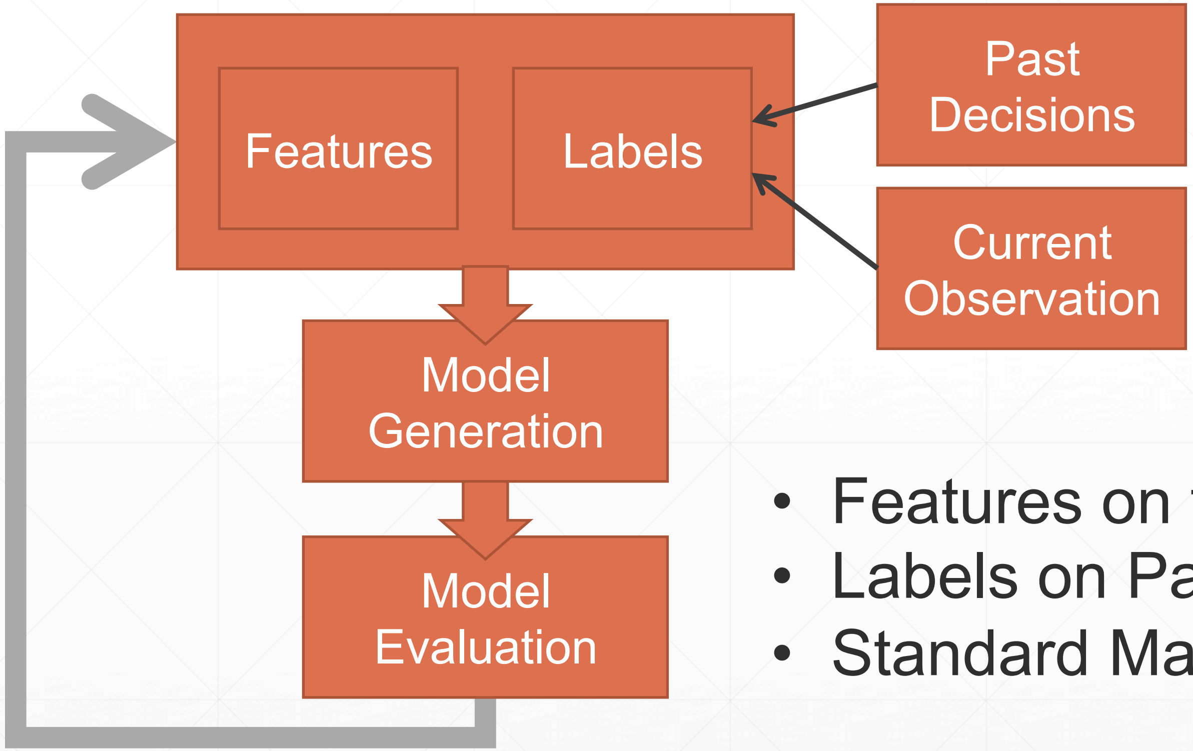
Many ways to Block

- Sorted Neighborhood
- Suffix Arrays/Trees
- Various kinds of Q-gram Indices
- Metric Space Embedding
- Semantic Hashing
- Cluster-based approaches

Best to use a mix.



The Basics of Pairwise Scoring



NAME	LARRY O’ BRIAN
STATE	CANADA
CITY	Montreal
STATE	Quebec
ADDRESS	121 Buffalo Drive
ZIP	H3G 1Z2
DOB	10/24/80

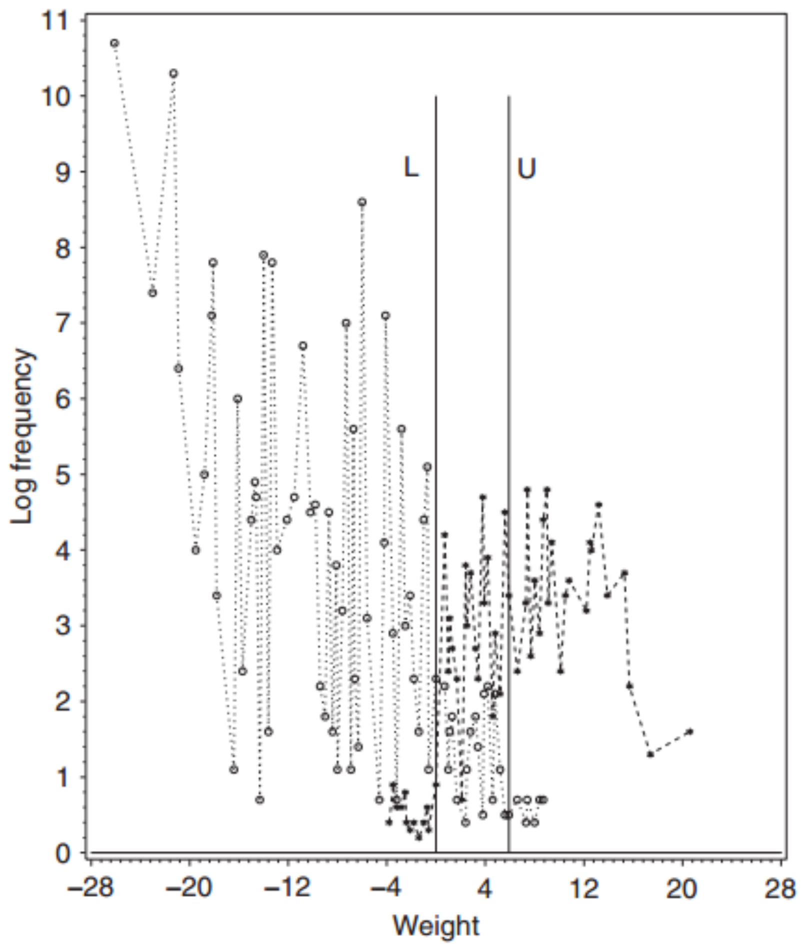
- Features on the Similarity of Fields
- Labels on Pairs are True/False
- Standard Machine Learning Techniques Apply

Simplest:
Empirical Summed Similarity

- F: feature functions (0 .. m) : (a,b) -> [0, 1]
- W: feature weights (0 .. m) : {0+}
- $SimSum(a,b)=\sum_{i=0}^{m} f_i w_i$

Thresholds such that:

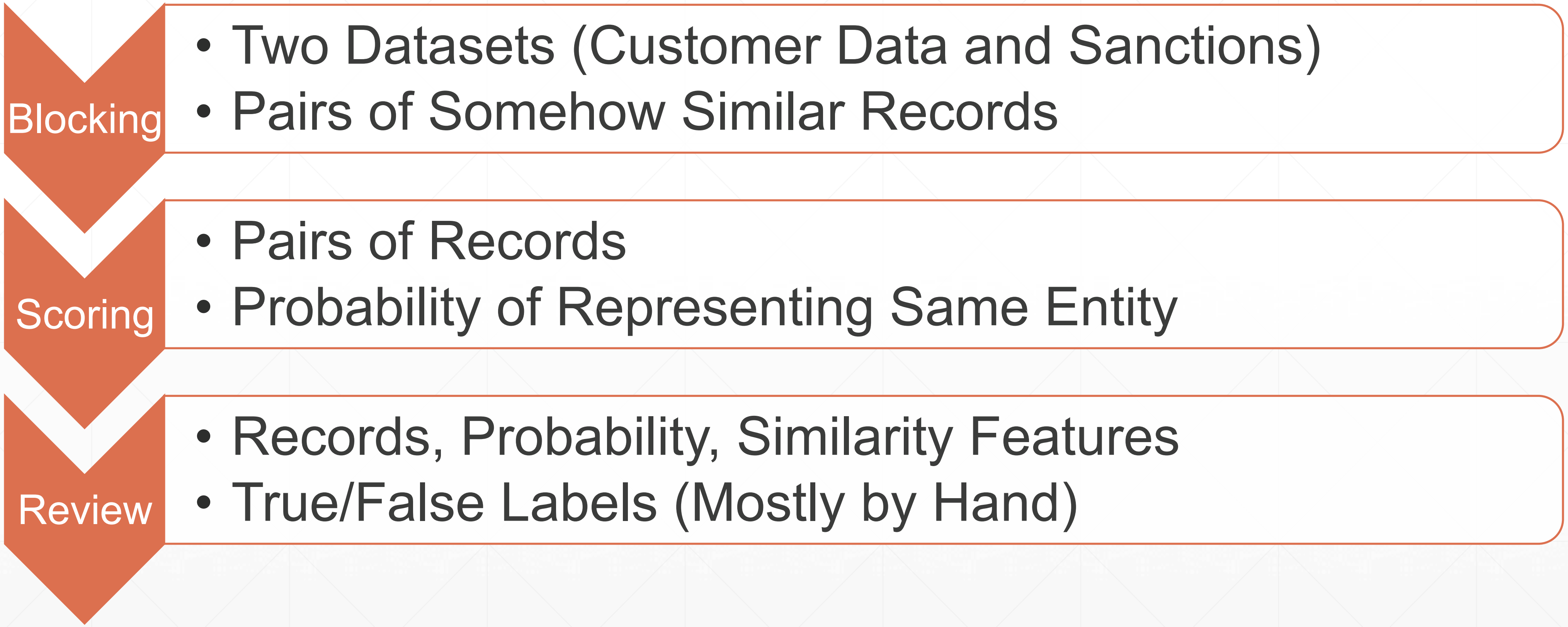
Match: SimSum(a,b) >= Upper
Review: Lower <= SimSum(a,b) <= Upper
Discard: SimSum(a,b) <= Lower



o=nonmatch, *=match
cutoff "L" = 0 and cutoff "U" = 6

Image Via: https://www.cs.umd.edu/class/spring2012/cmsc828L/Papers/HerzogEtWires10.pdf

The Pairwise Entity Resolution Process



Blocking

- Two Datasets (Customer Data and Sanctions)
- Pairs of Somehow Similar Records

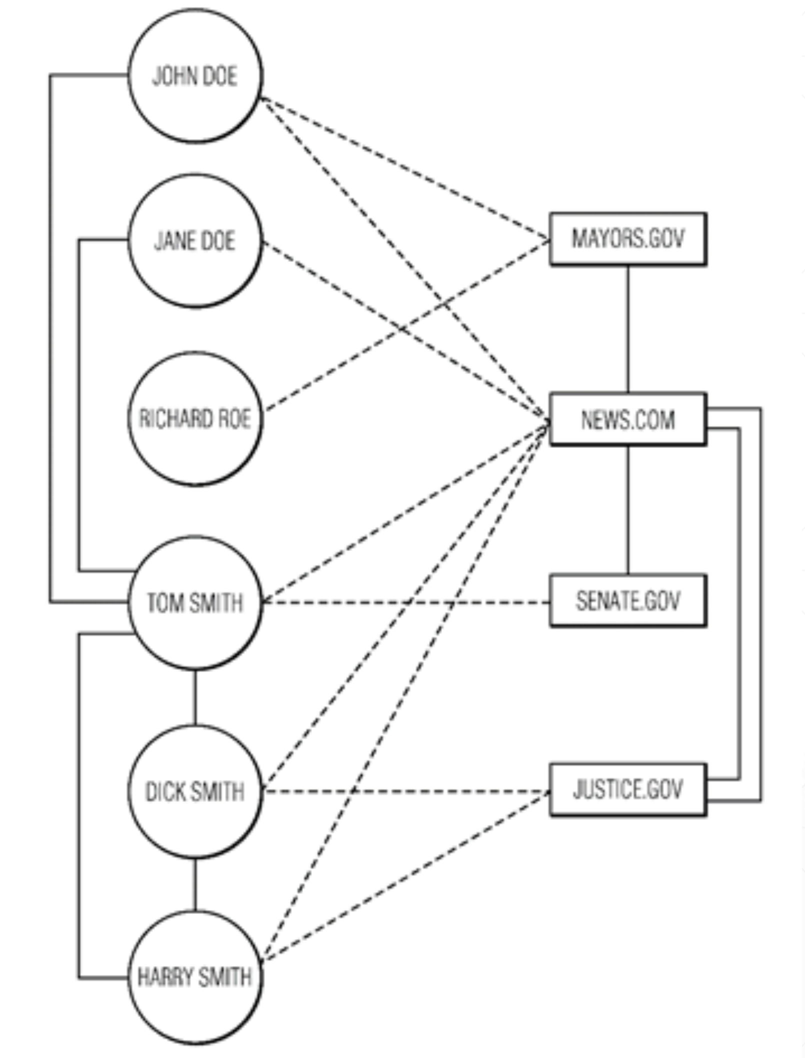
Scoring

- Pairs of Records
- Probability of Representing Same Entity

Review

- Records, Probability, Similarity Features
- True/False Labels (Mostly by Hand)

Another Dimension to aid review: Risk vs Probability



Money Laundering Risk

Same Person Probability

Useful in many domains:

- How Likely To Launder Money?
- How much money do I lose if I get the product wrong?
- Risk of Incorrect Medical Diagnosis

Simplest Ranking?
You Can “Learn to Rank” with Regression.

- The features are the difference in would-be regression features
- The value to predict is the difference in label rank

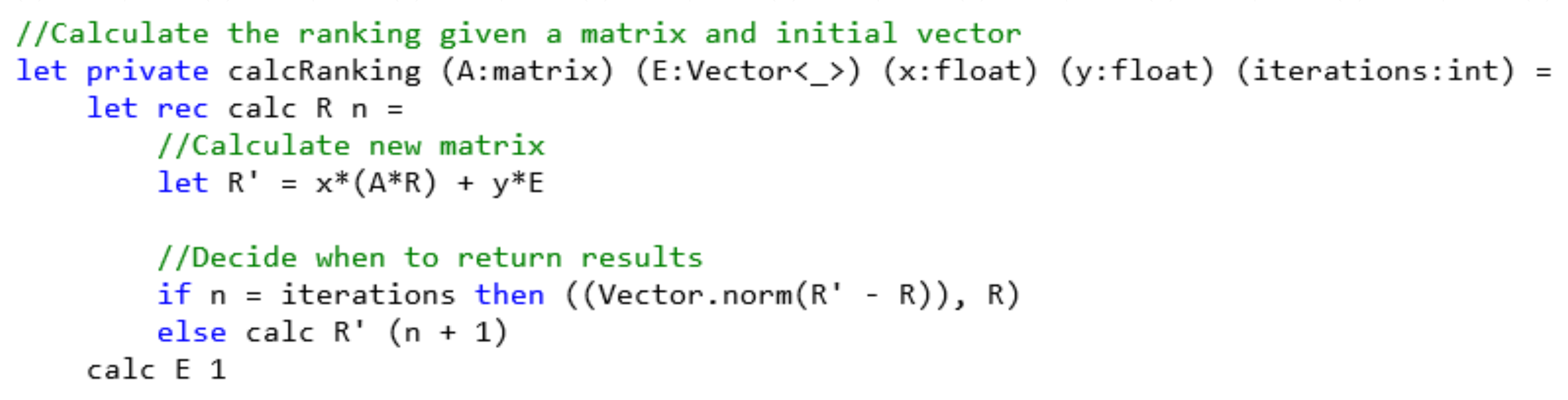
Select 2 labeled samples randomly => (x1,y1) (x2,y2)

$$x = x1 – x2$$
$$y = y1 – y2$$

	Sample 1	Sample 2	Result
Names?	1	1	0
Addresses?	1	0	1
DOB?	0	1	-1
Same Person?	0	0	0

Page Rank: The Easy Parts

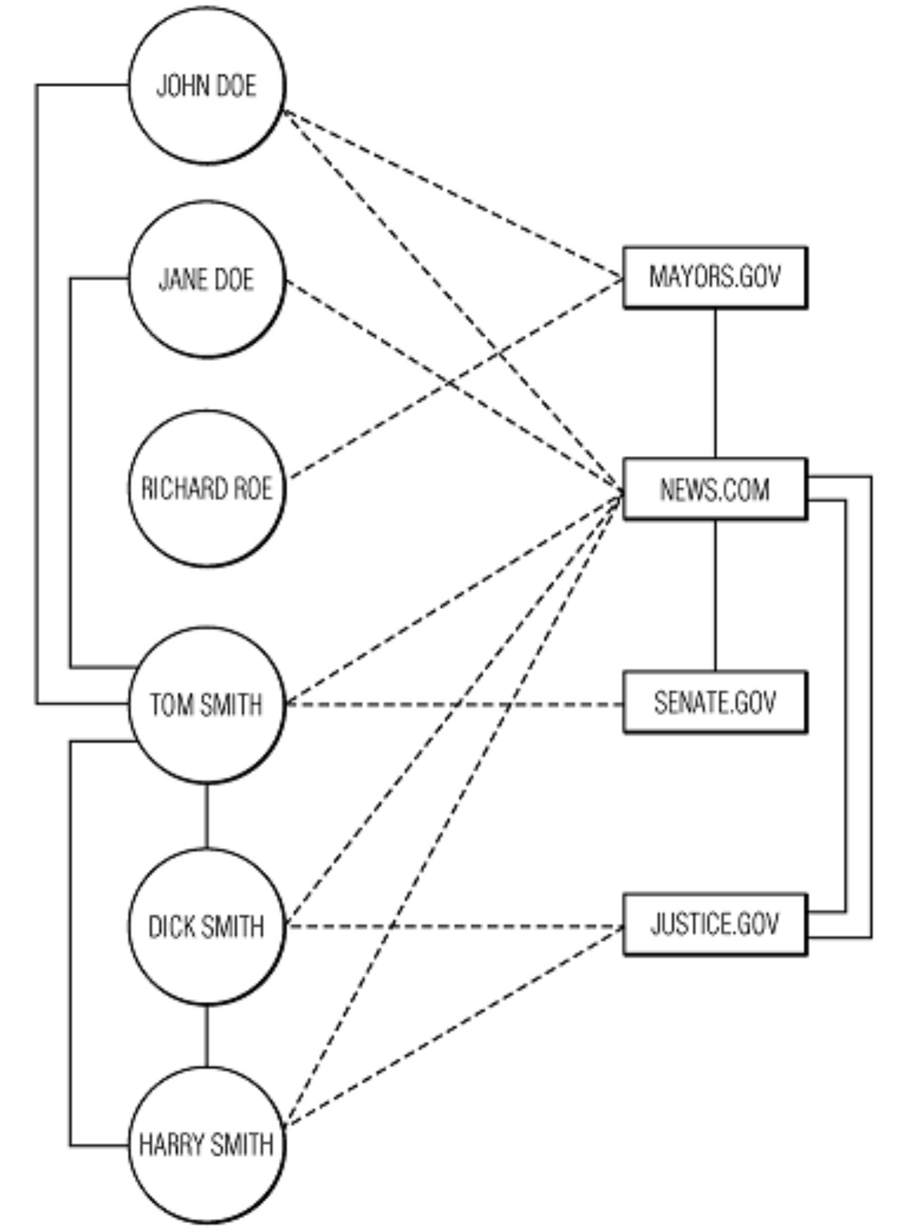
$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$



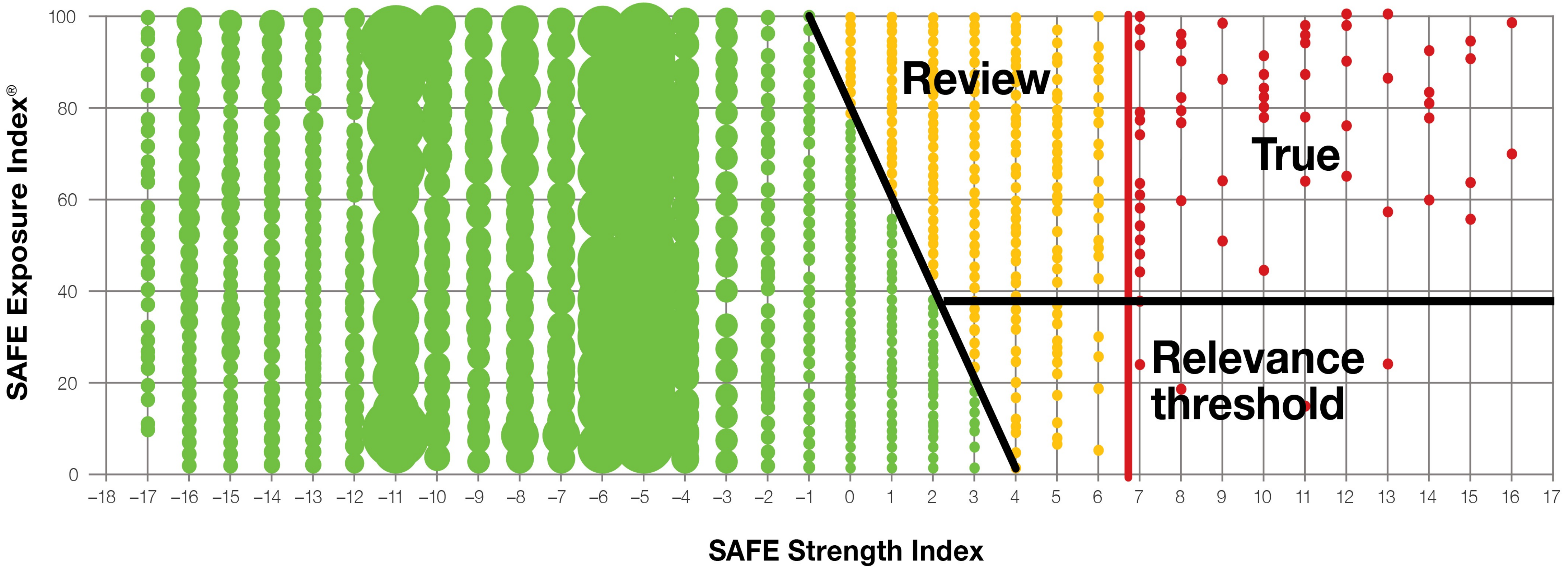
```
//Calculate the ranking given a matrix and initial vector
let private calcRanking (A:matrix) (E:Vector<_>) (x:float) (y:float) (iterations:int) =
    let rec calc R n =
        //Calculate new matrix
        let R' = x*(A*R) + y*E

        //Decide when to return results
        if n = iterations then ((Vector.norm(R' - R)), R)
        else calc R' (n + 1)
    calc E 1
```

Page Rank: The Hard Parts

- Domains, Websites, Pages in Context
- Determining Initial Risk for Sources
- 27 Pages of Data Transformation Code
- Fluctuation with no changes
- Prediction and Explainability



Combining Ranking and Probability: Big Picture



Thank You! Questions?

BAYARD ROCK

You can read more on my blog at:
http://richardminerich.com

Contact me on twitter:
@Rickasaurus

Email me with questions:

rick@bayardrock.com

Check out the NYC F# User Group:
http://www.meetup.com/nyc-fsharp