How We Use FP to Find the Bad Guys

Richard Minerich, Director of R&D at @Rickasaurus
HSBC to pay $1.9bn in US money laundering penalties

HSBC has confirmed it is to pay US authorities $1.9bn (£1.2bn) in a settlement over money laundering, the largest paid in such a case.

A US Senate investigation said the UK-based bank had been a conduit for "drug kingpins and rogue nations".

Money laundering is the process of disguising the proceeds of crime so that the money cannot be linked to the wrongdoing.

HSBC has admitted its money laundering controls have been too lax.
### Aliases

- COURNOYER, COSMO
- COURNOYER, SUPERMAN

### Alternative Spellings

### Occupations

<table>
<thead>
<tr>
<th>Position</th>
<th>Start Date</th>
<th>End Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>the Sinaloa Cartel and Hells Angels Motorcycle Club</td>
<td>2002</td>
<td>Oct 2011</td>
</tr>
<tr>
<td>Alleged kingpin of an international narcotics and money laundering enterprise</td>
<td></td>
<td></td>
</tr>
<tr>
<td>comprised of the Montreal Mafia (Rizzuto Clan)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Company(ies) reported in sources below: (3)

<table>
<thead>
<tr>
<th>UID</th>
<th>Name</th>
<th>Country</th>
<th>EI</th>
</tr>
</thead>
<tbody>
<tr>
<td>466172</td>
<td>SINALOA CARTEL</td>
<td>MEXICO</td>
<td>100</td>
</tr>
<tr>
<td>644456</td>
<td>HELLS ANGELS MOTORCYCLE CLUB</td>
<td>CANADA</td>
<td>100</td>
</tr>
<tr>
<td>1146414</td>
<td>RIZZUTO CLAN</td>
<td>CANADA</td>
<td>100</td>
</tr>
</tbody>
</table>
### Citation Network (Safe View)

#### Information Source(s) (10)

<table>
<thead>
<tr>
<th>R</th>
<th>Date</th>
<th>P</th>
<th>C</th>
<th>F</th>
<th>I</th>
<th>A</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2013/05/29</td>
<td>🌞</td>
<td>🏛</td>
<td>🇺🇸</td>
<td>@</td>
<td></td>
<td>@ Public Access to US Court Electronic Records / USA</td>
</tr>
<tr>
<td>2</td>
<td>2013/01/24</td>
<td>🏛</td>
<td>🇺🇸</td>
<td></td>
<td>@</td>
<td></td>
<td>@ Federal Bureau of Prisons / USA</td>
</tr>
<tr>
<td>3</td>
<td>2013/01/24</td>
<td>🙈</td>
<td>🇺🇸</td>
<td></td>
<td>@</td>
<td></td>
<td>@ The Daily Mail News Online / United Kingdom</td>
</tr>
<tr>
<td>4</td>
<td>2013/01/24</td>
<td>🙈</td>
<td>🇺🇸</td>
<td></td>
<td>@</td>
<td></td>
<td>@ The Montreal Gazette News Online / Canada</td>
</tr>
<tr>
<td>5</td>
<td>2013/01/24</td>
<td>🙈</td>
<td>🇺🇸</td>
<td></td>
<td>@</td>
<td></td>
<td>@ The New York Post News Online / USA</td>
</tr>
<tr>
<td>6</td>
<td>2012/03/21</td>
<td>🙈</td>
<td>🇺🇸</td>
<td></td>
<td>@</td>
<td></td>
<td>@ Canoe 24 Hours Montreal News Online / Canada</td>
</tr>
<tr>
<td>7</td>
<td>2012/03/21</td>
<td>🙈</td>
<td>🇺🇸</td>
<td></td>
<td>@</td>
<td></td>
<td>@ Department of Justice / USA</td>
</tr>
<tr>
<td>8</td>
<td>2012/03/21</td>
<td>🙈</td>
<td>🇫🇷</td>
<td></td>
<td>@</td>
<td></td>
<td>@ The Cyberpress News Online / Canada</td>
</tr>
<tr>
<td>9</td>
<td>2012/03/21</td>
<td>🙈</td>
<td>🇫🇷</td>
<td></td>
<td>@</td>
<td></td>
<td>@ The Laval Courier News Online / Canada</td>
</tr>
<tr>
<td>10</td>
<td>2012/03/21</td>
<td>🙈</td>
<td>🇺🇸</td>
<td></td>
<td>@</td>
<td></td>
<td>@ The New York Post News Online / USA</td>
</tr>
</tbody>
</table>
# Relationship Network (Safe View)

**Reported to be linked to: (12)**

<table>
<thead>
<tr>
<th>UID</th>
<th>Name</th>
<th>Country</th>
<th>Relationship</th>
<th>EI</th>
</tr>
</thead>
<tbody>
<tr>
<td>18723</td>
<td>GUZMAN LOERA, Joaquin</td>
<td>MEXICO</td>
<td></td>
<td>100</td>
</tr>
<tr>
<td>240136</td>
<td>CHAICHIC, German</td>
<td>USA</td>
<td></td>
<td>93</td>
</tr>
<tr>
<td>1281023</td>
<td>COURNOYER, Luc Normand</td>
<td>CANADA</td>
<td></td>
<td>83</td>
</tr>
<tr>
<td>1445785</td>
<td>CURATOLA, Dominick Vincent</td>
<td>USA</td>
<td></td>
<td>73</td>
</tr>
<tr>
<td>1445789</td>
<td>DECREASENZO, Michael</td>
<td>USA</td>
<td></td>
<td>71</td>
</tr>
<tr>
<td>1700695</td>
<td>RACINE, Mario Oliver</td>
<td>CANADA</td>
<td></td>
<td>95</td>
</tr>
<tr>
<td>1700698</td>
<td>PAISSE, Patrick</td>
<td>CANADA</td>
<td></td>
<td>93</td>
</tr>
<tr>
<td>1700709</td>
<td>CASTILLO MEDINA, Jose Alejandro</td>
<td>CANADA</td>
<td></td>
<td>77</td>
</tr>
<tr>
<td>1704437</td>
<td>GALEBI, Bobby</td>
<td>UNKNOWN</td>
<td></td>
<td>55</td>
</tr>
<tr>
<td>1939824</td>
<td>TALONI, Alessandro</td>
<td>USA</td>
<td></td>
<td>76</td>
</tr>
<tr>
<td>1939846</td>
<td>VENIZELOS, John</td>
<td>USA</td>
<td></td>
<td>91</td>
</tr>
<tr>
<td>2032556</td>
<td>TASCHEITI, John R</td>
<td>USA</td>
<td></td>
<td>55</td>
</tr>
</tbody>
</table>
SAFE BANKING SYSTEMS
PYRAMID OF RISK

Political Exposure
- Chiefs of State
- National
- State/Provincial
- Large City
- Local/Municipal

Law Enforcement Exposure
- Major Law Enforcement Action
- Minor Disciplinary Action

Sanction Lists
- International
- National

Media Exposure
- Trade/Industry
- Regional
- Local
Anatomy of Anti-Money Laundering

- Onboarding
- Bank
- Other Bank
- News
- Sanctions
- Watch Lists
- Bad Guy Database
- Real time Lookup (Efficient Search)
- Transaction Monitoring (Sparse Information)
- Batch Scanning (O(N*M) Result Space)
- Risk Calculation
Onboarding

- When a customer opens an account at a bank an agent does a search
  - As it is done by a human, errors and missing information are common
  - Low risk process as bad guys may be caught in the batch scanning later
- Blocking data structure is kept loaded into memory and queried against
- Results above some probability threshold are returned to the user ordered by probability and risk
Transaction Monitoring

- SWIFT messages are passed on the internet of money
  - Banks must process huge numbers of these
  - Account information is often not accessible
  - Messages are low information compared to accounts
  - Messages must leave within 24h of being received

- Similar to Onboarding (but huge numbers and time constraints):
  - Blocking data structure is kept loaded into memory and queried against
  - Results below some probability/risk threshold are discarded
  - Hits are manually reviewed in order by probability, risk and timeliness
Batch Scanning

- Initially, all customer records vs all bad guy records.
  - Often hundreds of millions of customer records vs ~3 million bad guy records

- Incrementally what we call Diff-Diff
  - All Customer records vs changed bad-guy records
  - Changed Customer records vs all bad-guy records

- Computation is distributed across many beefy machines
  - ~1TB of ram, 32 Cores

- Results are viewed in order by probability and risk with some thresholding on very lower probability or very low risk
Entity Resolution in Theory

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

Different tradeoffs per domain

\[ P(R_n \text{ represents the same entity as } T_m) \]
The Pairwise Entity Resolution Process

**Blocking**
- Two Datasets (Customer Data and Bad Guys Data)
- Pairs of Somehow Similar Records

**Scoring**
- Pairs of Records
- Score/Probability of Representing Same Entity

**Review**
- Records, Probability, Similarity Features
- True/False Labels (Mostly by Hand)
Why Blocking?

- 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)

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**

<table>
<thead>
<tr>
<th>RecID</th>
<th>GivenName</th>
<th>Surname</th>
<th>Postcode</th>
<th>Suburb</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>peter</td>
<td>christen</td>
<td>2010</td>
<td>north sydney</td>
</tr>
<tr>
<td>r2</td>
<td>paul</td>
<td>smith</td>
<td>2600</td>
<td>canberra</td>
</tr>
<tr>
<td>r3</td>
<td>pedro</td>
<td>kristen</td>
<td>2000</td>
<td>sydney</td>
</tr>
<tr>
<td>r4</td>
<td>pablo</td>
<td>smyth</td>
<td>2700</td>
<td>canberra sth</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>RecID</th>
<th>PC+Sndx(GiN)</th>
<th>Fi2D(PC)+DMc(SurN)</th>
<th>La2D(PC)+Sndx(SubN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>2010-p360</td>
<td>20-krst</td>
<td>10-n632</td>
</tr>
<tr>
<td>r2</td>
<td>2600-p400</td>
<td>26-sm0</td>
<td>00-c516</td>
</tr>
<tr>
<td>r3</td>
<td>2000-p360</td>
<td>20-krst</td>
<td>00-s530</td>
</tr>
<tr>
<td>r4</td>
<td>2700-p140</td>
<td>27-sm0</td>
<td>00-c516</td>
</tr>
</tbody>
</table>

Table: Peter Christen - Data Matching 2012
Canopy Clustering

1. Start with a set (S) of all records
   - Some cheap distance metric \( f(r_1, r_2) : \{0, 1\} \)
   - Some upper bound \( u < 1 \)
   - Some lower bound \( l < 1 \)

2. Take one out (c) and put it in a new cluster (C)

3. For each record still in (S) compare it to (c) via function (f)
   - if it’s higher than \( u \), add it to (C) and remove it from (S)
   - if higher than \( l \), add it to (C) and leave it in (S)

4. If (S) is not empty, go to 2
Many ways to Block

- Sorted Neighborhood
- Suffix Trees/Arrays
- Metric Space Embedding
- Semantic Hashing
- Cluster-based approaches like Swoosh

Most are terrible in their own way. So use a mix.
The Pairwise Entity Resolution Process

Blocking
- Two Datasets (Customer Data and Bad Guys Data)
- Pairs of Somehow Similar Records

Scoring
- Pairs of Records
- Score/Probability of Representing Same Entity

Review
- Records, Probability, Similarity Features
- True/False Labels (Mostly by Hand)
The Basics of Pairwise Similarity Scoring

- Smart Features and Clean Labels are Most Important
- Understandability is Key
- Inference Algorithm is Secondary
Simplest: Empirical Summed Similarity

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

Thresholds such that:

**Match**: $SimSum(a,b) \geq \text{Upper}$

**Review**: $\text{Lower} \leq SimSum(a,b) \leq \text{Upper}$

**Discard**: $SimSum(a,b) \leq \text{Lower}$

Inference for learning feature weights

- Logistic Regression
- Support Vector Machines
- Bayesian Networks/Probabilistic Graphical Models
- Neural Networks
- Random Forests

Complex models are harder to explain than complex features
Pairwise Probability Distribution

Upper Threshold 161

161,358

Tiny Bump 937
The Pairwise Entity Resolution Process

**Blocking**
- Two Datasets (Customer Data and Bad Guys Data)
- Pairs of Somehow Similar Records

**Scoring**
- Pairs of Records
- Score/Probability of Representing Same Entity

**Review**
- Records, Probability, Similarity Features
- True/False Labels (Mostly by Hand)
Overall Model: Risk vs Probability

Money Laundering Risk

Same Person Probability
Page Rank: The Easy Parts

\[ R = \left[ \begin{array}{c} \frac{(1-d)}{N} \\ \frac{(1-d)}{N} \\ \vdots \\ \frac{(1-d)}{N} \end{array} \right] \left[ \begin{array}{cccc} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & \vdots \\ \vdots & \ddots & \ell(p_i,p_j) \\ \ell(p_N,p_1) & \cdots & \ell(p_N,p_N) \end{array} \right] + dR \]

//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

Not Hard: The Algorithm
Normalizing Page Rank for Humans

![Graph](image)

*Fig. 4. PageRank distribution of the Web Base crawl.*

Power Law Picture: [Donato et. al., 2004](#)
Combining Ranking and Probability: Big Picture
Typed Functional Programming is a natural fit

- Small components (i.e. functions) can be independently tested and reused
- Changes are unlikely to break other parts of the system (~3 bugs in 5+ years)
- Code locality eases understanding of complex components
- Huge code reduction over standard object-oriented approaches
- Math reads like math, with proper operators and order of operation
Stages as (simplified) functions

<table>
<thead>
<tr>
<th>Stage</th>
<th>Function Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cleaning (map)</td>
<td>Rec -&gt; Rec</td>
</tr>
<tr>
<td>Blocking (half mapped)</td>
<td>PRec sequence -&gt; CRec sequence -&gt; (CRec, PRec) sequence</td>
</tr>
<tr>
<td>Featurization (map)</td>
<td>(CRec, PRec) -&gt; float Vector</td>
</tr>
<tr>
<td>Scoring (map)</td>
<td>float Vector -&gt; Probability</td>
</tr>
<tr>
<td>Slicing (map)</td>
<td>(Risk, Probability) -&gt; Class Label</td>
</tr>
</tbody>
</table>
Disgustingly Bad but Fairly Large Datasets

- Both Wide (many fields) and Tall (many records)
- From different systems (with different encodings)
- Missing data
- Poorly merged data
- Extra data
- Non-unique IDs

Every client is awful in a completely different way.

<table>
<thead>
<tr>
<th>NAME</th>
<th>LARRY O BRIAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>STATE</td>
<td>CANADA</td>
</tr>
<tr>
<td>CITY</td>
<td>121 Buffalo Drive, Montreal,</td>
</tr>
<tr>
<td></td>
<td>Quebec H3G 1Z2</td>
</tr>
<tr>
<td>ADDRESS</td>
<td>NULL</td>
</tr>
<tr>
<td>ZIP</td>
<td>12345</td>
</tr>
<tr>
<td>DOB</td>
<td>10/24/80; 1/1/1979</td>
</tr>
</tbody>
</table>
Functions on Record Tree Structure

Blocked Pair

- Names
- DOBs?
- Countries?
- States?
- Cities?

... List Record

- Names
- DOBs?
- Countries?
- States?
- Cities?

- ...

Cust Record

- Blocked Pair

- Hit.Cust.Names
  | > stripAccents
  | > oneToMany "nicknames.csv"

- List Record

- stripAccents
- stripCharacters
- replaceSubstring
- oneToManyFromFile
- isLocalCountry
<CleaningFunctionType("Map")>
<Description("Converts the specified characters to spaces")>

let spaceCharacters
    ([<Description("The characters to convert to spaces")>] toSpace: char [])
    ([<Description("The string to manipulate")>] inStr: string) =

    let sb = new StringBuilder()
    for c in inStr do
        if Array.IndexOf(toSpace, c) <> -1 then
            if sb[sb.Length - 1] <> ' ' then sb.Append(' ') |> ignore
            else sb.Append(c) |> ignore
        sb.ToString()
Fighting Bad Data with Configurable Functional Subsystems

```
"InputCleaning": {
  "Basic": [
    {
      "Field": "Cust.Names",
      "Operation": "stripAccents",
      "Args": []
    },
    {
      "Field": "Cust.Names",
      "Operation": "removeCharacters",
      "Args": [ ":", "\"" ]
    },
    {
      "Field": "Cust.Names",
      "Operation": "spaceCharacters",
      "Args": [ ":", ",", ",", ",", ":\", ":" ]
    }
  ]
}
```
Barb, a simple .net record query language (We use it for data cleaning and features)

Name.Contains "John" and (Age > 20 or Weight > 200)

https://github.com/Rickasaurus/Barb
Barb for Cleaning, Queries, and Features on the Fly
Thank You! Questions?

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

Code on Github:
http://github.com/BayardRock
http://github.com/Rickasaurus
Suffix Trees/Arrays

Input length: n, Search length: m

- Construction in O(n) KS[2003]
- Search in O(m) AKO[2004]
- Space is 4n Bytes Naively
- Compressed Space $O(n^*H(T)) + o(n)$
  Where T is the input text

Newer: Compressed Compact SA

How to Introduce Fuzziness?
Rotational Token Alignment

- Less forgiving than Gale-Shapley but also less prone to egregious errors.
- Also known as cyclic suborders of size $k$ of a cyclic order of size $n$
- $O(k(n \text{ choose } k))$

<table>
<thead>
<tr>
<th>Richard</th>
<th>Thomas</th>
<th>Minerich</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minerich</td>
<td>Richard</td>
<td>Thomas</td>
</tr>
<tr>
<td>Thomas</td>
<td>Minerich</td>
<td>Richard</td>
</tr>
</tbody>
</table>
Rotational Alignment (cont.)

- Pre-calculate matrix of $f(x,y)$ values
- Gosper’s Hack for Fast Rotations

```c
int set = (1 << k) - 1;
int limit = (1 << n);
while (set < limit) {
    doStuff(set);

    // Gosper's hack:
    int c = set & -set;
    int r = set + c;
    set = (((r^set) >> 2) / c) | r;
}
```

Gosper’s Hack via: http://programmers.stackexchange.com/questions/67065/whats-your-favorite-bit-wise-technique
Algorithms for Awful Data: String Matching

- Goal: Robust and Forgiving with the Fewest Possible Assumptions

Somewhat Reasonable Data: Rotational Alignment
Extremely Awful Data: Gale-Shapley
Baseline Function: Jaro-Winkler

\[ d_j = \frac{1}{3} \left( \frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m - t}{m} \right) \]

\[ d_w = d_j + (lp(1 - d_j)) \]

- \( m \) = number of matching characters
- \( t \) = number of character transpositions
- \(|s_1|\) = length of the first string
- \(|s_2|\) = length of the second string
- \( l \) = number of characters that match at the start of the string over the number considered
- \( p \) = proportion of the score given to the initial character matching

We use further tweaks on top of this for improved effectiveness.
Gale-Shapley for Stable Marriages: O(n^2)

Input: beau tokens m in M, belle tokens w in W, comparison function f

UM as the unattached beau, UW as unattached belle, P as the pair set (empty)

1) Select a beau m from UM

2) m selects the w in W s.t. f(m,w) is maximized and not previous selected by m

3) If w is in UW, remove m from UM and w from UW and add (m,w) to P
   if a pair (m’, w) exists and f(w,m) > f(w,m’) then
     remove (m’,w) from P, add m’ to UM, add (m,w) to P

4) If UM is not empty, go to 1
Canopy Clustering

- Wait, isn’t this O(n^2)?
- How do we pick the thresholds?
- What might we miss if upper threshold is less than 1?

Other approaches:
- Different functions for (u) and (l)
- Inverted Indices
Expectation Maximization for log odds via Fellegi-Sunter

Pros:

▪ Robust to missing data
▪ Easy enough to understand and well known

Cons:

▪ Needs careful sampling due to class imbalance.
▪ Starting probabilities need to be chosen carefully (local optimization).
Blocking: Industry Concerns

- Can we predict what will and won’t block with absolute certainty?
- Can it find matches across different fields, or in blobs of text?
- Can we improve the process if we find counterexamples?
- Will standard human errors mess up blocking?
- Does it scale to large data sizes on reasonable local hardware?
Directions for Future Research

▪ Record pair population estimation
▪ Safe partial inference for tuning
▪ Prediction of future risk
▪ Collective entity resolution
▪ Mixed entity resolution-fraud detection models