Fastlink

Haocheng, Kewei, Zaolin, Chuhan

Introduction

- FastLink is a scalable entity resolution methodology designed for merging large-scale datasets.
- Advantages
  - Addresses challenges such as missing data, measurement errors, and uncertainty in the merging process
  - Provide more flexibility by using auxiliary information (e.g., name frequency, migration rates)
  - Utilize a probabilistic match score for more accurate linking, even with incomplete or imprecise data
  - Scalable and capable of handling millions of records, making it efficient in terms of speed and accuracy
- Limitations
  - Less effective with long strings (e.g., full names, long addresses) due to variations and typographical errors without advanced string-matching algorithms

Methodology

- General settings

Latent Matches	$M_{ij} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\lambda)$
Distance	$\gamma_k(i,j) \mid M_{ij} = m \overset{\text{indep.}}{\sim} \text{Discrete}\,(\pi_{km})$
Missing Indicator	$\delta_k(i,j) \perp \gamma_k(i,j) \mid M_{ij}$
Probability	$\xi_{ij} := \Pr\,(M_{ij} = 1 \mid \delta(i,j), \gamma(i,j))$

- Quite similar to Fellegi-Sunter[1]
- Capable pf dealing MAP missings

Methodology

- Probabilistic model, see getPosterior.R

$$\xi_{ij} = \frac{\lambda \prod_{k=1}^{K} \left(\prod_{\ell=0}^{L_k - 1} \pi_{k\ell\ell}^{1\{\gamma_k(i,j)=\ell\}} \right)^{1-\delta_k(i,j)}}{\sum_{m=0}^{1} \lambda^m (1-\lambda)^{1-m} \prod_{k=1}^{K} \left(\prod_{\ell=0}^{L_k - 1} \pi_{km\ell}^{1\{\gamma_\ell(i,j)=\ell\}} \right)^{1-\delta_k(i,j)}}$$

- Nice for evaluation and post-merge analysis

$$X_i^* = \sum_{j=1}^{N_{\mathcal{B}}} \xi_{ij} X_j / \sum_{j=1}^{N_{\mathcal{B}}} \xi_{ij}$$

Methodology

- Nice for calculation
  - Likelihood and boosting with EM steps[2]

$$\lambda = \frac{1}{N_A N_B} \sum_{i=1}^{N_A} \sum_{j=1}^{N_B} \xi_{ij}$$

$$\pi_{km\ell} = \frac{\sum_{i=1}^{N_A} \sum_{j=1}^{N_B} \mathbf{1}\left\{\gamma_k(i,j) = l)\right\} (1-\delta_k(i,j))\, \xi_{ij}^{m} \left(1-\xi_{ij}\right)^{1-m}}{\sum_{i=1}^{N_A} \sum_{j=1}^{N_B} (1-\delta_k(i,j))\, \xi_{ij}^{m} \left(1-\xi_{ij}\right)^{1-m}}$$

  - logemlink.R and emlinkMARmov.R

Package Implement

The required package for FastLink is called “fastLink”:

```r
install.packages("fastLink")
```

Install the most recent version of “fastLink” package (version 0.6):

```r
library(devtools)
install_github("kosukeimai/fastLink",dependencies=TRUE)
```

```r
## Load the package and data
library(fastLink)
```

Tutorial Link: https://github.com/kosukeimai/fastLink

Package Implement

```
matches.out <- fastLink(
  dfA = dfA, dfB = dfB,
  varnames = c("given_name", "surname", "address_1", "suburb")
  stringdist.match = c("given_name", "surname"),
  partial.match = c("given_name", "surname"),
  return.all = TRUE
)
```

The merged dataset can be accessed using the getMatches() function:

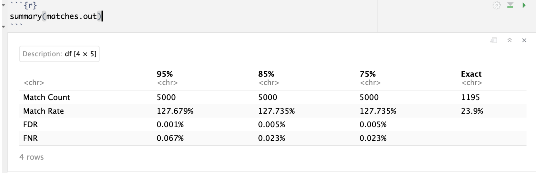
```
matched_dfs <- getMatches(
  dfA = dfA, dfB = dfB,
  fl.out = matches.out, threshold.match = 0.85
)
```

Package Implement

- Preprocessing Matches via Blocking: The blockData() function can block two datasets using one or more variables and various blocking techniques.
- Using Auxiliary Information to Inform fastLink: The algorithm could also incorporate auxiliary information on migration behavior to inform the matching of datasets over time.
- Aggregating Multiple Matches Together: The algorithm can also aggregate multiple matches into a single summary using the aggregateEM() function.
- Random Sampling with fastLink: The algorithm allows us to run the matching algorithm on a randomly selected smaller subset of data to be matched and then apply those estimates to the full sample of data.
- Finding Duplicates within a Dataset via fastLink: The algorithm uses the probabilistic match algorithm to identify duplicated entries.

Dataset

- Two sets of datasets explored
  - Products on Amazon & Google
  - fictious dataset from Freely Extensible Biomedical Record Linkage
- Empirical evidence of reduced effectiveness with long strings (e.g., product descriptions)
  - only 75 matches for Amazon (>1000 rows) & Google (>3000 rows) product datasets
- Avoid overly broad matching criteria / using too many variables
  - Inflate match rate (even over 100%), underestimate False Discovery Rate (FDR) and False Negative Rate (FNR)



~~~
```{r}
summary(matches.out)
```
~~~

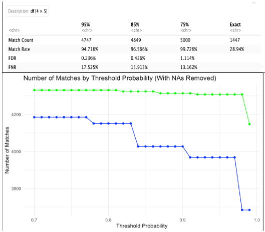
Description: df [4 × 5]

<chr>	95% <chr>	85% <chr>	75% <chr>	Exact <chr>
Match Count	5000	5000	5000	1195
Match Rate	127.679%	127.735%	127.735%	23.9%
FDR	0.001%	0.005%	0.005%	
FNR	0.067%	0.023%	0.023%	

4 rows

Dataset

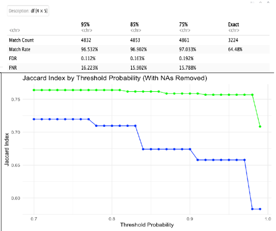
- choose more informative variable could reduce the use of matching variables
  - name + soc_sec_id vs name + address + suburb
- number of matches & jaccard index vs threshold probabilities
  - jaccard index = intersection size / union size



Description: df [4 × 5]

<chr>	95% <chr>	85% <chr>	75% <chr>	Exact <chr>
Match Count	4747	4849	5000	1447
Match Rate	94.716%	96.566%	99.726%	28.94%
FDR	0.236%	0.426%	1.114%	
FNR	17.525%	15.913%	13.162%	

Number of Matches by Threshold Probability (With NAs Removed)



Description: df [4 × 5]

<chr>	95% <chr>	85% <chr>	75% <chr>	Exact <chr>
Match Count	4832	4853	4861	3224
Match Rate	96.532%	96.902%	97.033%	64.48%
FDR	0.112%	0.163%	0.192%	
FNR	16.223%	15.902%	15.788%	

Jaccard Index by Threshold Probability (With NAs Removed)

References

[1] I. P. Fellegi and A. B. Sunter, “A theory for record linkage,” *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.

[2] W. E. Winkler, *Using the EM algorithm for weight computation in the fellegi-sunter model of record linkage*. US Bureau of the Census Washington, DC, 2000.