Module X: Distributied and Scalable Bayesian Graphical Entity Resolution

Rebecca C. Steorts

Reading

- Binette and Steorts (2020)
- Steorts, Hall, Fienberg (2016)
- Steorts (2015)
- Marchant et al. (2020)

Why is ER difficult?

Suppose that we have a total of N records in k databases.

1. We seek models that are much less than $O(N^k)$.
2. We seek models that are reliable, accurate, fit the data well, and account for the uncertainty of the model.
3. We seek models and algorithms to handle unbalanced data (containing duplications).

Existing ER methods

1. deterministic linking
2. probabilistic linking (Fellegi Sunter, random forests, deep learning)
3. Bayesian Fellegi Sunter

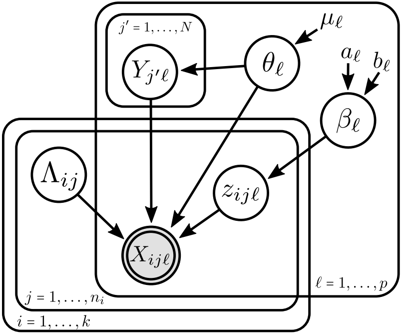
Limitations of Existing ER methods

- pairs of records are assessed independently
- awkward post-processing step (transitive closure)
- subjectivity in setting the decision threshold
- lack of uncertainty quantification
- require training data
- scalability achieved via deterministic dimension reduction of the data

[Fellegi and Sunter (1969), Ventura et al. (2014), Christen (2012), Dong and Shrivastava (2015), Belin and Rubin (1995), Gutman et al. (2013), McVeigh et al. (2020), Sadinle (2014+)].

Graphical Bayesian ER

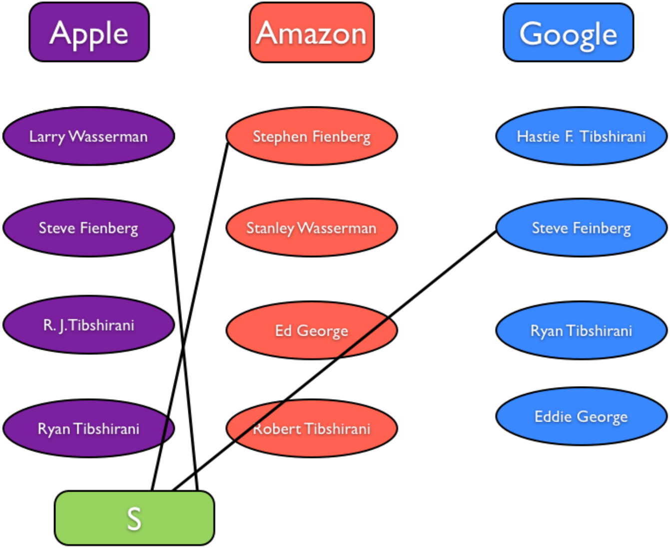
Builds off Copas and Hilton (2011), Tancredi and Liseo (2011).



[**RCS**, Hall, Fienberg (2014, 2016); **RCS** (2015), Zanella, et al. (2016), **RCS** et al. (2017), (2018), Tancredi et al. (2019), Betancourt et al. (2020)].

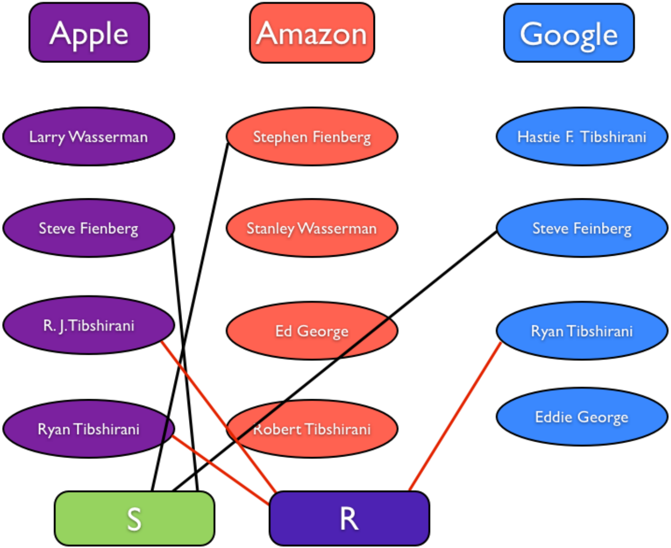
Review of Bayesian Graphical ER

\begin{figure}[htbp] \begin{center}



\end{figure}

Review of Bayesian Graphical ER



Our Goal

Scaling Bayesian ER methods to millions of records without sacrificing accuracy and crucially giving uncertainty of the ER task

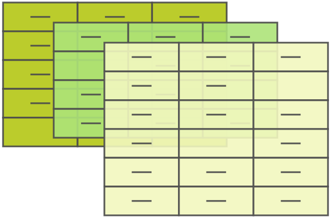
Our Solution

We propose a scalable joint (Bayesian) model for blocking and performing entity resolution, where the error from this joint task is exactly measured.

Problem setup

Key assumptions:

- multiple tables/sources
- duplicates within and across tables
- attributes are aligned
- attributes are discrete
- some missing values
- no ground truth (unsupervised)

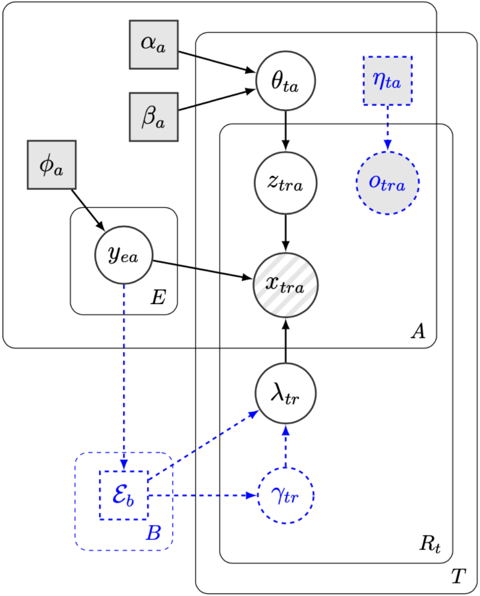


Output: approximate posterior distribution over the blocks and linkage structure

Contribution

1. Joint Bayesian model for blocking (latent entities) and ER.
2. Propose blocks (latent entities) that induce conditional independence between the latent entities.
3. Blocking function (responsible for partitioning the entities) groups similar entities together while achieving well-balanced partitions.
4. Application of partially-collapsed Gibbs sampling in the context of distributed computing.
5. Improving computational efficiency:
   a) Sub-quadratic algorithm for updating links based on indexing.
   b) Truncation of the attribute similarities.
   c) Perturbation sampling algorithm for updating the entity attributes, which relies on the Vose-Alias method.

dblink


α_a
β_a
ϕ_a
θ_{ta}
η_{ta}
z_{tra}
o_{tra}
y_{ea}
x_{tra}
E
A
λ_{tr}
$\mathcal{E}_b$
γ_{tr}
B
R_t
T

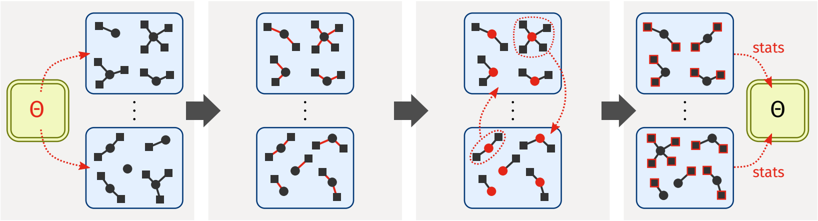
Posterior inference

Since the posterior for the linkage structure $p(\Lambda|X)$ is not tractable, we resort to *approximate inference*.

We propose an MCMC algorithm based on the *partially-collapsed Gibbs* sampler$\sim$(van Dyk and Park, 2008):

- regular Gibbs updates for the distortion probabilities θ_{ta}, distortion indicators z_{tra} and links λ_{tr}
- "marginalization" and "trimming" are applied to jointly update the entity attributes y_{ea} and the partition assignments for the linked records
- order of the updates is important (to preserve the stationary distribution)

distributed MCMC



Step 1

Update Θ on the master and broadcast to the workers.

Step 2

Update Λ on the workers. Records may only link to entities within their assigned partitions.

Step 3

Update Y and Γ on the workers. Move the entities and records to their newly-assigned partitions.

Step 4

Update Z, then calculate summary stats on the workers. Broadcast to the master.

Tricks for speeding up inference

1. linkage structure update $\mathcal{O}(\#\text{ records} \times \#\text{ entities})$
2. entity attribute update $\mathcal{O}(\#\text{ entities} \times \text{domain size})$

Solutions:

1. Indexing: Maintain indices from "entity attributes $\rightarrow$ entities" and "entities $\rightarrow$ linked records." This allows us to prune candidate links for a record
2. Thresholding similarity scores
3. Express the distribution for the entity attribute update as a two-component perturbation mixture model

Software

Two software packages:

1. dblink: Apache Spark
2. dblinkR: R wrapper for Spark package

dblinkR

Given the sensitivity of connections between R and Spark, we will perform this demo solely in R.