COMP90051 Statistical Machine Learning

Lecture 24: Bayesian Record Linkage

Semester 2, 2020

Guest Lecturer: Neil Marchant

This lecture

- Application of PGMs to perform *record linkage*
- Opportunity to hear about recent research
- We'll cover:
  - Brief background on record linkage
  - A solution based on Bayesian models (D-PGM)
  - How to perform inference
  - Research challenges

Record linkage

Identifying records that refer to the same entity

Motivation: integrating data from different sources

- Scenario: public health researchers want to investigate risk factors associated with COVID-19-related deaths
- Information not available from a single source
  - Risk factors: primary care (GP) records
  - COVID-19 deaths: health department
- Need to merge data, but it's non-trivial without a shared identifier (e.g. Medicare numbers)
- Problem known as record linkage

Name	…	Health conditions
Steven Butler	…	Diabetes, Heart disease
Alanna Thompson	…	
Antonio Ortiz	…	
Evelyn Zhang	…	
Abigail Williams	…	Hypertension

?

Name	DOD	Postcode
Phoebe Welch	03/03/20	3032
Vanessa Lowry	10/02/20	3130
Stephen Butler	05/07/20	3042
Mary Merritt	13/07/20	
Antonia Ortiz	29/08/20	3150

The record linkage (RL) problem

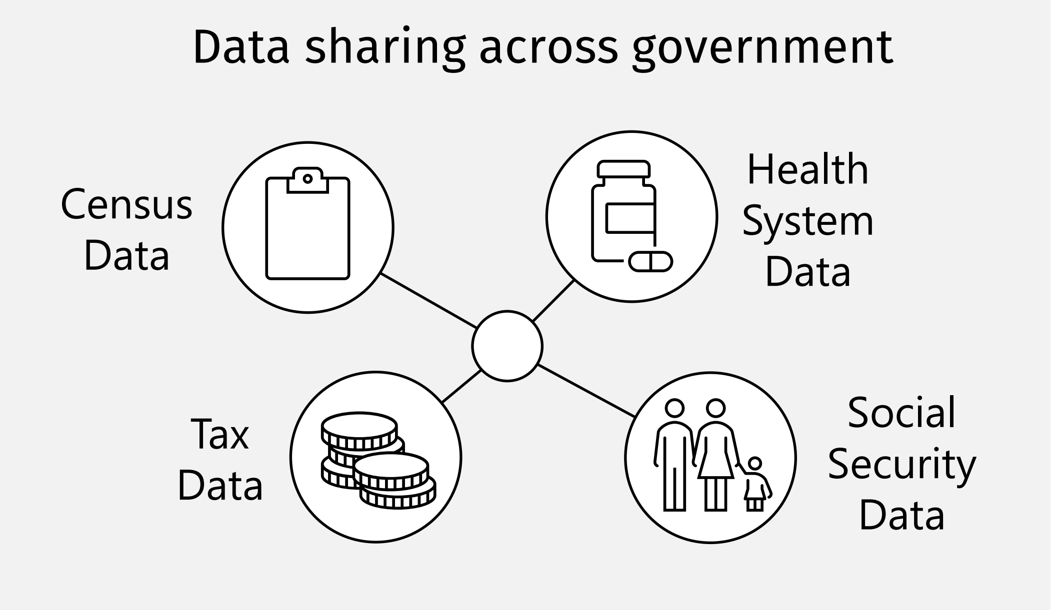
Definition
Consider a set of data sources providing a set of records $\mathcal{R}$. Let P be a (coreference) relation on $\mathcal{R}$ such that:

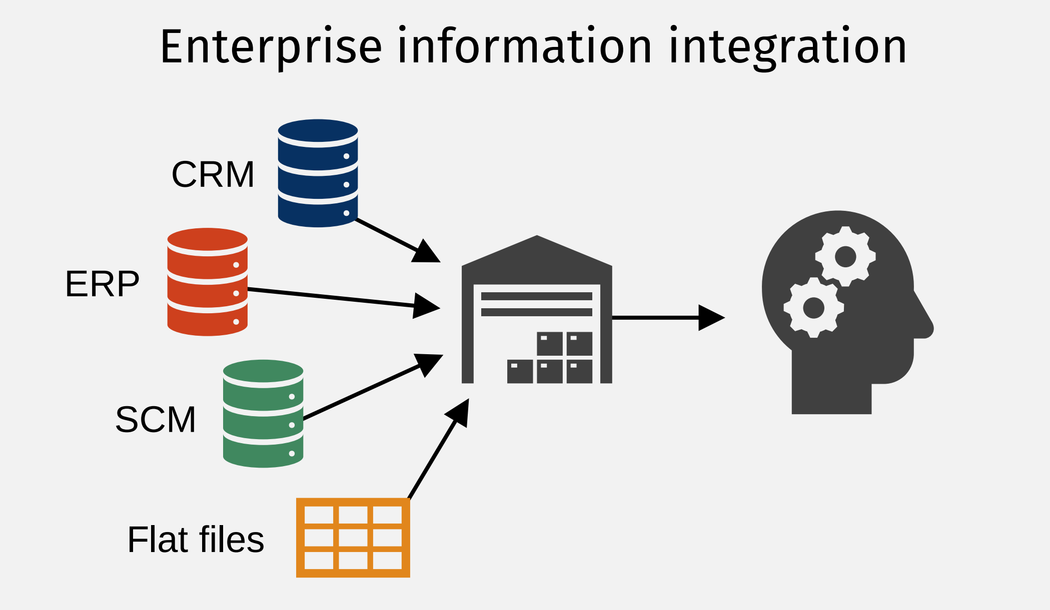
- $(r, r') \in P$ for any pair of records $r, r' \in \mathcal{R}$ that refer to the same entity,
- $(r, r') \notin P$ for any pair of records $r, r' \in \mathcal{R}$ that refer to distinct entities.

The record linkage (RL) problem is to approximate the true relation P by a predicted relation $\hat{P}$.

- Also known as entity resolution, data matching, deduplication, merge/purge
- Can formulate as a classification problem on pairs of records, although may get conflicting predictions
- Practical issue: ground truth labels are often unavailable

RL: a ubiquitous problem





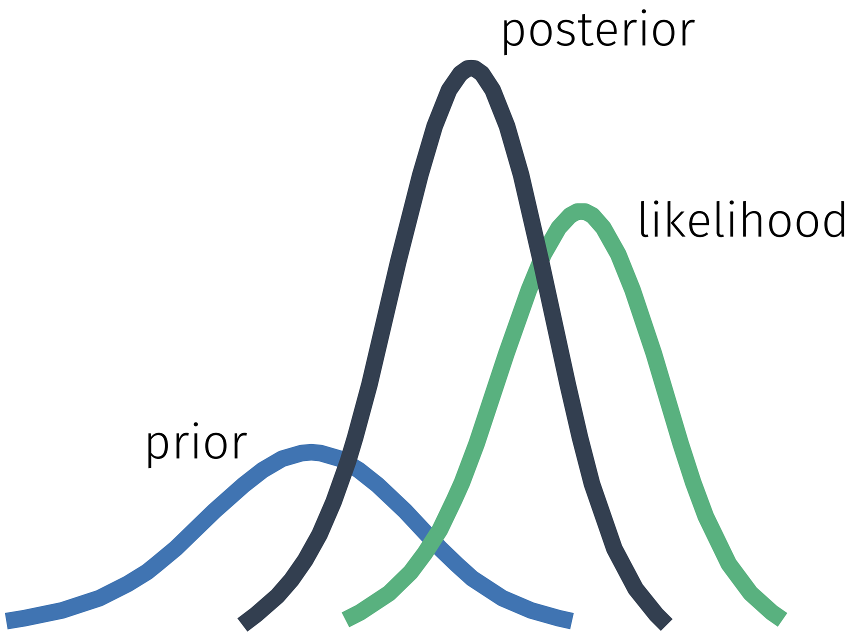
And many others: linking product listings across the web to build an e-commerce aggregator, linking accounts across social networks, linking records to produce credit ratings, building knowledge graphs using web sources, …

Bayesian record linkage

An effective class of methods for solving RL under uncertainty

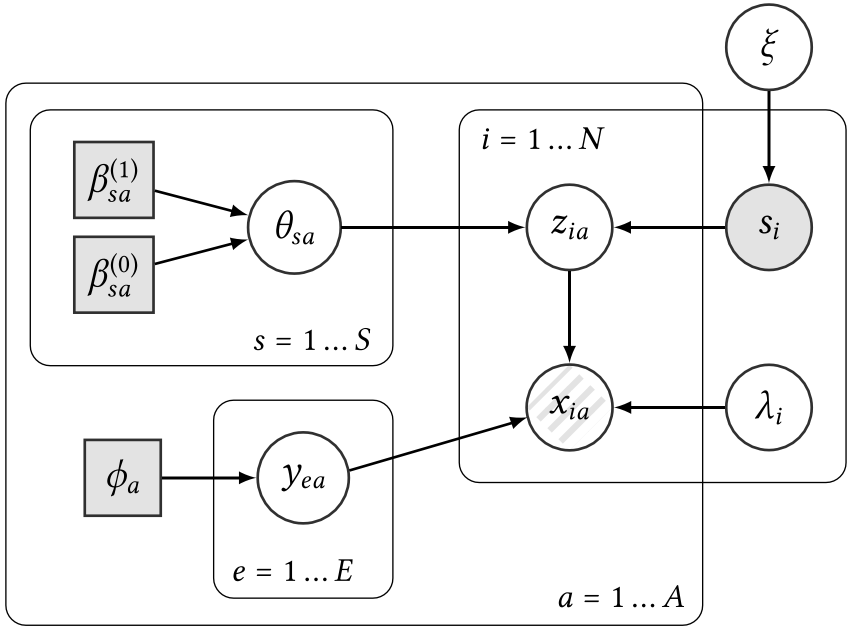
Why Bayesian models?

- They tend to be data-efficient—important since we often have no ground truth for RL
- Model encodes constraints and prior beliefs about the generative process
- Apply Bayes' rule to update beliefs about unknown parameters (i.e. coreference relation), conditional on observed data
- Distributions represent uncertainty in beliefs—can propagate to analyses on linked data



blink model for RL

- A Bayesian model for record linkage of structured data from multiple sources
- The model incorporates a population of latent entities with “true” attributes
- Records are generated from the entities by copying their attributes subject to distortion
- More sophisticated distortion model than previous methods—e.g. allows for typos
- Proposed by Steorts (2015)



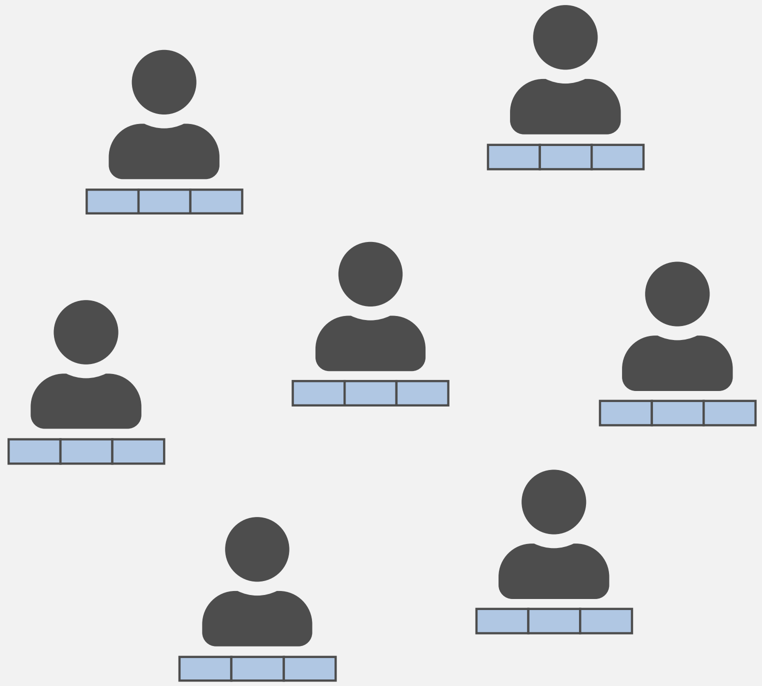
blink model for RL

Entity model

- Fixed population of entities indexed by $e \in \{1, \dots, E\}$
- Each entity e described by a tuple of true attributes $\mathbf{y}_e = (y_{e1}, \dots, y_{eA})$
- Value of attribute a for entity e is generated according to

$$y_{ea} \sim \mathrm{Categorical}(\boldsymbol{\phi}_a)$$

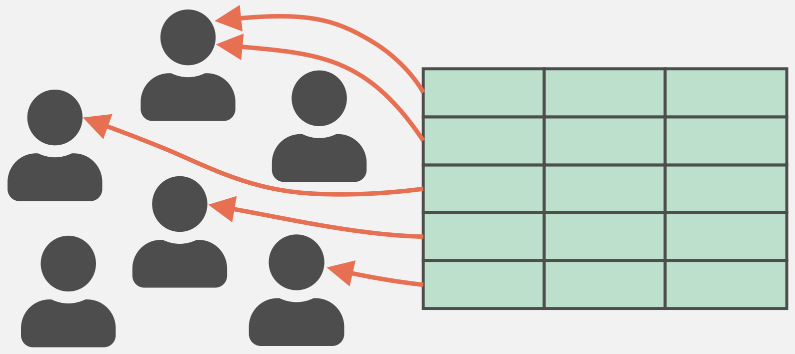
  where $\boldsymbol{\phi}_a$ is a distribution over attribute domain $\mathcal{V}_a$ (set empirically)



blink model for RL

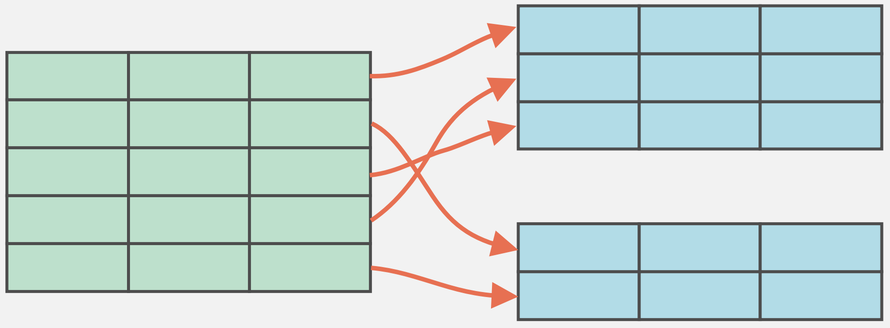
Linkage model

- Record i is generated by linking to an entity uniformly at random
$$\lambda_i \sim \text{DiscreteUniform}(1, \ldots, E)$$



Source model

- Record i is associated with source
$$s_i \sim \text{Categorical}(\boldsymbol{\xi})$$
where $\boldsymbol{\xi}$ is an unknown distribution over sources $s \in \{1, \ldots, S\}$



blink model for RL

Distortion model

- A distortion probability is associated with each attribute a and source s:

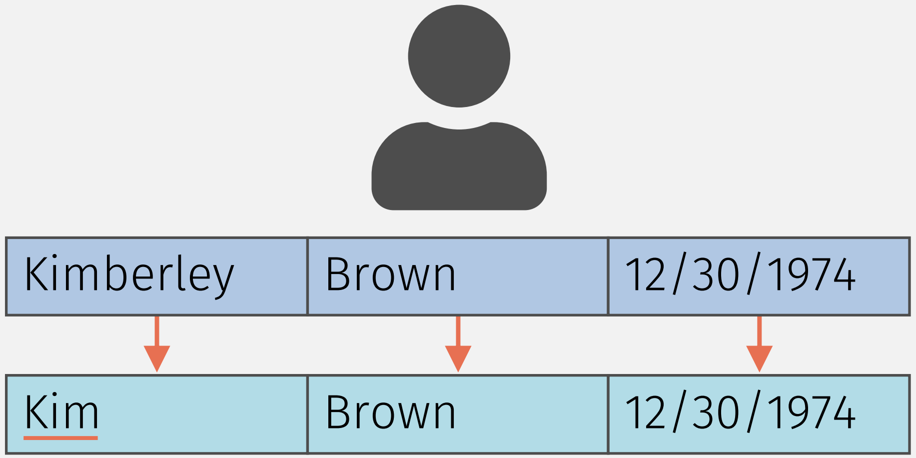
$$\theta_{sa} \sim \mathrm{Beta}(\alpha_a, \beta_a)$$

- The value of attribute a for record i follows a *hit-miss model*:

$$\begin{aligned} z_{ia} | \theta_{s_i a} &\sim \mathrm{Bernoulli}\left(\theta_{s_i a}\right) \\ x_{ia} | z_{ia}, y_{\lambda_i a} &\sim (1 - z_{ia})\delta\left(y_{\lambda_i a}\right) \\ &\quad + z_{ia}\, \mathrm{Discrete}(\boldsymbol{\psi}_a(y_{\lambda_i a})) \end{aligned}$$

Binary distortion indicator

Distortion distribution over domain of attribute



Joint distribution for blink

- Can write down the joint distribution over all variables:

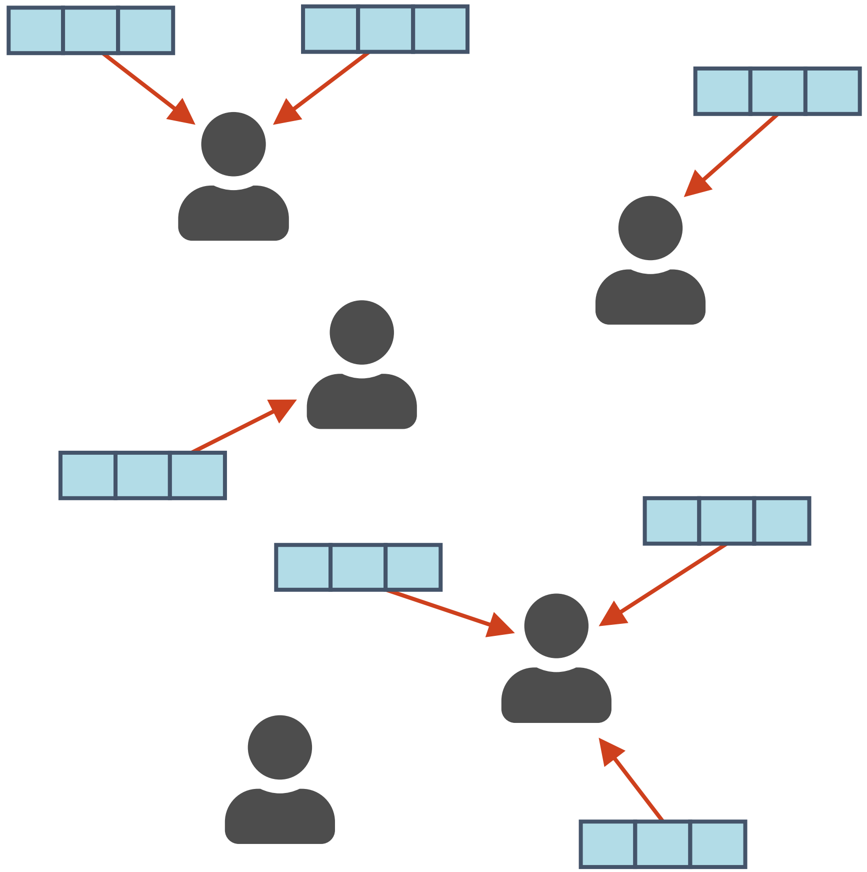
$$p(\mathbf{\Lambda}, \mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{\Theta}) = \prod_{e,a} p(y_{ea} | \boldsymbol{\phi}_a) \times \prod_i p(\lambda_i) \times \prod_{s,a} p(\theta_{sa} | \alpha_a, \beta_a)$$
$$\prod_{i,a} p\big(z_{ia} | \theta_{s_i a}\big) p(x_{ia} | z_{ia}, \lambda_i, y_{\lambda_i a}, \boldsymbol{\psi}_a)$$

- Conditionals specified on previous slides
- To do record linkage, we infer $\mathbf{\Lambda} = (\lambda_1, \ldots, \lambda_N)$ (the linkage structure) conditional on $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ (the records)
- Talk about inference next

Inference for blink

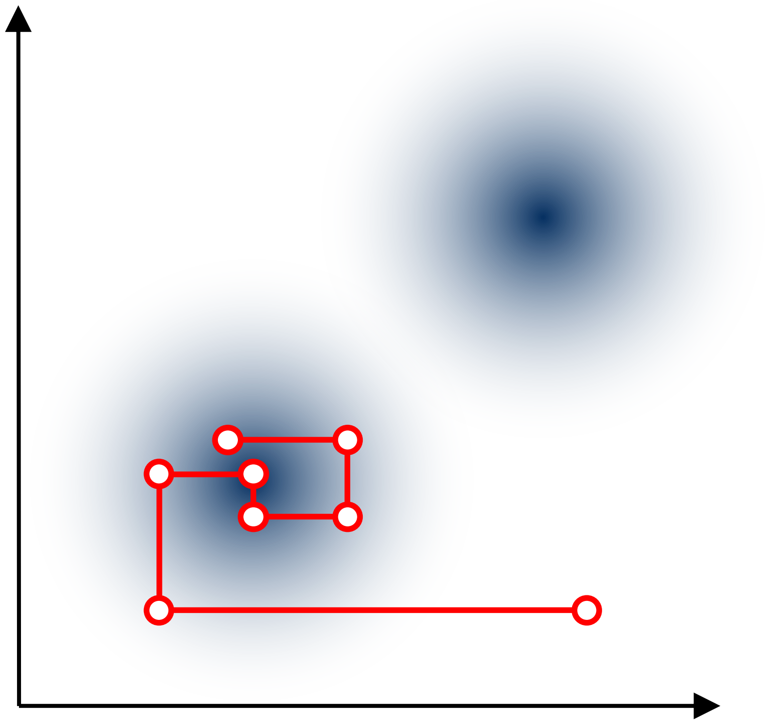
How to make predictions?

- Want to compute $p(\mathbf{\Lambda}|\mathbf{X}) = \frac{p(\mathbf{\Lambda},\mathbf{X})}{p(\mathbf{X})}$ for record linkage
- Although we can write down the joint $p(\mathbf{\Lambda}, \mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{\Theta})$, marginalising out latent variables is infeasible
- Must resort to approximate inference
- Standard approach is to approximate $p(\mathbf{\Lambda}|\mathbf{X})$ using samples obtained via Markov chain Monte Carlo (MCMC)
- Gibbs sampling is one of the simplest MCMC methods



Refresher: Gibbs sampling

- Method for obtaining samples from a (high-dimensional) joint distribution—in our case $p(\mathbf{\Lambda},\mathbf{Y},\mathbf{Z},\mathbf{\Theta}|\mathbf{X})$
- Only need to know the joint distribution up to a constant factor
- Sample one variable at a time, holding all others fixed
- Caveat: conditional distributions must be known and easy to sample from → they are for blink



Gibbs sampler for blink

Need to derive conditional distributions for each unobserved variable and ensure we can sample from them. Let's look at an example.

Conditional for λ_i

$$p(\lambda_i|\mathbf{\Lambda}_{-i}, \mathbf{Y}, \mathbf{X}, \mathbf{Z}, \mathbf{\Theta}) \propto p(\lambda_i) \prod_a p\left(x_{ia}\middle|z_{ia}, \lambda_i, y_{\lambda_i a}, \boldsymbol{\psi}_a\right)$$

$$\propto \prod_a \left\{(1 - z_{ia})\, \mathbb{I}\left(x_{ia} = y_{\lambda_i a}\right) + z_{ia}\, \boldsymbol{\psi}_a\left(x_{ia}\middle|y_{\lambda_i a}\right)\right\}$$

- A discrete distribution over the entities $1, \ldots, E$, although some entities may have zero weight if the entity attributes are a poor match for the record
- Notice: sampling naively takes $O(E)$ time—inefficient for large E

Gibbs sampler for blink

- Relatively straightforward to derive conditional distributions for the other variables θ_{sa}, y_{ea}, z_{ia} [exercise: try it yourself]
- Gibbs sampler is implemented in an R package released with the blink paper

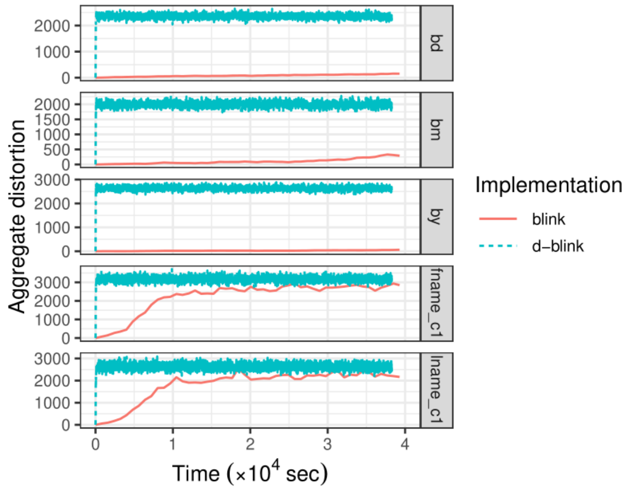
Research directions

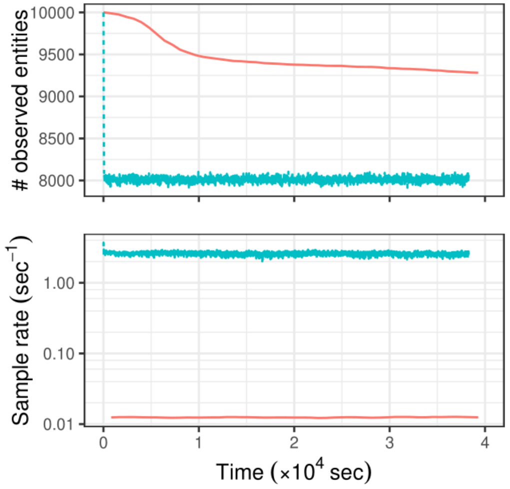
Improving MCMC efficiency

Gibbs sampling: Markov chain converges slowly and exhibits high autocorrelation

Research directions:

- Marginalising out latent variables can help (teal curve on right demonstrates improvement)
- Designing proposals that make more “global” updates under a Metropolis-Hastings framework—e.g. proposing to split/merge entities
- Bear in mind: parameter space is discrete → challenging for gradient-based methods





Scaling to large databases

A single Gibbs update for $\boldsymbol{\Lambda}$ (linkage structure) takes $O(N \cdot E)$ time. Since $E \approx N$, inference scales roughly quadratically in the number of records N.

Research directions:

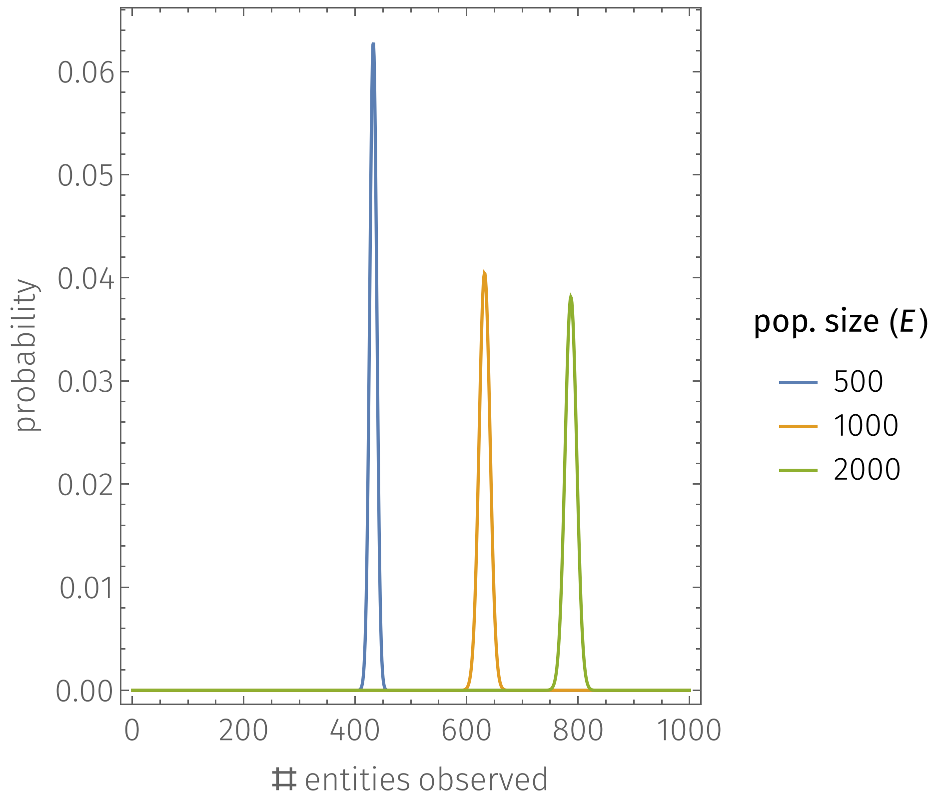
- Can speed up Gibbs update for blink using an inverted index
- Parallel/distributed MCMC
- More generally, can exploit the fact that many links are extremely unlikely, so it's wasteful to consider them
  - Blocking
  - Locality sensitive hashing (Indyk & Motwani, 1998)
  - Canopy clustering (McCallum et al., 2000)

Modelling improvements

Prior on $\boldsymbol{\Lambda}$ used in blink is too informative—have no control over spread (see right plot). Furthermore, several parameters are assumed known and are set empirically.

Research directions:

- Appropriate priors on $\boldsymbol{\Lambda}$—surprisingly challenging to ensure appropriate behaviour asymptotically
- Bayesian nonparametrics—scaling "number" of parameters based on data
- More sophisticated distortion models
- Fewer independence assumptions



Summary

- Introduced record linkage (RL) → an important task for integrating and cleaning data that can be solved using ML methods
- blink Bayesian model for RL
  - Suited to linking/deduplicating structured databases
  - Unsupervised
  - Relatively simple to implement → leverage concepts covered in this subject
- Inference using Gibbs sampling
- Active areas of research