# Multifile Record Linkage and Duplicate Detection Via a Structured Prior for Partitions

Serge Aleshin-Guendel
Joint work with Mauricio Sadinle
University of Washington, Department of Biostatistics

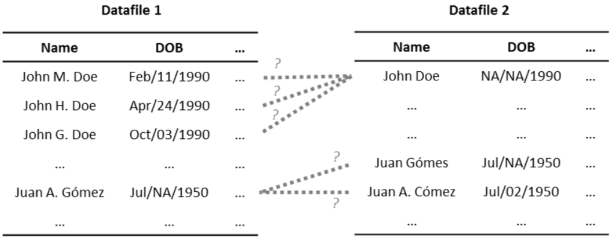
June 5, 2020

# What is this talk about?

- It's common to have data sources containing information on possibly overlapping sets of entities
- We'd like to merge these sources to harness all the available information for an analysis
- But how do you accomplish this merging when there are no unique identifiers for the records?

# Why "Record Linkage"?

- Common scenario: 2 data sources containing records on overlapping subsets of some population
- Due to knowledge of the data collection, we assume that there *are no* duplicates within either source
- But there are no unique identifiers for the records!
- How do we "link" records between sources? **Record Linkage**



# Why "Duplicate Detection"?

- Another common scenario: 1 data source
- Due to knowledge of the data collection, we assume that there *are* duplicates within the data source
- Again there are no unique identifiers for the records!
- How do we "detect" which of these records are duplicates? **Duplicate Detection**

**Datafile 2**

| Name | DOB | ... |
|---|---|---|
| John Doe | NA/NA/1990 | ... |
| ... | ... | ... |
| ... | ... | ... |
| Juan Gómes | Jul/NA/1950 | ... |
| Juan A. Cómez | Jul/02/1950 | ... |
| ... | ... | ... |

# Why "Multifile Record Linkage and Duplicate Detection"?

- Wording in the last two slides was very deliberate
- What if we have something *in between or beyond*?
- These scenarios all fall under the overarching problem of **Multifile Record Linkage and Duplicate Detection**

# Why “Via a Structured Prior for Partitions”?

- This is SDSS so there should be statistics somewhere
- As a statistical problem, we want to estimate a **partition** of the records *into clusters representing the same entity*

**Datafile 1**

| Name | DOB | ... |
|---|---|---|
| John M. Doe | Feb/11/1990 | ... |
| John H. Doe | Apr/24/1990 | ... |
| John G. Doe | Oct/03/1990 | ... |
| ... | ... | ... |
| Juan A. Gómez | Jul/NA/1950 | ... |
| ... | ... | ... |

**Datafile 2**

| Name | DOB | ... |
|---|---|---|
| John Doe | NA/NA/1990 | ... |
| ... | ... | ... |
| ... | ... | ... |
| Juan Gómes | Jul/NA/1950 | ... |
| Juan A. Cómez | Jul/02/1950 | ... |
| ... | ... | ... |

# Why “Via a Structured Prior for Partitions”?

- But how do you estimate a partition?
- If you’re Bayesian how do you construct priors on partitions?
- Further how do you construct priors on partitions that are *relevant to our setting*?
  **Via a Structured Prior for Partitions**

# Setup

- Have $r$ records in $K$ files $\boldsymbol{X}_1, \cdots, \boldsymbol{X}_K$
- Each record has $F$ fields of information
- Our *data* are these fields
- Our *parameter of interest* is a partition, $\mathcal{C}$, of the records
- As in most statistical models, want to model our *data* conditional on our *parameter of interest*

| First Name | Last Name | Age | Zip Code | Phone Number |
|---|---|---|---|---|
| Jennifer | Smith | 30 | 96024 | 301-867-5309 |

# Generative Processes

- We need a **prior for partitions**, and a **likelihood for fields**
- Will first focus on the prior for partitions first
- A useful starting point is to construct a hypothetical generative process for our data

# A Generative Process for Record Linkage

"True" records of
the latent entities

| Name | DOB |
|---|---|
| John Smith | 07/14/1987 |
| Jane Doe | 06/22/1992 |
| Robert Kim | 05/03/1979 |
| ... | |

# A Generative Process for Record Linkage

"True" records of the latent entities

| Name | DOB |
|---|---|
| John Smith | 07/14/1987 |
| Jane Doe | 06/22/1992 |
| Robert Kim | 05/03/1979 |
| ... | |

Data Collection Process 1

Observed records

| File 1 | |
|---|---|
| Name | DOB |
| John Smit | 07/14/1987 |
| Jon Smith | 07/14/1986 |
| John Smyth | 07/19/1987 |
| ... | |

# A Generative Process for Record Linkage

"True" records of
the latent entities

| Name | DOB |
|---|---|
| John Smith | 07/14/1987 |
| Jane Doe | 06/22/1992 |
| Robert Kim | 05/03/1979 |
| ... | |

Data Collection Process 2

Observed records

File 1

| Name | DOB |
|---|---|
| John Smit | 07/14/1987 |
| Jon Smith | 07/14/1986 |
| John Smyth | 07/19/1987 |
| ... | |

File 2

| Name | DOB |
|---|---|
| John | NA |
| Jan | NA |
| ... | |

# A Generative Process for Record Linkage

"True" records of the latent entities

| Name | DOB |
|---|---|
| John Smith | 07/14/1987 |
| Jane Doe | 06/22/1992 |
| Robert Kim | 05/03/1979 |
| ... | |

Data Collection Process 3

Observed records

File 1

| Name | DOB |
|---|---|
| John Smit | 07/14/1987 |
| Jon Smith | 07/14/1986 |
| John Smyth | 07/19/1987 |
| ... | |

File 2

| Name | DOB |
|---|---|
| John | NA |
| Jan | NA |
| ... | |

File 3

| Name | DOB |
|---|---|
| Robert Kim | 05/03/1974 |
| Bob Kim | 05/03/1979 |
| ... | |

# A Generative Process for Record Linkage

"True" records of
the latent entities

| Name | DOB |
|---|---|
| John Smith | 07/14/1987 |
| Jane Doe | 06/22/1992 |
| Robert Kim | 05/03/1979 |
| ... | |

Observed records

| File 1 | |
|---|---|
| Name | DOB |
| John Smit | 07/14/1987 |
| Jon Smith | 07/14/1986 |
| John Smyth | 07/19/1987 |
| ... | |

| File 2 | |
|---|---|
| Name | DOB |
| John | NA |
| Jan | NA |
| ... | |

| File 3 | |
|---|---|
| Name | DOB |
| Robert Kim | 05/03/1974 |
| Bob Kim | 05/03/1979 |
| ... | |

# From a Generative Process to a Prior for Partitions

- By parameterizing each step of the generative process we can form a prior for partitions!

# Step 1: Number of Latent Entities

- First place a prior on the number of latent entities, $n$
- Lots of distributions on $\{1, 2, 3, \cdots\}$ that can be used to incorporate prior information

$$P(\mathcal{C}) = P(n) \times \cdots$$

# Step 1: Number of Latent Entities

"True" records of the latent entities

| Name | DOB |
|---|---|
| John Smith | 07/14/1987 |
| Jane Doe | 06/22/1992 |
| Robert Kim | 05/03/1979 |

There are $n$=3 latent entities represented in the observed records

Observed records

| File 1 | |
|---|---|
| Name | DOB |
| John Smit | 07/14/1987 |
| Jon Smith | 07/14/1986 |
| John Smyth | 07/19/1987 |

| File 2 | |
|---|---|
| Name | DOB |
| John | NA |
| Jan | NA |

| File 3 | |
|---|---|
| Name | DOB |
| Robert Kim | 05/03/1974 |
| Bob Kim | 05/03/1979 |

## Step 2: Overlap

- Conditional on $n$, we place a prior on the number of entities "captured" by each subset of files $\{1, \cdots, K\}$
- E.g. for $K = 3$ files, the counts can be represented as

| | Not In File 2 | | In File 2 | |
|---|---|---|---|---|
| | Not In File 1 | In File 1 | Not In File 1 | In File 1 |
| Not In File 3 | - | $n_{100}$ | $n_{010}$ | $n_{110}$ |
| In File 3 | $n_{001}$ | $n_{101}$ | $n_{011}$ | $n_{111}$ |

- Refer to this collection of counts as

$$\boldsymbol{n} = (n_{100}, n_{010}, n_{001}, n_{110}, n_{101}, n_{011}, n_{111})$$

- We want to place a prior on $\boldsymbol{n} \mid n$
- Natural choices are multinomial or Dirichlet-multinomial

$$P(\mathcal{C}) = P(n) \times P(\boldsymbol{n} \mid n) \times \cdots$$

# Step 2: Overlap

| | Not In File 2 | | In File 2 | |
|---|---|---|---|---|
| | Not In File 1 | In File 1 | Not In File 1 | In File 1 |
| Not In File 3 | – | $n_{100} = 0$ | $n_{010} = 1$ | $n_{110} = 1$ |
| In File 3 | $n_{001} = 1$ | $n_{101} = 0$ | $n_{011} = 0$ | $n_{111} = 0$ |

Observed records

| File 1 | |
|---|---|
| Name | DOB |
| John Smit | 07/14/1987 |
| Jon Smith | 07/14/1986 |
| John Smyth | 07/19/1987 |

| File 2 | |
|---|---|
| Name | DOB |
| John | NA |
| Jan | NA |

| File 3 | |
|---|---|
| Name | DOB |
| Robert Kim | 05/03/1974 |
| Bob Kim | 05/03/1979 |

John Smith is in File 1 and File 2

Jane Doe is in File 2

Robert Kim is in File 3

# Step 3: Number of Duplicates

- Conditional on the number of entities in each file, place a prior on the number of duplicates for each entity *in each file*
- Call this collection of duplicate counts $\boldsymbol{d}$
- Lots of distributions on $\{1, 2, 3, \cdots\}$ that can be used to incorporate prior information

$$P(\mathcal{C}) = P(n) \times P(\boldsymbol{n}|n) \times P(\boldsymbol{d}|\boldsymbol{n}) \times \cdots$$

# Step 3: Number of Duplicates

"True" records of
the latent entities

| Name | DOB |
|---|---|
| John Smith | 07/14/1987 |
| Jane Doe | 06/22/1992 |
| Robert Kim | 05/03/1979 |

John Smith has 3 duplicates in File 1

John Smith has 1 duplicate in File 2

Jane Doe has 1 duplicate in File 2

Robert Kim has 2 duplicates in File 3

Observed records

| File 1 | |
|---|---|
| Name | DOB |
| John Smit | 07/14/1987 |
| Jon Smith | 07/14/1986 |
| John Smyth | 07/19/1987 |

| File 2 | |
|---|---|
| Name | DOB |
| John | NA |
| Jan | NA |

| File 3 | |
|---|---|
| Name | DOB |
| Robert Kim | 05/03/1974 |
| Bob Kim | 05/03/1979 |

# Step 4: Putting it All Together

- So far I've just been putting priors on summaries of the partition
- E.g. we know there is an entity that's in File 1 and File 2, but we haven't specified which entity it is!
- Need to count how many partitions give rise to our summaries!
  - Simple counting argument

$$\boxed{P(\mathcal{C}) = P(n) \times P(\boldsymbol{n}|n) \times P(\boldsymbol{d}|\boldsymbol{n}) \times P(\mathcal{C}|\boldsymbol{n}, \boldsymbol{d})}$$

# Sidenote: $K$-partite Matchings

- For a given file, can enforce an assumption of no duplicates
  - Just need to make the prior for the number of duplicates a point mass at 1!
- If we make this restriction for all $K$ files, we wind up with a prior on $K$-partite matchings!
- Seems to be novel (Besides the bipartite case)

# Inspirations

Inspired by previous work in record linkage and duplicate detection

- Two-File Record Linkage: Priors on bipartite matchings [Fortini et al. (2001, 2002), Larsen (2005), Sadinle (2017)]
- Single-File Duplicate Detection: Kolchin partition priors [Zanella et al. (2016)]

# Comparison-Based Modeling of Fields

- Modeling fields directly is hard! (How do you model names?)
- Instead compare fields for each pair of records, model that
- Idea is that similar records are probably matches

| Record | First Name | Last Name | Age | $\cdots$ |
|---|---|---|---|---|
| $i$ | Benedict | Cumberbatch | 40 | $\cdots$ |
| $j$ | Benedict | Cucumberbatch | 39 | $\cdots$ |

- And dissimilar records are probably not matches

| Record | First Name | Last Name | Age | $\cdots$ |
|---|---|---|---|---|
| $i$ | Benedict | Cumberbatch | 40 | $\cdots$ |
| $j$ | Martin | Freeman | 45 | $\cdots$ |

# Comparison Data

- For each pair of records $i, j$, generate a vector containing comparisons for each field $\boldsymbol{\gamma}_{ij} = (\gamma_{ij}^1, \cdots, \gamma_{ij}^F)$
- Examples:
  - Strings (names, telephone numbers, etc.) can use Levenshtein distance (also known as the edit distance)
  - Categorical data can use binary comparison
  - Numeric data can use absolute distance
- For each field $f$ being compared, discretize the comparison $\gamma_{ij}^f$ into $L_f$ categories
  - Rely on generic models for categorical data

# Comparison Data Model

- Let record $i$ be from file $\boldsymbol{X}_k$ and record $j$ be from file $\boldsymbol{X}_{k'}$
- Let $\mathcal{C}(i)$ represent the cluster in $\mathcal{C}$ that record $i$ belongs to

$$\begin{aligned}
\gamma_{ij}^f|\mathcal{C}(i) = \mathcal{C}(j) &\overset{iid}{\sim} \text{Multinomial}(1, \boldsymbol{m}_{kk'}^f), \\
\gamma_{ij}^f|\mathcal{C}(i) \neq \mathcal{C}(j) &\overset{iid}{\sim} \text{Multinomial}(1, \boldsymbol{u}_{kk'}^f), \\
\mathcal{C} &\sim \text{Prior on Partitions}
\end{aligned}$$

- Use flat Dirichlet priors on $\boldsymbol{m}_{kk'}^f$, $\boldsymbol{u}_{kk'}^f$
- Different likelihood for each pair of files!

# Posterior Computation

- Gibbs sampler

# Point Estimates

- Combine the posterior $P(\mathcal{C} \mid \gamma)$ with an appropriate loss function $L(\hat{\mathcal{C}}, \mathcal{C})$
- Bayes estimate is partition $\hat{\mathcal{C}}$ that minimizes $E[L(\hat{\mathcal{C}}, \mathcal{C}) \mid \gamma] = \sum_{\mathcal{C}} L(\hat{\mathcal{C}}, \mathcal{C}) P(\mathcal{C} \mid \gamma)$
- We'll specify $L$ that allows uncertain portions of the partition to be left unresolved (**abstain option**)
- Unresolved portions can get resolved in clerical review
- Use MCMC samples to approximate posterior loss

# Loss Function

- We will specify the loss additiviely $L(\hat{\mathcal{C}}, \mathcal{C}) = \sum_{i=1}^{r} L_i(\hat{\mathcal{C}}, \mathcal{C})$
- Let $\Delta_{ij} = I(\mathcal{C}(i) = \mathcal{C}(j))$, and likewise $\hat{\Delta}_{ij} = I(\hat{\mathcal{C}}(i) = \hat{\mathcal{C}}(j))$

$$L_i(\hat{\mathcal{C}}, \mathcal{C}) = \begin{cases} \lambda_A, & \text{if } \hat{\mathcal{C}}(i) = A, \\ & \\ & \\ & \end{cases}$$

- Loss $\lambda_A$ when we abstain from making a decision for record $i$
- No abstain option when $\lambda_A = \infty$

# Loss Function

- We will specify the loss additiviely $L(\hat{\mathcal{C}}, \mathcal{C}) = \sum_{i=1}^{r} L_i(\hat{\mathcal{C}}, \mathcal{C})$
- Let $\Delta_{ij} = I(\mathcal{C}(i) = \mathcal{C}(j))$, and likewise $\hat{\Delta}_{ij} = I(\hat{\mathcal{C}}(i) = \hat{\mathcal{C}}(j))$

$$L_i(\hat{\mathcal{C}}, \mathcal{C}) = \begin{cases} \lambda_A, & \text{if } \hat{\mathcal{C}}(i) = A, \\ 0, & \text{if } \Delta_{ij} = \hat{\Delta}_{ij} \text{ for all } j \text{ where } \hat{\mathcal{C}}(j) \neq A, \\ & \\ & \end{cases}$$

- Loss 0 when we get record $i$'s cluster correct

# Loss Function

- We will specify the loss additiviely $L(\hat{\mathcal{C}}, \mathcal{C}) = \sum_{i=1}^{r} L_i(\hat{\mathcal{C}}, \mathcal{C})$
- Let $\Delta_{ij} = I(\mathcal{C}(i) = \mathcal{C}(j))$, and likewise $\hat{\Delta}_{ij} = I(\hat{\mathcal{C}}(i) = \hat{\mathcal{C}}(j))$

$$L_i(\hat{\mathcal{C}}, \mathcal{C}) = \begin{cases} \lambda_A, & \text{if } \hat{\mathcal{C}}(i) = A, \\ 0, & \text{if } \Delta_{ij} = \hat{\Delta}_{ij} \text{ for all } j \text{ where } \hat{\mathcal{C}}(j) \neq A, \\ \lambda_{\mathrm{FNM}}, & \text{if } \sum_{j \neq i} \hat{\Delta}_{ij} = 0, \ \ \sum_{j \neq i} \Delta_{ij} > 0, \\ \\ \\ \end{cases}$$

- Loss $\lambda_{\mathrm{FNM}}$ when we have a false non-match
- Deciding that record $i$ does not match any other record when in fact it does

# Loss Function

- We will specify the loss additiviely $L(\hat{\mathcal{C}}, \mathcal{C}) = \sum_{i=1}^{r} L_i(\hat{\mathcal{C}}, \mathcal{C})$
- Let $\Delta_{ij} = I(\mathcal{C}(i) = \mathcal{C}(j))$, and likewise $\hat{\Delta}_{ij} = I(\hat{\mathcal{C}}(i) = \hat{\mathcal{C}}(j))$

$$L_i(\hat{\mathcal{C}}, \mathcal{C}) = \begin{cases} \lambda_A, & \text{if } \hat{\mathcal{C}}(i) = A, \\ 0, & \text{if } \Delta_{ij} = \hat{\Delta}_{ij} \text{ for all } j \text{ where } \hat{\mathcal{C}}(j) \neq A, \\ \lambda_{\mathrm{FNM}}, & \text{if } \sum_{j \neq i} \hat{\Delta}_{ij} = 0, \quad \sum_{j \neq i} \Delta_{ij} > 0, \\ \lambda_{\mathrm{FM1}}, & \text{if } \sum_{j \neq i} \hat{\Delta}_{ij} > 0, \quad \sum_{j \neq i} \Delta_{ij} = 0, \end{cases}$$

- Loss $\lambda_{\mathrm{FM1}}$ when we have a type 1 false match
- Deciding that record $i$ matches other records when it doesn't actually match any other record

# Loss Function

- We will specify the loss additiviely $L(\hat{\mathcal{C}}, \mathcal{C}) = \sum_{i=1}^{r} L_i(\hat{\mathcal{C}}, \mathcal{C})$
- Let $\Delta_{ij} = I(\mathcal{C}(i) = \mathcal{C}(j))$, and likewise $\hat{\Delta}_{ij} = I(\hat{\mathcal{C}}(i) = \hat{\mathcal{C}}(j))$

$$L_i(\hat{\mathcal{C}}, \mathcal{C}) = \begin{cases} \lambda_A, & \text{if } \hat{\mathcal{C}}(i) = A, \\ 0, & \text{if } \Delta_{ij} = \hat{\Delta}_{ij} \text{ for all } j \text{ where } \hat{\mathcal{C}}(j) \neq A, \\ \lambda_{\text{FNM}}, & \text{if } \sum_{j \neq i} \hat{\Delta}_{ij} = 0, \quad \sum_{j \neq i} \Delta_{ij} > 0, \\ \lambda_{\text{FM1}}, & \text{if } \sum_{j \neq i} \hat{\Delta}_{ij} > 0, \quad \sum_{j \neq i} \Delta_{ij} = 0, \\ \lambda_{\text{FM2}}, & \text{if } \sum_{j \neq i} \hat{\Delta}_{ij} > 0, \quad \sum_{j \neq i} (1 - \hat{\Delta}_{ij}) \Delta_{ij} > 0. \end{cases}$$

- Loss $\lambda_{\text{FM2}}$ when we have a type 2 false match
- Deciding that record $i$ is matched to other records but it does not match all of the records it should be matching

# Approximating the Bayes Estimate

- Minimizing $E[L(\hat{\mathcal{C}}, \mathcal{C}) \mid \gamma] = \sum_{\mathcal{C}} L(\hat{\mathcal{C}}, \mathcal{C}) P(\mathcal{C} \mid \gamma)$ exactly is computationally intractable
  - The number of partitions of $r$ records gets very large very fast
- In practice large number of record pairs will have $\approx 0$ posterior probability of matching
- Break records up into connected components with posterior probability of matching $> \delta$
  - These connected components will hopefully have $\ll r$ records
- Minimize loss over MCMC samples within each connected component

# Simulations

- Our approach worked well in simulations
- Omitted for time, additional slides in appendix

# Application: Homicides in Colombia

- Data provided by the Conflict Analysis Resource Center (CERAC)
- 3 record systems containing information on homicides from 2004 in the Quindio province of Colombia
  - Departamento Administrativo Nacional de Estadistica, **DANE** (323 records)
  - Policia Nacional de Colombia, **PN** (157 records)
  - Instituto Nacional de Medicina Legal y Ciencias Forenses, **ML** (289 records)
- All 3 systems are believed to be free of duplicates
- Records previously linked by hand, gives us a ground truth

# Application: Homicides in Colombia

- Fields available for all 3 systems:
  - Municipality and date of the homicide
  - Whether the location of the homicide was urban or rural
  - Age, sex, and marital status of the victim
- Additionally educational status of the victim is available in DANE and ML

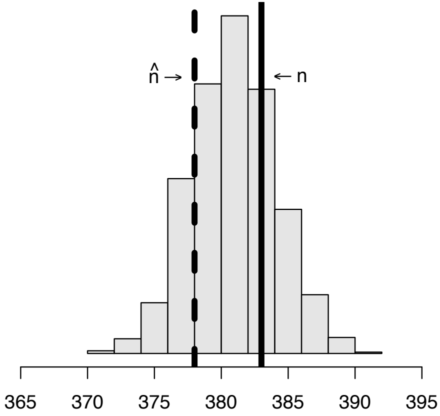
# Application: Results

- Full Bayes estimate (not using abstain option):
  - Precision of 93%
    - How many of the links we made were correct?
  - Recall of 96%
    - How many of the true links did we get correct?

- Partial Bayes estimate (using abstain option):
  - Precision of 95%
    - How many of the links we made were correct?
  - Abstention rate of 10%
    - For how many of the records did we abstain?

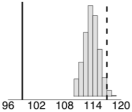
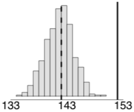
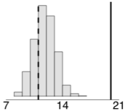
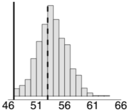
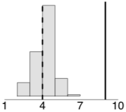
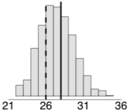
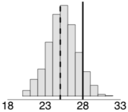
# Application: Results

- True number of entities was $n = 383$
  - 95 % credible interval of $[376, 388]$
  - Estimate (based on full Bayes estimate) of $\hat{n} = 378$



# Application: Results

- Dashed lines are estimates (based on full Bayes estimate)
- Solid lines are ground truth

| | In PN | | Out PN | |
|---|---|---|---|---|
| DANE | In ML | Out ML | In ML | Out ML |
| In | $n_{111}$  | $n_{101}$  | $n_{110}$  | $n_{100}$  |
| Out | $n_{011}$  | $n_{001}$  | $n_{010}$  | – |

# Conclusions

- It always helps to think about data generating processes!
- Novel prior on partitions (and $K$-partite matchings)
- Loss function with abstain option allows uncertain portions of the partition to be left unresolved

# That's All!

- Questions?
- Email: `aleshing@uw.edu`
- Paper and accompanying `R` package `multilink` coming soon
- Research was supported by NSF grant SES-1852841

# Sampling the Partition

Suppose we have samples of the partition $\mathcal{C}$ and parameters of the likelihood $\Phi = \{\boldsymbol{m}^f_{kk'},\ \boldsymbol{u}^f_{kk'}\}$, and we'd like to resample record $j$'s cluster assignment, where $j$ is in file $\boldsymbol{X}_k$. Let $\mathcal{C}_{-j}$ denote the partition with record $j$ removed. Then if $c \in \mathcal{C}_{-j}$ or $c = \emptyset$ (i.e. we're creating a new cluster):

$$p(\text{record } j \text{ gets assigned to } c \mid \mathcal{C}_{-j}, \Phi) \propto$$
$$\begin{cases} p_k(1) \times \left[\dfrac{(n(\mathcal{C}_{-j})+1)(n_{h(k)}(\mathcal{C}_{-j}) + \alpha_{h(k)})}{(n(\mathcal{C}_{-j}) + \boldsymbol{\alpha}_0)}\right] \times \left[\dfrac{p(n(\mathcal{C}_{-j})+1)}{p(n(\mathcal{C}_{-j}))}\right], & \text{if } |c| = 0 \\ \left[\prod_{i\in c} \mathcal{L}_{ij}\right] \times p_k(1) \times \left[\dfrac{n_{h_{c,j}}(\mathcal{C}_{-j}) + \alpha_{h_{c,j}}}{n_{h_{c,-j}}(\mathcal{C}_{-j}) + \alpha_{h_{c,-j}} - 1}\right], & \text{if } |c^k| = 0, |c| > 0 \\ \left[\prod_{i\in c} \mathcal{L}_{ij}\right] \times \left[(|c^k|+1)\dfrac{p_k(|c^k|+1)}{p_k(|c^k|)}\right], & \text{if } |c^k| > 0 \end{cases}$$

# Sampling the Partition

If $|c| = 0$, we're creating a new cluster,

$$p(\text{record } j \text{ gets assigned to } c \mid \mathcal{C}_{-j}, \Phi) \propto$$

$$p_k(1) \times \left[ \frac{(n(\mathcal{C}_{-j}) + 1)(n_{h(k)}(\mathcal{C}_{-j}) + \alpha_{h(k)})}{(n(\mathcal{C}_{-j}) + \boldsymbol{\alpha}_0)} \right] \times \left[ \frac{p(n(\mathcal{C}_{-j}) + 1)}{p(n(\mathcal{C}_{-j}))} \right]$$

- $p_k(1)$: prior prob. of having 1 duplicate for a cluster in file $\boldsymbol{X}_k$
- $n(\mathcal{C}_{-j})$: number of clusters in $\mathcal{C}_{-j}$
- $n_{h(k)}(\mathcal{C}_{-j})$: number of clusters in $\mathcal{C}_{-j}$ only containing records from $\boldsymbol{X}_k$
- $\alpha_{\dots}$: prior hyperparameters for contingency table of overlap

# Sampling the Partition

If $c \neq \emptyset$ but doesn't contain other records from file $\boldsymbol{X}_k$,

$$p(\text{record } j \text{ gets assigned to } c \mid \mathcal{C}_{-j}, \Phi) \propto$$

$$\left[\prod_{i \in c} \mathcal{L}_{ij}\right] \times p_k(1) \times \left[\frac{n_{h_{c,j}}(\mathcal{C}_{-j}) + \alpha_{h_{c,j}}}{n_{h_{c,-j}}(\mathcal{C}_{-j}) + \alpha_{h_{c,-j}} - 1}\right]$$

- $\mathcal{L}_{ij}$: the likelihood contribution for the comparison between record $i$ and record $j$
- $n_{h_{c,j}}(\mathcal{C}_{-j})$: number of clusters with same overlap as $c \cup \{j\}$ (i.e. the cluster $c$ if you add $j$ to it)
- $n_{h_{c,-j}}(\mathcal{C}_{-j})$: number of clusters with same overlap as $c$ (i.e. the cluster $c$ if you don't add $j$ to it)

# Sampling the Partition

If $c$ contains other records from file $\boldsymbol{X}_k$,

$$p(\text{record } j \text{ gets assigned to } c \mid \mathcal{C}_{-j}, \Phi) \propto$$

$$\left[\prod_{i \in c} \mathcal{L}_{ij}\right] \times \left[(|c^k| + 1)\frac{p_k(|c^k| + 1)}{p_k(|c^k|)}\right]$$

- $c^k$: the number of records in $c$ from file $\boldsymbol{X}_k$

# Simulations

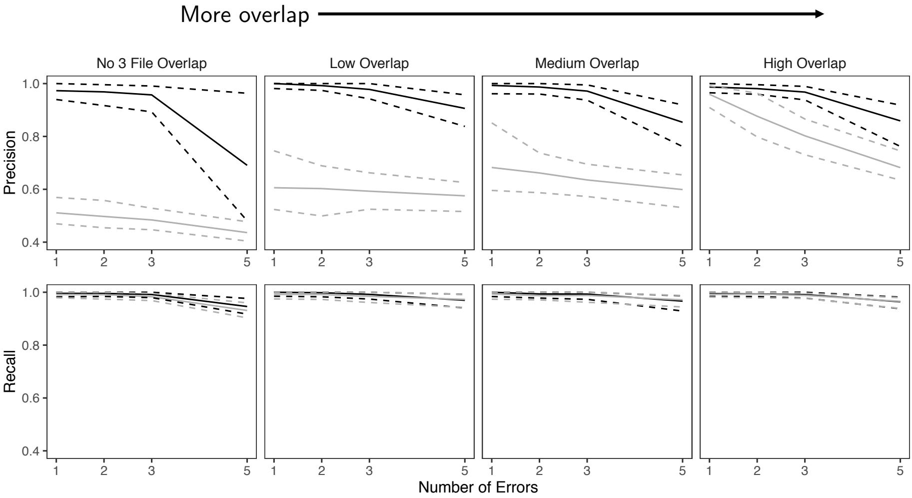
- 3 files, 500 latent entities
- Varying scenarios of measurement error, overlap, and duplication
- 100 simulated data sets for each scenario
  - Partitions generated roughly according to our prior
  - Actual records generated using code from group at ANU[1]

[1] https://dmm.anu.edu.au/geco/index.php

# Simulation 1: No Duplicates

- Vary amount of measurement error, overlap between files
- No duplicates, target is $K$-partite matching
- Comparisons between our comparison based model with
  - Our proposed prior on $K$-partite matchings
  - Uniform prior on $K$-partite matchings
- Full Bayes estimates (not using abstain option)

# Simulation 1: No Duplicates

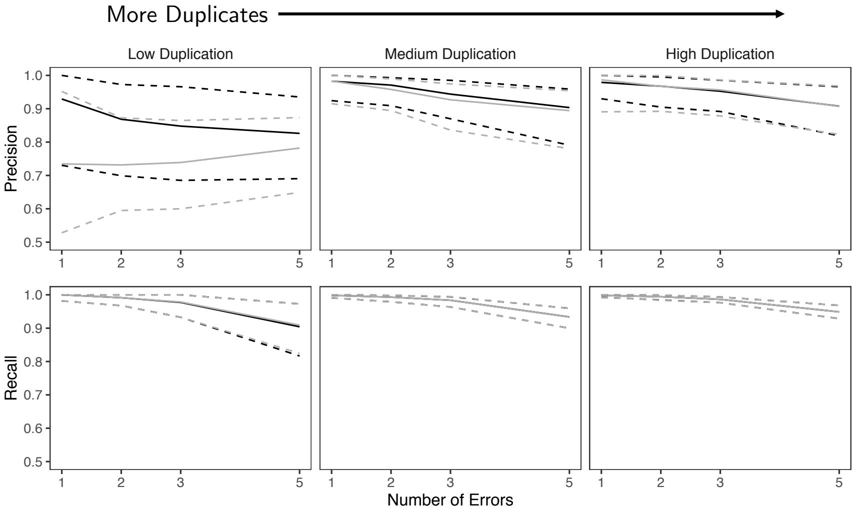



- Black is proposed prior, grey is flat prior, solid lines are medians, dotted lines are 2nd and 98th quantiles

# Simulation 2: Duplicates

- Vary amount of measurement error, duplication within files
  - Number of duplicates generated from Poisson with varying means, truncated to $\{1, \cdots, 5\}$
- Fix overlap to be low, $\sim 90\%$ of entities only in one file
- Comparisons between
  - Our model with Poisson(1) prior on duplicates, truncated to $\{1, \cdots, 10\}$
  - Model of Sadinle (2014) which uses a flat prior on partitions and treats all records as coming from one file
- Indexing to reduce number of comparisons
- Full Bayes estimates (not using abstain option)

# Simulation 2: Duplicates

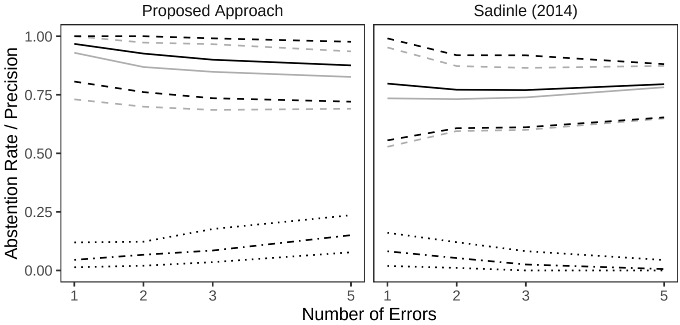



- Black is proposed approach, grey is Sadinle (2014), solid lines are medians, dotted lines are 2nd and 98th quantiles

# Simulation 3: Duplicates, Abstain Option

- Low Duplication setting from Simulation 2
- How does performance change when we use partial Bayes estimates (using the abstain option)?

# Simulation 3: Duplicates, Abstain Option




- Black are partial estimates, grey are full estimates, solid lines are medians, dotted lines are 2nd and 98th quantiles