From Multi-Site Observational Health Data to Real World Evidence: Privacy-Preserving Distributed Algorithms

Mackenzie Edmondson
PhD Candidate in Biostatistics
Department of Biostatistics, Epidemiology, and Informatics
University of Pennsylvania, Perelman School of Medicine

November 24, 2020 – OHDSI Community Call
Outline

‣ Background
  • Privacy challenges in multi-site studies
  • Existing approaches for privacy-preserving multi-site analysis

‣ Our approach: Privacy-preserving Distributed Algorithms (PDA)
  • Distributed regression; surrogate likelihood method
  • ODAP and ODAH
  • Real-world use cases

‣ Summary
  • Newly available R package!
Background: Privacy Challenges and Existing Approaches for Multi-Site Analysis
Privacy challenges in multi-site studies

- Multi-site studies: larger sample size, improved generalizability

- HIPAA: sharing of patient protected health information (PHI) often prohibited across institutions
  - De-identified data can be shared (e.g. “limited dataset”)

- De-identified PHI susceptible to re-identification (Benitez & Malin 2010)

- Distributed Health Data Networks: no data centralization
  - Common data model
  - Analyses performed distributively without patient-level data transfer
Existing Multi-Site Analysis Approaches: Meta-Analysis

- Collaborating sites send estimated coefficients and standard errors to lead site for aggregation

- Very popular, easy to implement
  - Most common analytic method in OHDSI studies

- Biased estimation in rare-event settings

- Issues with ecological bias
Distributed Regression

- Regression model fit in distributed fashion across sites without sharing patient-level data
- Involves aggregation of summary statistics to estimate parameters
- Multi-site distributed linear regression is **lossless** (Chen et al. 2006)
  - Estimated coefficients equivalent to those in pooled analysis

Pooled analysis: \( \hat{\beta}_{\text{pooled}} = (X^T X)^{-1} X^T Y \)

From each site \( i \), obtain \( (X_i^T X_i) \) and \( X_i^T Y_i \) (aggregated summary measures)

\[ \hat{\beta}_{\text{dist}} = (\sum_i X_i^T X_i)^{-1} (\sum_i X_i^T Y_i) = \hat{\beta}_{\text{pooled}} \]
Distributed Regression

- What if $\hat{\beta}$ doesn’t have a closed-form solution?
- Iterative procedures for distributed regression
  - Newton-Raphson method
  - Also lossless
- GLORE (distributed logistic regression)
- WebDISCO (distributed Cox regression)

Wu et al. 2012, JAMIA

Lu et al. 2015, JAMIA
Distributed Regression: Limitations in Existing Approaches

- Iterative procedures may require several rounds of communication
  - Privacy risk, even with aggregate data transfer (especially for very small data sets)
  - Inefficient, communication takes time!

- **Goal**: Can we perform distributed regression without using iterative procedure?
Our Approach: Privacy-preserving Distributed Algorithms (PDA)
PDA: Privacy-preserving Distributed Algorithms

1) Broadcast initial value

2) Share aggregated data

3) Synthesize evidence

Final results
Surrogate Likelihood Estimation

- **Communication-efficient** distributed inference (Jordan et al. 2018)
  - **One-shot**: Non-iterative communication among sites

- Approximates complete data (pooled) log-likelihood using **patient-level data at only one site (local site)**
  - Aggregate information obtained from collaborating (non-local) sites
  - Not lossless, but typically closer approximation than meta-analysis
  - Uses Taylor series expansion of complete data log-likelihood
  - First-order surrogate likelihood function:

\[
\tilde{L}(\beta) = L_1(\beta) + \{\nabla L(\bar{\beta}) - \nabla L_1(\bar{\beta})\} \beta
\]
Surrogate Likelihood Estimation: Intuition

Local

Site 2

Site 3

Initial estimate

Negative slope → shift left

Positive slope → shift right

Correcting shape of local likelihood

Surrogate likelihood function
Surrogate Likelihood Estimation

- ODAL: Algorithm for performing distributed logistic regression (Duan et al. 2020)
Surrogate Likelihood Estimation: Distributed Cox Regression

- ODAC: Algorithm for performing distributed Cox regression (Duan et al. 2020)
Communication-Efficient Distributed Regression for Count Outcomes (ODAP and ODAH)
ODAP Motivation: COVID-19 Hospitalization

- As of Sunday 11/22: > 58 million confirmed cases, 1.38 million deaths across 191 countries and territories (JHU COVID-19 Dashboard)

- Estimating demand for hospital beds crucial for contingency planning

- Length of stay (LoS) dependent on disease severity
  - Highly variable
RWD Motivation: COVID-19 Hospitalization

- Interest in characterizing association between LoS and patient characteristics

- Many individual sites with COVID-19 patient data, but typically too small for proper inference

- In a pandemic, many institutions willing to collaborate!

- **Goal:** Devise communication-efficient algorithm for modeling LOS in COVID-19 patients using data at several collaborating sites

- **Contribution:** ODAP (One-Shot Distributed Algorithm for performing Poisson regression)
ODAP: Distributed Poisson Regression Algorithm

- Count outcomes commonly modeled using Poisson regression
  - Assumption: mean = variance
  - Often in practice: mean < variance (overdispersion)
    - Poisson regression of overdispersed data results in biased standard errors (Cox 1984)

- Quasi-Poisson: account for extra variation in outcome by estimating dispersion and scaling variance
  - \( E(Y_i|X_i) = \exp(X_i^T \beta) = \mu_i \)
  - \( Var(Y_i|X_i) = \phi \mu_i, \ \phi > 0 \)
ODAP: Distributed Poisson Regression Algorithm

- Clinical data at $K$ sites, $j^{th}$ site has $n_j$ unique patient records, $N = \sum_{j=1}^{K} n_j$ total patient records.

- $(Y_{ij}, X_{ij})$: outcome, covariate vector for $i^{th}$ subject at $j^{th}$ site.

- Pooled, complete data log-likelihood function

$$ L_N(\beta) = \frac{1}{N} \sum_{j=1}^{K} \sum_{i=1}^{n_j} Y_{ij} X_{ij}^T \beta - \exp(X_{ij}^T \beta) $$

- Requires sharing of patient-level data across sites.
ODAP: Distributed Poisson Regression Algorithm

- Distributed data network: Assume we only have patient-level data at local site

- Surrogate log-likelihood function (second-order):

\[
\tilde{L}(\beta) = L_1(\beta) + \{\nabla L_N(\bar{\beta}) - \nabla L_1(\bar{\beta})\} \beta + \frac{1}{2} (\beta - \bar{\beta})^T \{\nabla^2 L_N(\bar{\beta}) - \nabla^2 L_1(\bar{\beta})\}(\beta - \bar{\beta})
\]

  - $\nabla, \nabla^2$: first- and second-order gradients of log-likelihood
  - $\nabla L_N$: weighted average of individual site gradients
  - $\bar{\beta}$: initial estimate for algorithm (e.g. meta-analysis estimate, local estimate)

- ODAP estimator: $\tilde{\beta} = \arg\max_\beta \tilde{L}(\beta)$

- $\nabla(\tilde{\beta})$: inverse Hessian scaled by overdispersion estimate $\hat{\phi}_a(\tilde{\beta})$
A. Initialization

1. Collaborating site estimates sent to local site.

2. Initial estimates computed (meta-analysis).

B. Surrogate Likelihood Estimation

3. Collaborating site gradients sent to local site.

4. $L(\hat{\beta})$ computed at local site; $\hat{\beta}$ and obtained, sent to each individual site.

C. Dispersion Estimation & Variance Calculation

5. Collaborating site dispersion estimates calculated, sent to local site.

6. $\phi_a(\tilde{\beta})$ calculated at local site, used for scaling variance.
Application: OneFlorida Clinical Research Consortium

- Limited dataset, patient-level RWD of 15 million Floridians (> 50% state population)
- Centralized data
- Q: In patients hospitalized with COVID-19, which risk factors are most associated with length of stay?
- Study data: 4,212 COVID-19 patients from 4 clinical sites
- High overdispersion: $\hat{\phi} \approx 10$ ($\hat{\phi} = 1$: no dispersion)
ODAP RWD Application: Results

95% confidence intervals of log relative risk estimates

- **Age**
- **Cancer**
- **CCI**
- **COPD**
- **Diabetes**
- **Heart Disease**
- **Hyperlipidemia**
- **Hypertension**
- **Kidney Disease**
- **Male**
- **Obesity**

- Local
- Meta
- ODAP
- Pooled

Local, Meta, ODAP, Pooled
ODAH Motivation: Serious Adverse Events (Pharmacovigilance)

- Post-market drug safety evaluated via adverse event reporting

- Real-world example: FOLFIRI chemotherapy treatment for colorectal cancer (CC)

- Interest in modeling serious adverse event (SAE) frequency for CC patients taking FOLFIRI
RWD Motivation: Severe Adverse Events (Pharmacovigilance)

- Most patients do not have SAE $\rightarrow$ many zero counts (e.g. > 80%)

- High variance in quantity and quality of adverse event reporting

- **Goal:** Devise communication-efficient algorithm to model SAE frequency using data at several collaborating sites

- **Contribution:** ODAH (One-Shot Distributed Algorithm for performing Hurdle regression)
ODAH: Distributed Hurdle Regression Algorithm

- Zero-inflation: another common feature of count data in practice
  - Excess zero counts -- more than would be expected under traditional count distribution (e.g. Poisson or Negative Binomial)
  - Often zero-inflated: length of stay, number of hospitalizations, lab tests ordered

- Methods for handling zero-inflated counts
  - Zero-inflated regression model
  - Hurdle model
ODAH: Distributed Hurdle Regression Algorithm

- Hurdle model: two-part model
  - "Zero" part: logistic regression
  - "Non-zero" part: zero-truncated count model (Poisson/Negative Binomial)

- No shared parameters: completely independent

- Interpretation differs from zero-inflated model
  - One source of zeros (sample) vs two (structural)

\[
P(w_i = 1) = \pi_i \\
\log \left( \frac{\pi_i}{1 - \pi_i} \right) = x_i^T \beta \\
w_i = 0 \hspace{1cm} w_i = 1 \\
y_i = 0 \\
P(Y_i = y_i | Y_i > 0) = \frac{e^{-\lambda_i} \lambda_i^{y_i}}{(1 - e^{-\lambda_i}) y_i!} \\
\log(\lambda_i) = z_i^T \gamma
\]
ODAH: Distributed Hurdle Regression Algorithm

- Log-likelihood of Poisson-Logit hurdle: sum of Binomial, zero-truncated Poisson log-likelihoods
  \[ L(\beta, \gamma) = L_1(\beta) + L_2(\gamma) \]

- Surrogate likelihood analogous to that for ODAP
  - Now have two surrogate log-likelihood functions, one for each component

- At most three rounds of non-iterative communication among sites
  - Depends on choice of initial value, estimation of dispersion
Real-World Data Application: OneFlorida CRC

- **Goal:** Assess drug safety in terms of severe adverse event (SAE) frequency
- **Q:** Given demographics and risk factors, how many SAEs are expected for colorectal cancer patient receiving FOLFIRI?
- **Data:** 660 colorectal cancer patients taking FOLFIRI from three clinical sites
ODAH RWD Application: Results

A. 95% confidence intervals of log odds ratio estimates

B. 95% confidence intervals of log relative risk estimates
Summary

- **PDA methods are:**
  - **Accurate:** High accuracy relative to pooled estimates; large advantage over meta-analysis in rare-outcome settings
  - **Safe:** At-most three rounds of communicating aggregate data; as little as one round
  - **Efficient:** Non-iterative communication among collaborating sites

- **ODAP/ODAH currently assume homogeneity, where statistical model is the same across all sites**
  - Estimating dispersion helps capture heterogeneity in outcome
  - Future extension: modify algorithms to further account for heterogeneity

- R package on CRAN soon and currently on GitHub; website in the works!
R package: pda

- GitHub repo: https://github.com/Penncil/pda

```r
# Install the latest version of PDA in R:
install.packages("pda")
library(pda)

# Or you can install via github:
install.packages("devtools")
library(devtools)
devtools::install_github("penncil/pda")
library(pda)
```
R pda Package Demo

- `pda::demo(ODAL)`

**PDA – ODAL (One-shot Distributed Algorithm for Logistic regression)**

```
status ~ age + sex
```

Local site 1

Site 2

Site 3
R pda Package Demo

- pda::demo(ODAL)

PDA – ODAL (One-shot Distributed Algorithm for Logistic regression)

Step 1: initialize
R pda Package Demo

- `pda::demo(ODAL)`

**PDA – ODAL (One-shot Distributed Algorithm for Logistic regression)**

Step 1: initialize
R pda Package Demo

- `pda::demo(ODAL)`

**PDA – ODAL (One-shot Distributed Algorithm for Logistic regression)**

- **Step 1: initialize**
  - Initial estimate

- **Local site 1**
  - Estimate (site 1)
  - Estimate (site 2)
  - Estimate (site 3)

- **Site 2**

- **Site 3**
R pda Package Demo

- pda::demo(ODAL)

**PDA – ODAL (One-shot Distributed Algorithm for Logistic regression)**

Step 2: derivative

Local site 1  
Site 2  
Site 3

Derivative of log-likelihood at initial estimate (site 2)
R pda Package Demo

- `pda::demo(ODAL)`

**PDA – ODAL (One-shot Distributed Algorithm for Logistic regression)**

Step 2: derivative

- Local site 1
- Site 2
- Site 3

Derivative of log-likelihood at initial estimate (site 3)
R pda Package Demo

- pda::demo(ODAL)

**PDA – ODAL (One-shot Distributed Algorithm for Logistic regression)**

Step 3: estimate

- Local site 1
- Site 2
- Site 3

Local data (site 1) + Derivative of log-likelihood at initial estimate (site 2, 3) → Surrogate estimate
Acknowledgements

Yong Chen, PhD
Chongliang Luo, PhD
Rui Duan, PhD
Jiayi Tong, BS

https://penncil.med.upenn.edu
Thank you!

Questions or ideas? Email me!
macjohn@pennmedicine.upenn.edu