

Model Fitting with Distributed Data

Department of Biomedical Data Sciences
and
Department of Statistics
Stanford University

March 7, 2018

Goal

- ▶ Make it possible to build statistical models on distributed data.
 - ▶ Let participant sites maintain access control
 - ▶ Use distributed computation, i.e., aggregate computation results on distributed data
- ▶ Create an extensible framework so that others can contribute methods
- ▶ Create a deployment framework using a well-established service.

Pooled Data

	Y	X_1	X_2	\cdots	X_p
1	Y_1	$X_{1,1}$	$X_{1,2}$	\cdots	$X_{1,p}$
2	Y_2	$X_{2,1}$	$X_{2,2}$	\cdots	$X_{2,p}$
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
n	Y_n	$X_{n,1}$	$X_{n,2}$	\cdots	$X_{n,p}$

Y is the *outcome* (response) and the X s are *covariates* (features or predictors). The numbers $1, 2, \dots, n$ are unique observation (row) numbers.

Row-partitioned data

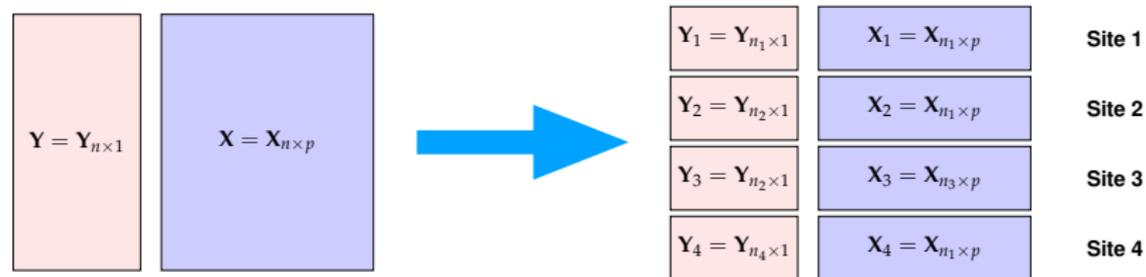
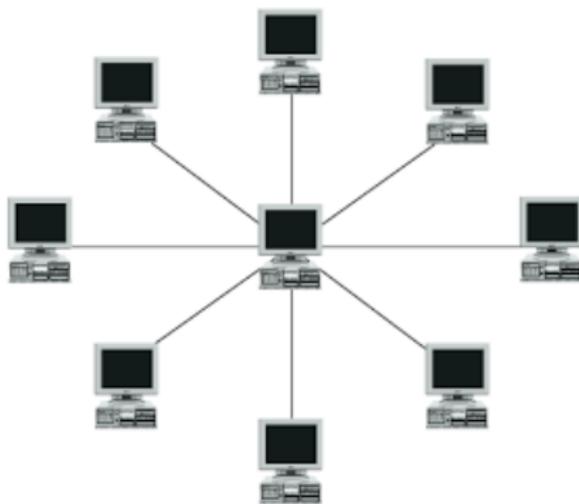


Figure: Left: Pooled data. Right: Distributed Data.

Communication Topology

A commonly used topology is the star network.



In a star network, a single node runs a master process that communicates with other nodes in the network that are clients or workers.

The final results are available at the master and can then be shared back to clients.

Common Models

- ▶ Standard Linear Regression: response Y on predictors X
- ▶ Poisson, Logistic, Multinomial
- ▶ Cox Regression
- ▶ Penalized versions of the above

In cancer studies, response is often *time to a particular event* (TTE), (death, relapse etc.)

A commonly used model for TTE is stratified Cox Regression.

Maximization via Newton-Raphson

K sites, $\ell_k(\beta)$, $S_k(\beta)$, $I_k(\beta)$ are site-specific likelihood, score and information matrix.

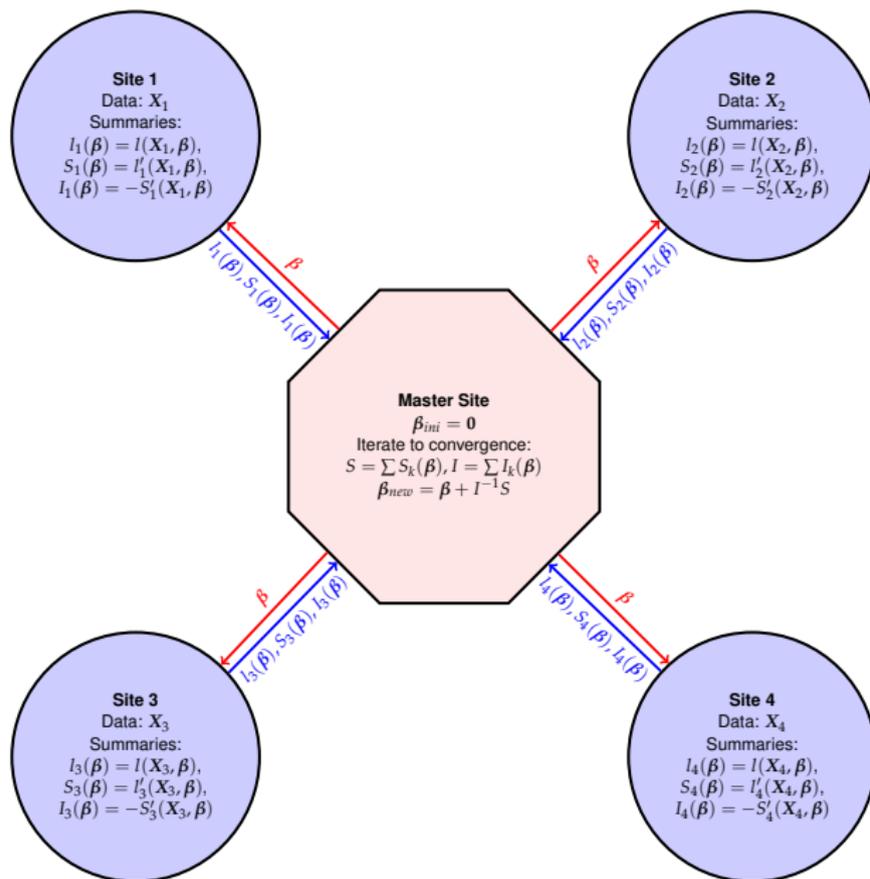
0. Set $i = 0$, $\beta_0 = 0$, a tolerance ϵ and a maximum number of iterations B .
1. Transmit β_i to each site
2. Each site k sends back $\ell_k(\beta_i)$, $S_k(\beta_i)$ and $I_k(\beta_i)$
3. Compute $\ell(\beta_i) = \sum_k \ell_k(\beta_i)$, $S(\beta_i) = \sum_k S_k(\beta_i)$, $I(\beta_i) = \sum_k I_k(\beta_i)$,
4. Set

$$\beta_{i+1} = \beta_i + I^{-1}(\beta_i)S(\beta_i)$$

5. Stop if converged or iteration count exceeded. Else increment i and repeat step 1.

For the Cox Model, the convergence is very fast.

Schematic



Workflow and Implementation

The prototype uses the R language and OpenCPU that exposes R as a RESTful (`http`) service.

The implementation is in the CRAN package **distcomp** described in N., Rubin, Gross, Bendersky, Lavori, *Journal of Statistical Software*, 77 (13) (2017).

The main steps are the following.

1. Define the Computation
2. Set up a Worker Process for the Computation
3. Set up a Master Process for the Computation
4. Run the Computation

We address each in turn.

1. Defining the Computation

Requirements: **R**, **distcomp** package



2. Setting up Worker



3. Setting up Master



Example: Treatment data on reducing drug abuse.

Pooled Data fit:

	coef	exp(coef)	se(coef)	z	Pr(> z)	
age	-0.028076	0.972315	0.008131	-3.453	0.000554	***
becktota	0.009146	1.009187	0.004991	1.832	0.066914	.
ndrugfp1	-0.521973	0.593349	0.124424	-4.195	2.73e-05	***
ndrugfp2	-0.194178	0.823512	0.048252	-4.024	5.72e-05	***
ivhx3TRUE	0.263634	1.301652	0.108243	2.436	0.014868	*
race	-0.240021	0.786611	0.115632	-2.076	0.037920	*
treat	-0.212616	0.808466	0.093747	-2.268	0.023331	*

Distributed Data fit:

	coef	exp(coef)	se(coef)	z	p
age	-0.0280495	0.97234	0.0081301	-3.4501	5.6041e-04
becktota	0.0091441	1.00919	0.0049918	1.8318	6.6979e-02
ndrugfp1	-0.5219296	0.59337	0.1244240	-4.1948	2.7315e-05
ndrugfp2	-0.1941709	0.82352	0.0482507	-4.0242	5.7168e-05
ivhx3TRUE	0.2636376	1.30166	0.1082448	2.4356	1.4868e-02
race	-0.2400609	0.78658	0.1156319	-2.0761	3.7887e-02
treat	-0.2125720	0.80850	0.0937466	-2.2675	2.3359e-02

Field Pilot

A distributed Stratified Cox Model fit for Stage 4 Breast Cancer data was fit using time to death (overall survival) among

- ▶ Stanford University
- ▶ Vanderbilt University
- ▶ Mt. Sinai Hospital
- ▶ Palo Alto Medical Foundation
- ▶ Group Health Seattle (now part of Kaiser).

Variables in the model, included *age*, *race*, whether they were on *chemo*, or *hormonal* or *targeted* therapies.

The model found significant coefficients for targeted and hormonal therapies. When Mt. Sinai was included, race was also significant.

Homomorphic Computations on Encrypted Data

Sites may be wary of sharing even summaries.

A *homomorphic encryption* scheme E is an encryption scheme such that:

$$E(x) + E(y) = E(x + y).$$

So, one can recover the value of $x + y$ by adding encrypted values of x and y and decrypting the sum!

One Implementation

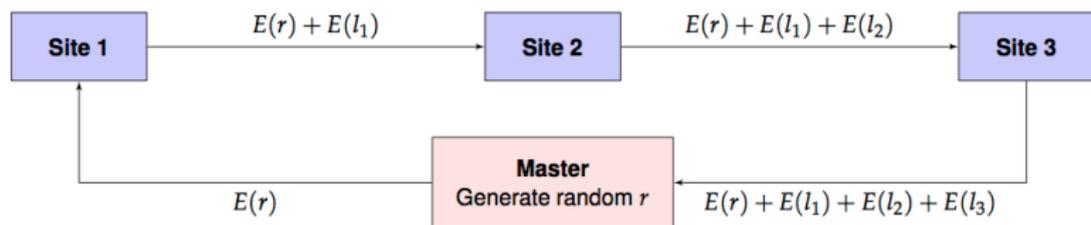


Figure: A homomorphic computation with three sites

Homomorphic Encryption ensures that

$$E(r) + E(l_1) + E(l_2) + E(l_3) = E(r + l_1 + l_2 + l_3) = e_l(\text{say}).$$

So we can decrypt $D(e_l) = e_l - r$ will yield $l_1 + l_2 + l_3$!

The R package `homomorpher` implements a prototype using Paillier encryption scheme.

Simulated Example

Three sites each with data sizes 1000, 500 and 1500 with

- ▶ sex (0, 1) for male/female
- ▶ age between 40 and 70
- ▶ a biomarker bm with a coefficient close to zero
- ▶ another biomarker $bm2$ with a non-zero coefficient
- ▶ a time to some event of interest
- ▶ an indicator event which is 1 if an event was observed and 0 otherwise.

Model is

$$S(t, age, sex, bm, bm2) = [S_0^i(t)]^{\exp(\beta_1 age + \beta_2 sex + \beta_3 bm + \beta_4 bm2)}$$

Results

Pooled Data fit:

	coef	exp(coef)	se(coef)	z	p
sex	-0.19015	0.82684	0.05063	-3.76	0.00017
age	0.01228	1.01236	0.00284	4.33	1.5e-05
bm	0.00311	1.00311	0.02622	0.12	0.90572
bm2	-0.08672	0.91693	0.02554	-3.40	0.00068

Distributed Data, Homomorphic fit:

	coef	exp(coef)	se(coef)	z	Pr(> z)
age	-0.19015	0.827	0.05063	-3.756	1.73e-04
sex	0.01228	1.012	0.00284	4.332	1.48e-05
bm	0.00311	1.003	0.02622	0.118	9.06e-01
bm2	-0.08672	0.917	0.02554	-3.396	6.85e-04

Challenges

- ▶ Data quality varies from site to site, especially follow-up
- ▶ Further privacy guarantees: differential privacy?
- ▶ Tools needed for data harmonization and social networking aspects of collaboration
- ▶ Deployment times at sites varied from 3 to 6 months
- ▶ IT folks don't know R!
- ▶ Security concerns, audit trails, fault tolerance