Semi-Parallel GWAS using RNS-CKKS

Miran Kim[1], Baiyu Li[2], Daniele Micciancio[2], Yongsoo Song[2]

1: UT Health Science Center at Houston, 2: UC San Diego

Background

Approximate Homomorphic Encryption

The Approximate Homomorphic Encryption scheme [CKKS]: (Cheon et al. ASIACRYPT 2017)

- Supports three operations:
  - Add : $(Enc(m_1), Enc(m_2)) \rightarrow Enc(m' \approx m_1 + m_2)$
  - Mult: $(Enc(m_1), Enc(m_2)) \rightarrow Enc(m' \approx m_1 m_2)$
  - Scale: $Enc(m) \rightarrow Enc(m' \approx \Delta^{-1} m)$

CKKS is suitable for encrypted computation on real numbers
- Scale operation can manage the growth of space and complexity

Single Logistic Regression

Training of Logistic Regression Model [iDASH17]: (Kim et al. BMC Med Genomics 2018)

- Binary Classifier: Given $(y_i, \boldsymbol{x}_i) \in \{\pm 1\} \times \mathbb{R}^k$, find $\boldsymbol{\beta} \in \mathbb{R}^{k+1}$ such that sign[$(1, \boldsymbol{x}_i) \cdot \boldsymbol{\beta}$] $= y_i$.
- Minimize the following loss using accelerated GD:

$$L(\boldsymbol{\beta}) = \sum_i \log \left[1 + \exp(-y_i (1, \boldsymbol{x}_i) \cdot \boldsymbol{\beta}) \right].$$

This Year's Task (Track 2)

Semi-parallel Genome Wide Association Studies:

Given $(y_i, \boldsymbol{x}_i, s_{ij}) \in \{\pm 1\} \times \mathbb{R}^{k+1}$, find β_{s_j} $(1 \le j \le p)$ such that

$$\text{sign}[(1, \boldsymbol{x}_i, s_{ij}) \cdot \boldsymbol{\beta}] = y_i \text{ for some } \boldsymbol{\beta} = (\boldsymbol{\beta}_X, \beta_{s_j}) \in \mathbb{R}^{k+2}.$$

- Step I: find a common logistic regression model $\boldsymbol{\beta}_X$ minimizing

$$L(\boldsymbol{\beta}_X) = \sum_i \log \left[1 + \exp(-y_i (1, \boldsymbol{x}_i) \cdot \boldsymbol{\beta}_X) \right]$$

- Step II: from $\boldsymbol{\beta} = (\boldsymbol{\beta}_X, 0)$, find individual β_{s_j} minimizing

$$\tilde{L}(\boldsymbol{\beta}^+) = \sum_i \log \left[1 + \exp\left(-y_i (1, \boldsymbol{x}_i, s_{ij}) \cdot \boldsymbol{\beta}^+\right) \right]$$

for some $\boldsymbol{\beta}^+ = (*, \beta_{s_j})$.

Our Solution

(Variant of) Full RNS Variant of CKKS

A Full RNS Variant of CKKS [RNS] (Cheon et al. SAC 2018):

- Use Residue Number System based on $q = p_1 p_2 \dots p_L$ where p_i's are distinct primes close to $p = 2^k$

- All computations are performed on RNS representation without high-precision arithmetic (GMP, NTL free)

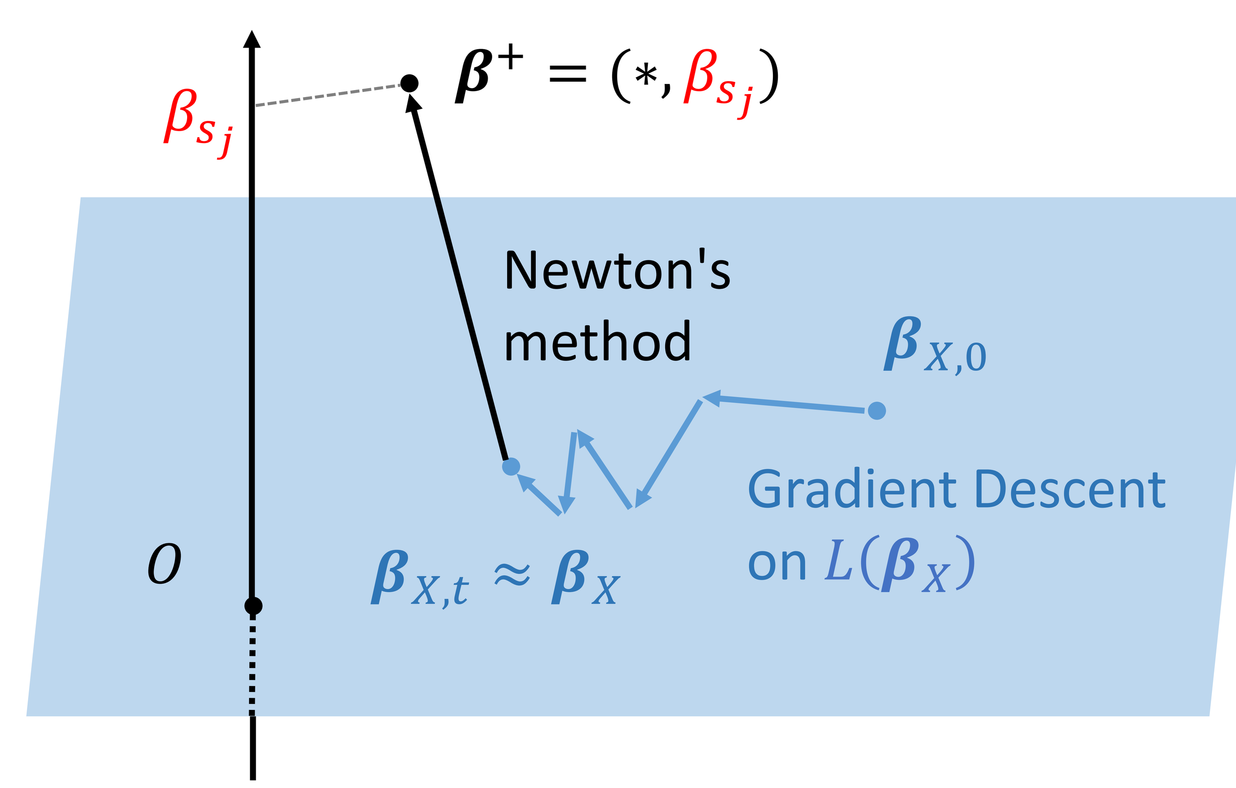
- Decomposition based key-switching (different from [RNS]) to minimize the parameter

Single GD & Individual (Parallel) NM

- Accelerated gradient descent [iDASH17] to build a model $\boldsymbol{\beta}_X$.

- Run (single-iteration) Newton's method to compute $\boldsymbol{\beta}^+$ by

$$\boldsymbol{\beta}^+ \leftarrow \boldsymbol{\beta} - \left(\nabla_{\boldsymbol{\beta}}^2 \tilde{L}\right)^{-1} \cdot \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta})$$



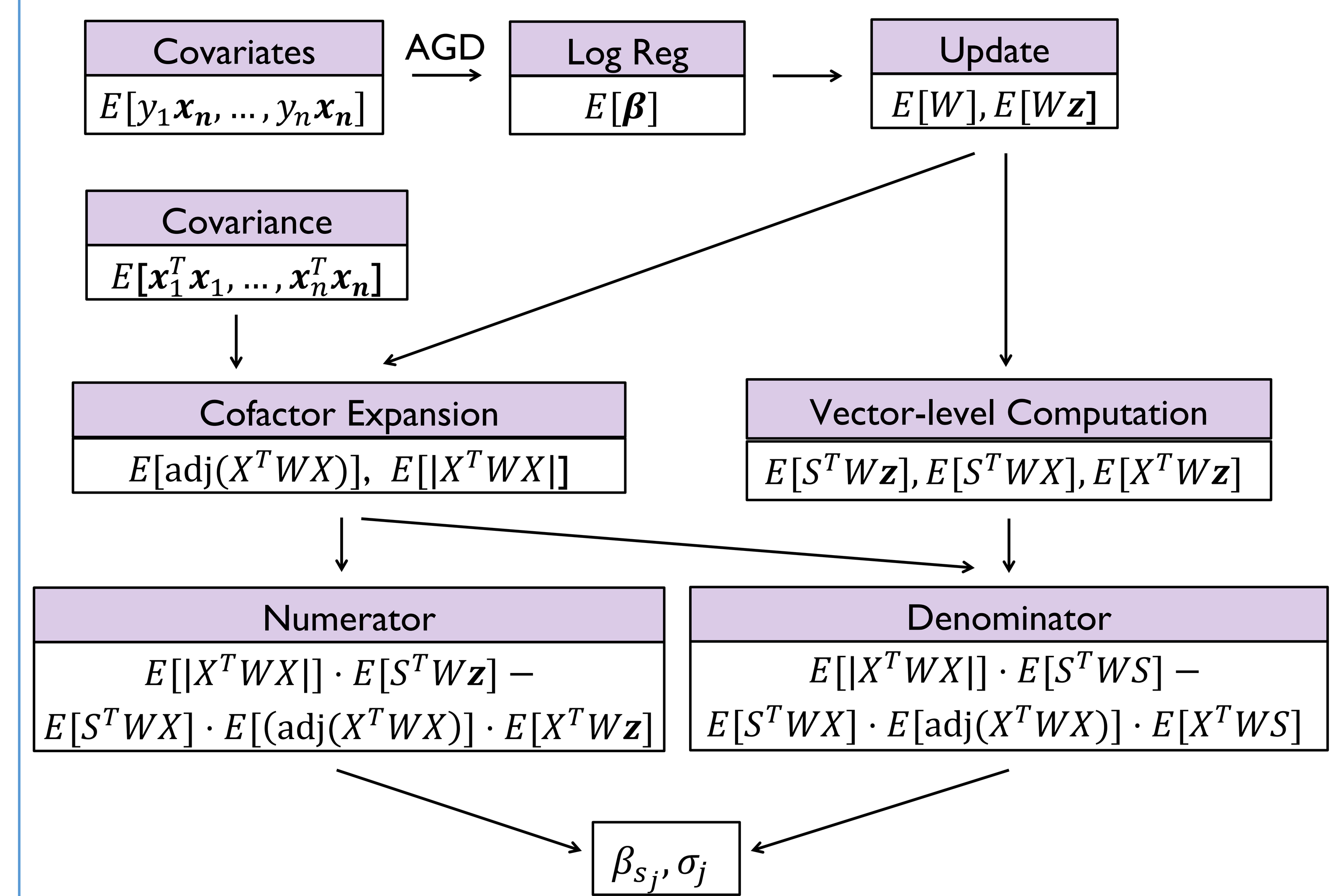
Newton's Method Formula

$$\boldsymbol{\beta}^+ = (*, \beta_{s_j}) \leftarrow \left(\tilde{X}^T W \tilde{X}\right)^{-1} \cdot \tilde{X}^T W \boldsymbol{z} \quad \text{where}$$

- $p_i = \sigma(\boldsymbol{x}_i^T \boldsymbol{\beta})$: probability predicted by a model w/o SNPs
- $W = \text{diag}(w_1, \dots, w_n)$ for $w_i = p_i \cdot (1 - p_i)$
- $\boldsymbol{z} = \left(\boldsymbol{x}_i^T \boldsymbol{\beta} + w_i^{-1}(y_i - p_i)\right)_{1 \le i \le n}$
- $\tilde{X} = (X, \boldsymbol{s}_j)$

$$\beta_{s_j} \leftarrow \frac{(\boldsymbol{s}_j{}^T W \boldsymbol{z}) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s}_j{}^T W \boldsymbol{s}_j) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s}_j)}, \quad 1 \le j \le p.$$

Evaluation Strategy



Adjugate Matrix and Determinant

Let $A = X^T W X$ of size 4×4. Its adjugate matrix $\text{adj}(A) = [b_{ij}]_{1 \le i,j \le 4}$ and determinant $|A|$ are computed by:

$$b_{11} = a_{22}a_{33}a_{44} + a_{23}a_{34}a_{42} + a_{24}a_{32}a_{43} - a_{24}a_{33}a_{42} - a_{23}a_{32}a_{44} - a_{22}a_{34}a_{43},$$

$$\dots$$

$$|A| = a_{11}a_{22}a_{33}a_{44} + \dots + a_{12}a_{24}a_{33}a_{41}.$$

- Use 'lazy' key-switching technique to reduce the complexity

Experimental Results

- Intel Core i5 @ 3.8GHZ processor
- Use the closed formula for linear regression method which minimizes the mean squared error (much faster, less accurate)

Method	$\log N$	$\log Q$	h	Timing				Memory	p-values
				KeyGen	Enc	Eval	Dec	Encrypted DB	(FP + FN)
Linear	13	245	130	0.12s	3.35s	1.61s	2.6ms	714MB	0.0059
Logistic	15	1060	170	7.14s	6.59s	53.66s	18 ms	1.7GB	0.0052

References

[CKKS] J. H. Cheon, A. Kim, M. Kim, & Y. Song, *Homomorphic encryption for arithmetic of approximate numbers – ASIACRYPT 2017.*

[iDASH17] A. Kim, Y. Song, M. Kim, K. Lee, & J. H. Cheon, *Logistic regression model training based on the approximate homomorphic encryption – BMC Med. Genomics, 2018.*

[RNS] J. H. Cheon, K. Han, A. Kim, M. Kim, & Y. Song, *A Full RNS Variant of Approximate Homomorphic Encryption – SAC 2018.*