Semi-parallel GWAS using RNS-CKKS

Miran Kim[⋆], Baiyu Li[†], Daniele Micciancio[†], Yongsoo Song[†]

⋆ University of Texas, Health Science Center at Houston

† University of California, San Diego

Oct. 15, 2018

Modifications of RNS-CKKS

- Base scheme: [CKKS]
  - Efficient for real number arithmetic

[CKKS] Homomorphic Encryption for Arithmetic of Approximate Numbers, Asiacrypt 2017

[RNS] A Full-RNS variant of Approximate Homomorphic Encryption, SAC 2018

Modifications of RNS-CKKS

- Base scheme: [CKKS]
  - Efficient for real number arithmetic

- RNS variant [RNS]
  - Residue Number System based on $q = p_1 p_2 \ldots p_L$ for distinct primes $p_i \approx p = 2^k$
  - All computation are performed on Residue Number System representation

[CKKS] Homomorphic Encryption for Arithmetic of Approximate Numbers, Asiacrypt 2017

[RNS] A Full-RNS variant of Approximate Homomorphic Encryption, SAC 2018

Modifications of RNS-CKKS

- Base scheme: [CKKS]
  - Efficient for real number arithmetic

- RNS variant [RNS]
  - Residue Number System based on $q = p_1 p_2 \dots p_L$ for distinct primes $p_i \approx p = 2^k$
  - All computation are performed on Residue Number System representation

- More optimizations (This work)
  - Decomposition based key-switching (vs modulus-raising)
  - Delay the key-switching & Compute on "Extended ctxt" (a.k.a. Lazy Key-Switching)

[CKKS] Homomorphic Encryption for Arithmetic of Approximate Numbers, Asiacrypt 2017

[RNS] A Full-RNS variant of Approximate Homomorphic Encryption, SAC 2018

Semi-parallel GWAS

- Given $\left(y_i, \boldsymbol{x}_i, s_{ij}\right) \in \{\pm 1\} \times \mathbb{R}^{k+1}$,

  find $\boldsymbol{\beta}_j = (\boldsymbol{\beta}_X, \beta_{\boldsymbol{s}_j}) \in \mathbb{R}^{k+2}$ such that $\text{sign}\left[\left(1, \boldsymbol{x}_i, s_{ij}\right) \cdot \boldsymbol{\beta}\right] = y_i$ for all j.

[Sikorska et al.] GWAS on your notebook, BMC bioinformatics 2013

Semi-parallel GWAS

- Given $(y_i, \boldsymbol{x}_i, s_{ij}) \in \{\pm 1\} \times \mathbb{R}^{k+1}$,

  find $\boldsymbol{\beta}_j = (\boldsymbol{\beta}_X, \beta_{\boldsymbol{s}_j}) \in \mathbb{R}^{k+2}$ such that $\mathrm{sign}\left[(1, \boldsymbol{x}_i, s_{ij}) \cdot \boldsymbol{\beta} \right] = y_i$ for all j.

- Semi-parallel Logistic Regression [Sikorska et al.]

[Sikorska et al.] GWAS on your notebook, BMC bioinformatics 2013

Semi-parallel GWAS

- Given $(y_i, \boldsymbol{x}_i, s_{ij}) \in \{\pm 1\} \times \mathbb{R}^{k+1}$,

  find $\boldsymbol{\beta}_j = (\boldsymbol{\beta}_X, \beta_{\boldsymbol{s}_j}) \in \mathbb{R}^{k+2}$ such that $\mathrm{sign}\left[(1, \boldsymbol{x}_i, s_{ij}) \cdot \boldsymbol{\beta}\right] = y_i$ for all j.

- Semi-parallel Logistic Regression [Sikorska et al.]
  - Step 1: Train a common (independent from j) model $\boldsymbol{\beta}_X$ minimizing

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

[Sikorska et al.] GWAS on your notebook, BMC bioinformatics 2013

Semi-parallel GWAS

- Given $(y_i, \boldsymbol{x}_i, s_{ij}) \in \{\pm 1\} \times \mathbb{R}^{k+1}$,

  find $\boldsymbol{\beta}_j = (\boldsymbol{\beta}_X, \beta_{\boldsymbol{s}_j}) \in \mathbb{R}^{k+2}$ such that $\mathrm{sign}\left[(1, \boldsymbol{x}_i, s_{ij}) \cdot \boldsymbol{\beta}\right] = y_i$ for all j.

- Semi-parallel Logistic Regression [Sikorska et al.]
  - Step 1: Train a common (independent from j) model $\boldsymbol{\beta}_X$ minimizing

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

  - Step 2: From $\boldsymbol{\beta} = (\boldsymbol{\beta}_X, 0)$, for each $\boldsymbol{s}_j$ $(1 \leq j \leq p)$, find $\beta_{\boldsymbol{s}_j}$ which minimizes

$$\tilde{L}(\boldsymbol{\beta}_j) = \sum \log\left[1 + \exp\left(-y_i\left(1, \boldsymbol{x}_i, s_{ij}\right) \cdot \boldsymbol{\beta}_j\right)\right] \text{ for some } \boldsymbol{\beta}_j = \left(*, \beta_{s_j}\right).$$

[Sikorska et al.] GWAS on your notebook, BMC bioinformatics 2013

Step 1: Common Logistic Regression

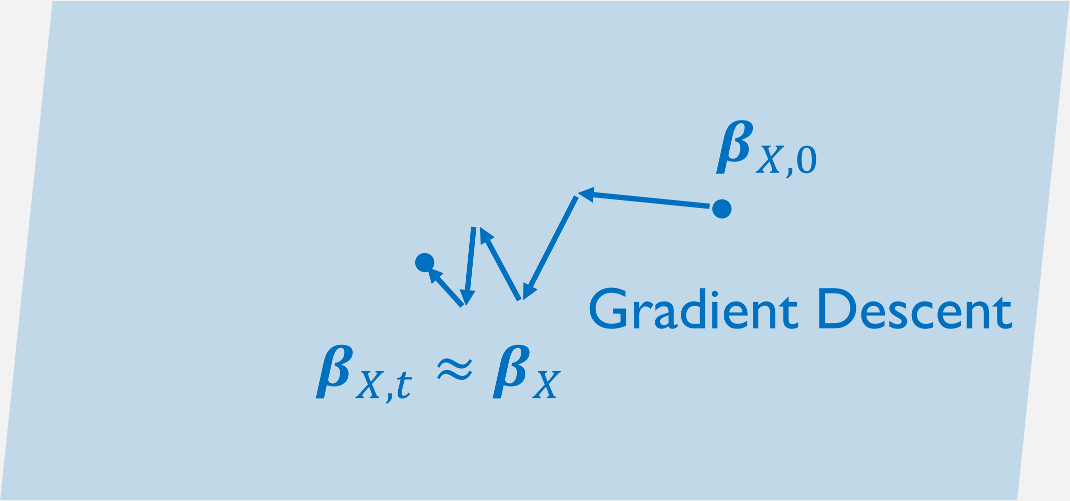
Train a common model $\boldsymbol{\beta}_X$ minimizing $L(\boldsymbol{\beta}_X) = \sum \log[1 + \exp(-(1, \boldsymbol{x}_i) \cdot \boldsymbol{\beta}_X)]$

[iDASH17] Logistic regression model training based on the approximate HE, BMC Med Genomics 2018.

Step 1: Common Logistic Regression

Train a common model $\boldsymbol{\beta}_X$ minimizing $L(\boldsymbol{\beta}_X) = \sum \log[1 + \exp(-(1, \boldsymbol{x}_i) \cdot \boldsymbol{\beta}_X)]$

- Gradient Decent method [iDASH17]



[iDASH17] Logistic regression model training based on the approximate HE, BMC Med Genomics 2018.

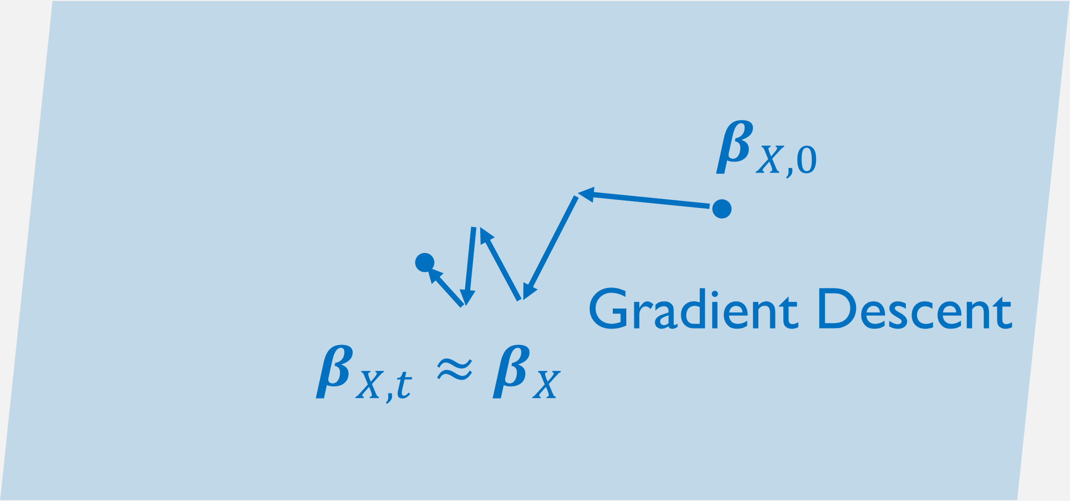
Step 1: Common Logistic Regression

Train a common model $\boldsymbol{\beta}_X$ minimizing $L(\boldsymbol{\beta}_X) = \sum \log[1 + \exp(-(1, \boldsymbol{x}_i) \cdot \boldsymbol{\beta}_X)]$

- Gradient Decent method [iDASH17]
  - Evaluate the formula recursively:

$$\boldsymbol{\beta}_{t+1} \leftarrow \boldsymbol{\beta}_t + \frac{1}{n} \cdot \sum_{i=1}^{n} \sigma_3(-(1, \boldsymbol{x}_i) \cdot \boldsymbol{\beta}_t) \cdot (1, \boldsymbol{x}_i)$$

  - Matrix encoding & accelerated GD

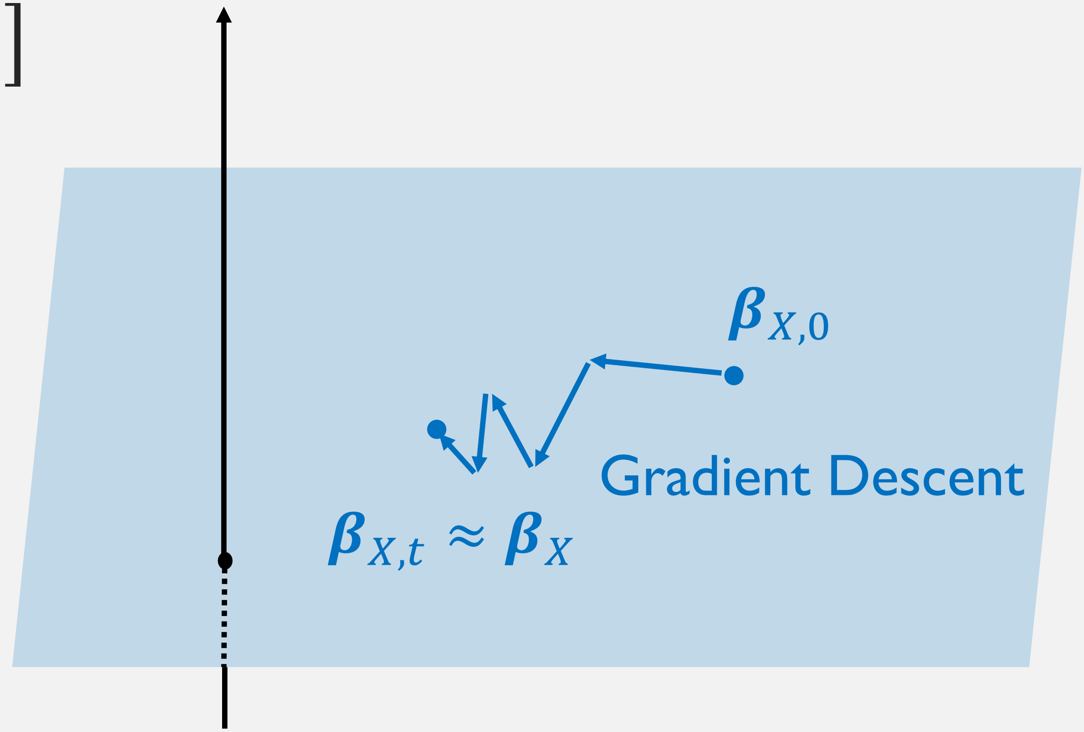


[iDASH17] Logistic regression model training based on the approximate HE, BMC Med Genomics 2018.

Step 2: Individual (parallel) Logistic Regression

From $\boldsymbol{\beta} = (\boldsymbol{\beta}_X, 0)$, find $\beta_{\boldsymbol{s}_j}$ of $\boldsymbol{\beta}^+ = \left(*, \beta_{s_j}\right)$ minimizing

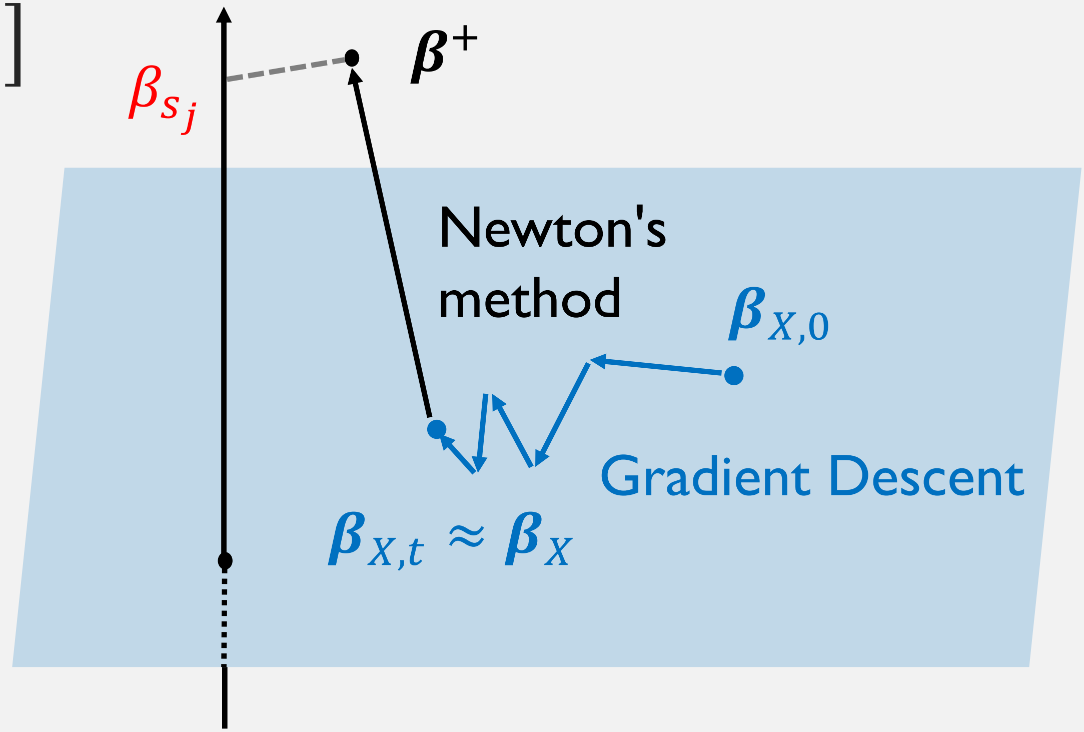
$$\tilde{L}(\boldsymbol{\beta}^+) = \sum \log\left[1 + \exp\left(-y_i\left(1, \boldsymbol{x}_i, s_{ij}\right) \cdot \boldsymbol{\beta}^+\right)\right]$$



Step 2: Individual (parallel) Logistic Regression

From $\boldsymbol{\beta} = (\boldsymbol{\beta}_X, 0)$, find $\beta_{\boldsymbol{s}_j}$ of $\boldsymbol{\beta}^+ = \left(*, \beta_{s_j}\right)$ minimizing

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



- Newton's method (single iteration)

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

for the Hessian matrix $\nabla^2_{\boldsymbol{\beta}} \tilde{L}$

Newton's method

- Single iteration at a starting point $\boldsymbol{\beta} = (\boldsymbol{\beta}_X, 0)$

$$\boldsymbol{\beta}^+ = \left(*, \beta_{\boldsymbol{s}_j}\right) \leftarrow \boldsymbol{\beta} - \left(\nabla^2_{\boldsymbol{\beta}} \tilde{L}\right)^{-1} \cdot \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta}) = \left(\tilde{X}^T W \tilde{X}\right)^{-1} \cdot \tilde{X}^T W \boldsymbol{z},$$

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

Newton's method

- Single iteration at a starting point $\boldsymbol{\beta} = (\boldsymbol{\beta}_X, 0)$

$$\boldsymbol{\beta}^{+} = \left(*, \beta_{\boldsymbol{s}_j}\right) \leftarrow \boldsymbol{\beta} - \left(\nabla_{\boldsymbol{\beta}}^2 \tilde{L}\right)^{-1} \cdot \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta}) = \left(\tilde{X}^T W \tilde{X}\right)^{-1} \cdot \tilde{X}^T W \boldsymbol{z},$$

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

$$\beta_{\boldsymbol{s}_j} = \frac{(\boldsymbol{s}_j{}^T W \boldsymbol{z}) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s}_j{}^T W \boldsymbol{s}_j) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s}_j)}$$

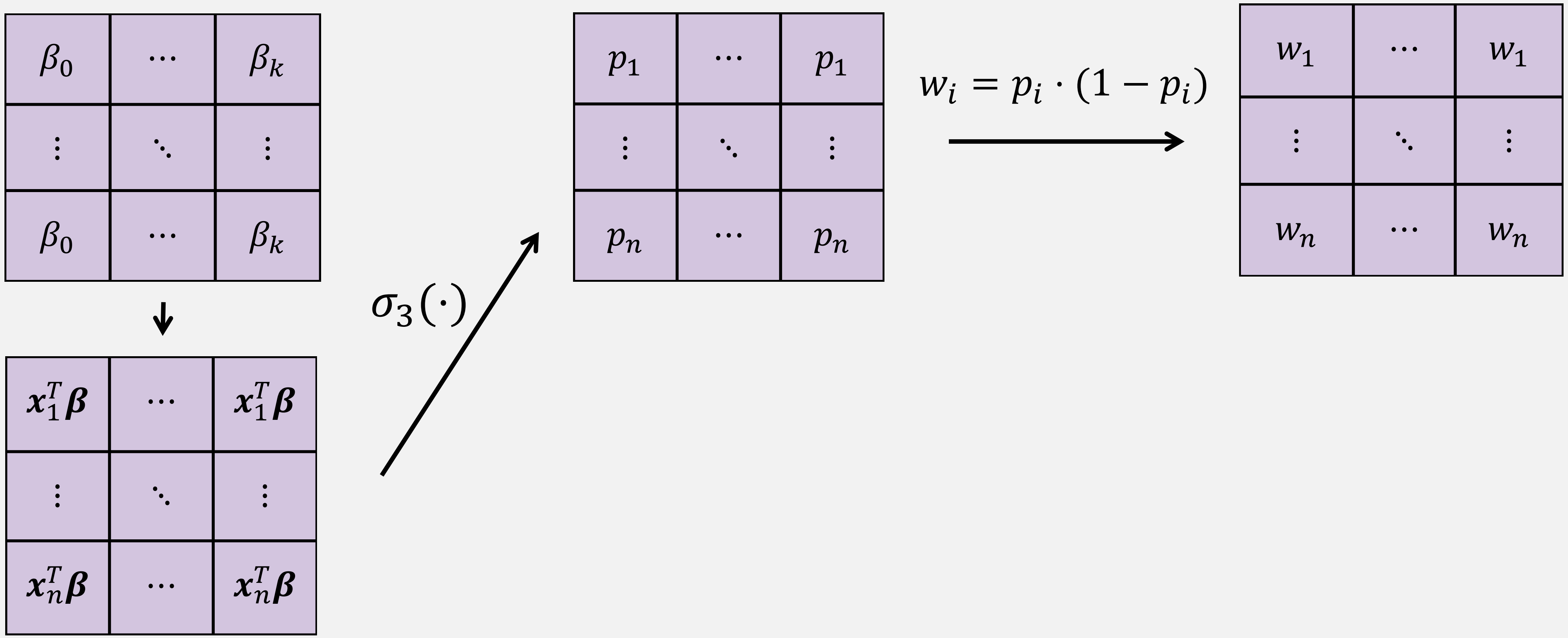
Evaluation Strategy 1

$$\beta_{s_j} = \frac{(\boldsymbol{s}_j{}^T W \boldsymbol{z}) \; - \; (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s}_j{}^T W \boldsymbol{s}_j) \; - \; (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s}_j)}$$

β_0	$\cdots$	β_k
$\vdots$	$\ddots$	$\vdots$
β_0	$\cdots$	β_k

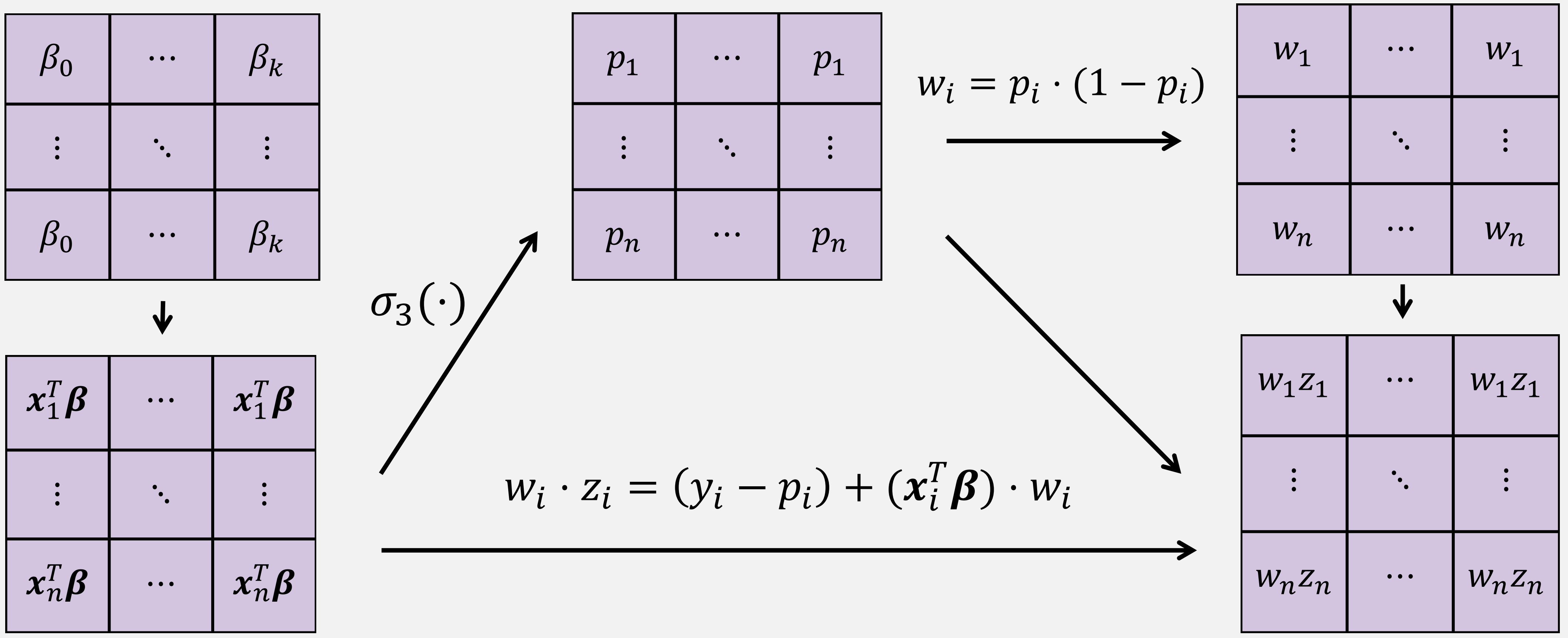
Evaluation Strategy I

$$\beta_{s_j} = \frac{(\mathbf{s}_j{}^T W\mathbf{z}) - (\mathbf{s}_j{}^T WX) \cdot (X^T WX)^{-1} \cdot (X^T W\mathbf{z})}{(\mathbf{s}_j{}^T W\mathbf{s}_j) - (\mathbf{s}_j{}^T WX) \cdot (X^T WX)^{-1} \cdot (X^T W\mathbf{s}_j)}$$



Evaluation Strategy I

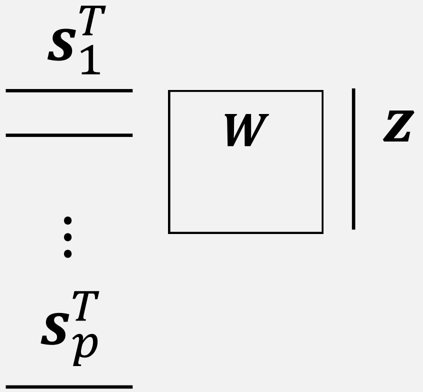
$$\beta_{s_j} = \frac{(\boldsymbol{s}_j{}^T W\boldsymbol{z}) - (\boldsymbol{s}_j{}^T WX) \cdot (X^T WX)^{-1} \cdot (X^T W\boldsymbol{z})}{(\boldsymbol{s}_j{}^T W\boldsymbol{s}_j) - (\boldsymbol{s}_j{}^T WX) \cdot (X^T WX)^{-1} \cdot (X^T W\boldsymbol{s}_j)}$$



Evaluation Strategy 2

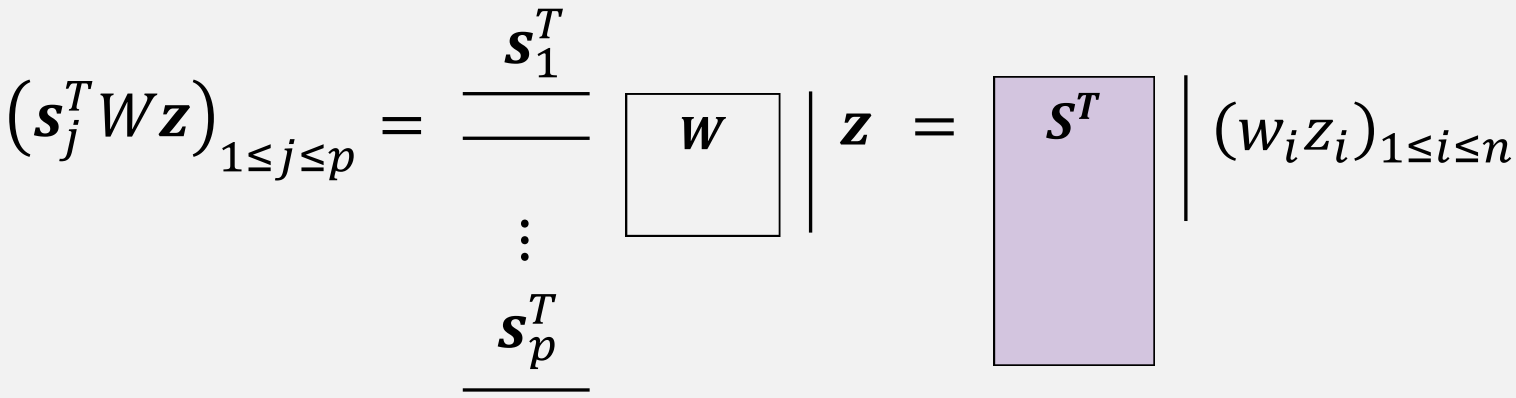
$$\beta_{s_j} = \frac{(\boldsymbol{s_j}^T W \boldsymbol{z}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s_j}^T W \boldsymbol{s_j}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s_j})}$$

$$\left(\boldsymbol{s}_j^T W \boldsymbol{z}\right)_{1 \le j \le p} =$$



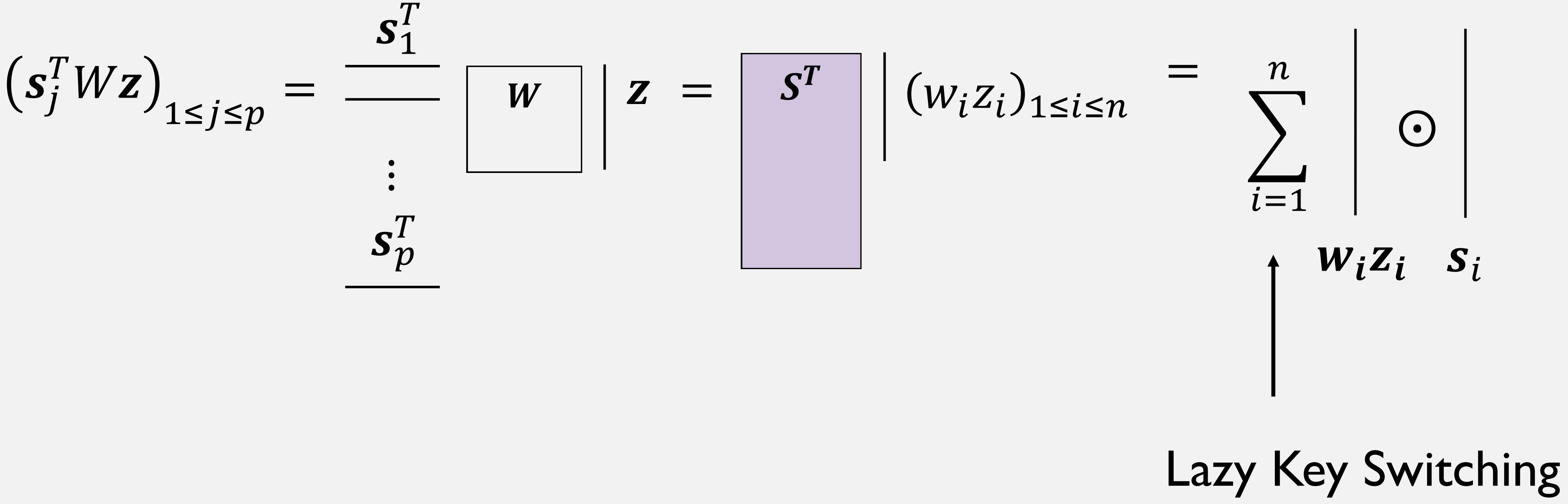
Evaluation Strategy 2

$$\beta_{s_j} = \frac{(\boldsymbol{s_j}^T W \boldsymbol{z}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s_j}^T W \boldsymbol{s_j}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s_j})}$$



Evaluation Strategy 2

$$\beta_{s_j} = \frac{(\boldsymbol{s}_j{}^T W \boldsymbol{z}) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s}_j{}^T W \boldsymbol{s}_j) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s}_j)}$$



Evaluation Strategy 3

$$\beta_{\boldsymbol{s}_j} = \frac{(\boldsymbol{s}_j{}^T W \boldsymbol{z}) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s}_j{}^T W \boldsymbol{s}_j) - (\boldsymbol{s}_j{}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s}_j)}$$

$A = X^T W X$ (size 4×4) $\rightarrow \mathrm{adj}(A) = \left((-1)^{k+\ell} \cdot |A_{k\ell}|\right)_{1 \le k,\ell \le 4}$

- $\mathrm{adj}(A)_{11} = a_{22}a_{33}a_{44} - a_{22}a_{34}a_{43} + a_{23}a_{34}a_{42} - a_{23}a_{32}a_{44} + a_{24}a_{32}a_{43} - a_{24}a_{33}a_{42}$
- $|A| = a_{11} \cdot \mathrm{adj}(A)_{11} + \cdots + a_{14} \cdot \mathrm{adj}(A)_{14}$

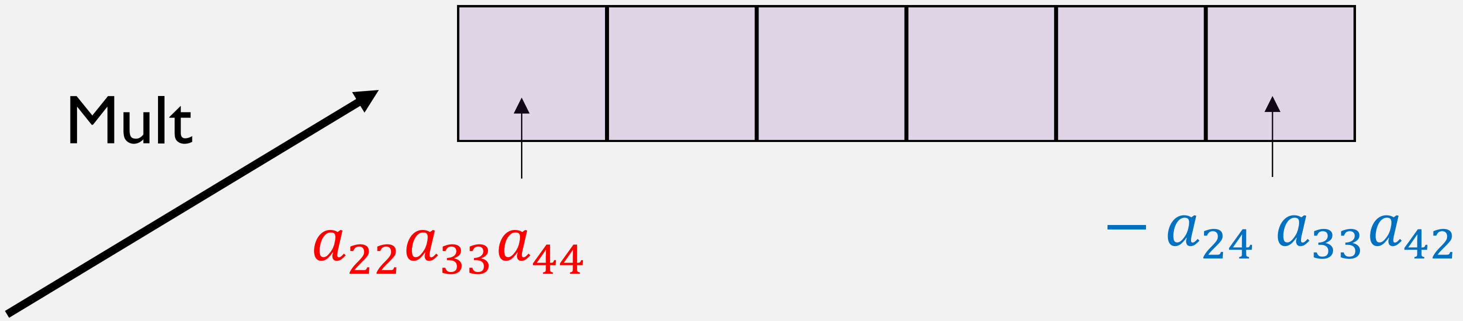
Evaluation Strategy 3

$$\beta_{s_j} = \frac{(\boldsymbol{s_j}^T W \boldsymbol{z}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s_j}^T W \boldsymbol{s_j}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s_j})}$$

$A = X^T W X$ (size 4×4) $\rightarrow \mathrm{adj}(A) = \left((-1)^{k+\ell} \cdot |A_{k\ell}|\right)_{1 \le k,\ell \le 4}$

- $\mathrm{adj}(A)_{11} = a_{22}a_{33}a_{44} - a_{22}a_{34}a_{43} + a_{23}a_{34}a_{42} - a_{23}a_{32}a_{44} + a_{24}a_{32}a_{43} - a_{24}a_{33}a_{42}$
- $|A| = a_{11} \cdot \mathrm{adj}(A)_{11} + \cdots + a_{14} \cdot \mathrm{adj}(A)_{14}$

a_{22}	$-a_{22}$	a_{23}	$-a_{23}$	a_{24}	$-a_{24}$
a_{33}	a_{34}	a_{34}	$-a_{32}$	a_{32}	a_{33}
a_{44}	a_{43}	a_{42}	a_{44}	a_{43}	a_{42}



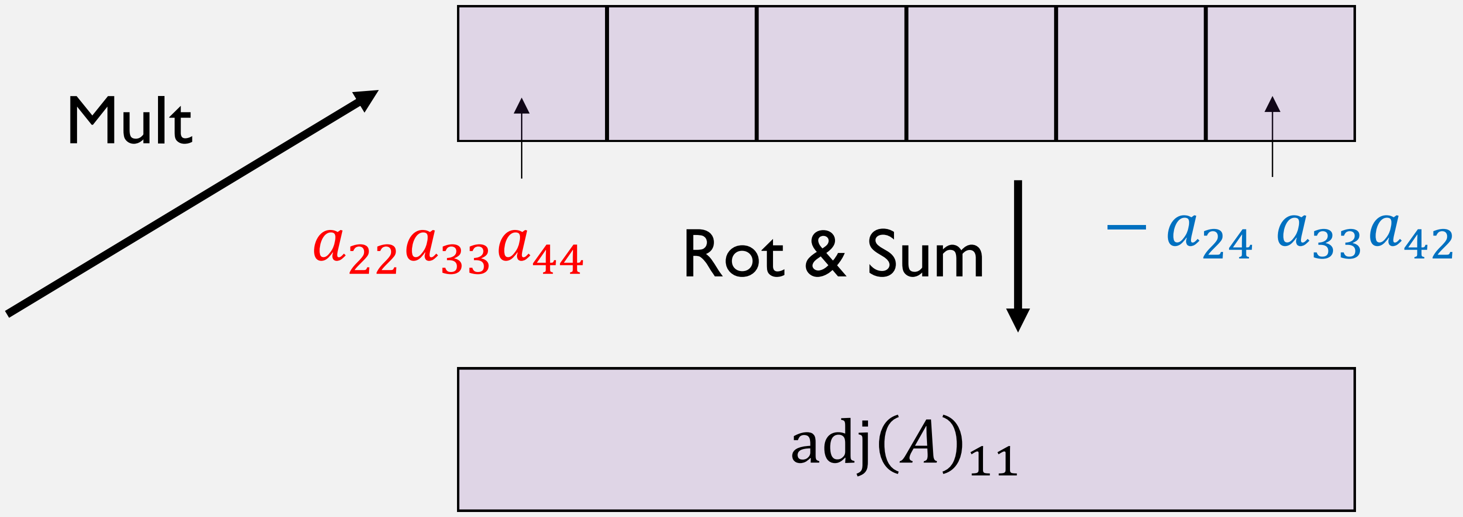
Evaluation Strategy 3

$$\beta_{s_j} = \frac{(\boldsymbol{s_j}^T W \boldsymbol{z}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{z})}{(\boldsymbol{s_j}^T W \boldsymbol{s_j}) - (\boldsymbol{s_j}^T W X) \cdot (X^T W X)^{-1} \cdot (X^T W \boldsymbol{s_j})}$$

$A = X^T W X$ (size 4×4) $\rightarrow \mathrm{adj}(A) = \left((-1)^{k+\ell} \cdot |A_{k\ell}|\right)_{1 \le k,\ell \le 4}$

- $\mathrm{adj}(A)_{11} = a_{22}a_{33}a_{44} - a_{22}a_{34}a_{43} + a_{23}a_{34}a_{42} - a_{23}a_{32}a_{44} + a_{24}a_{32}a_{43} - a_{24}a_{33}a_{42}$
- $|A| = a_{11} \cdot \mathrm{adj}(A)_{11} + \cdots + a_{14} \cdot \mathrm{adj}(A)_{14}$

a_{22}	-a_{22}	a_{23}	-a_{23}	a_{24}	-a_{24}
a_{33}	a_{34}	a_{34}	-a_{32}	a_{32}	a_{33}
a_{44}	a_{43}	a_{42}	a_{44}	a_{43}	a_{42}



Implementation Results

Intel Core i5 @ 3.8GHZ processor

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

Linear Regression

- Much faster (1.6 seconds) but less accurate
- Depth 3 evaluation (vs 22 of Logistic Regression)


Thank you and love you