Min-wise hashing for large-scale regression

Rajen D. Shah (Statistical Laboratory, University of Cambridge)
joint work with Nicolai Meinshausen (Seminar für Statistik, ETH Zürich)

University of Bristol
13 December 2013

High-dimensional regression: "large p, small n"

- The field of high-dimensional statistics has received a lot of attention in recent years.

High-dimensional regression: "large p, small n"

- The field of high-dimensional statistics has received a lot of attention in recent years.
- Microarray data has motivated much of the development. There the number of variables, p, can be in the tens of thousands, and the number of observations, n, may be less than a few hundred.

High-dimensional regression: "large p, small n"

- The field of high-dimensional statistics has received a lot of attention in recent years.
- Microarray data has motivated much of the development. There the number of variables, p, can be in the tens of thousands, and the number of observations, n, may be less than a few hundred.
- The low number of observations presents a formidable statistical challenge.
- To make progress, we often assume that the signal is sparse, and methods are developed take advantage of this sparsity.

Large-scale sparse regression

- **Text analysis**. Given a collection of documents, construct variables which count the number of occurrences of different words. Can add variables giving the frequency of pairs of words (bigrams) or triples of words (trigrams). An example from (Kogan, 2009) contains $4,272,227$ predictor variables for $n = 16,087$ documents.
- **URL reputation scoring** (Ma *et al.*, 2009). Information about a URL comprises $>$ 3 million variables which include word-stem presence and geographical information, for example. Number of observations, $n > 2$ million.

Large-scale sparse regression

- **Text analysis**. Given a collection of documents, construct variables which count the number of occurrences of different words. Can add variables giving the frequency of pairs of words (bigrams) or triples of words (trigrams). An example from (Kogan, 2009) contains $4,272,227$ predictor variables for $n = 16,087$ documents.
- **URL reputation scoring** (Ma *et al.*, 2009). Information about a URL comprises $>$ 3 million variables which include word-stem presence and geographical information, for example. Number of observations, $n > 2$ million.
- The size of the data presents both computational and statistical challenges.

Large-scale sparse regression

- **Text analysis**. Given a collection of documents, construct variables which count the number of occurrences of different words. Can add variables giving the frequency of pairs of words (bigrams) or triples of words (trigrams). An example from (Kogan, 2009) contains $4,272,227$ predictor variables for $n = 16,087$ documents.
- **URL reputation scoring** (Ma *et al.*, 2009). Information about a URL comprises $>$ 3 million variables which include word-stem presence and geographical information, for example. Number of observations, $n > 2$ million.
- The size of the data presents both computational and statistical challenges.
- How can we make progress? Exploit sparsity in the *design matrix*.

Sparsity

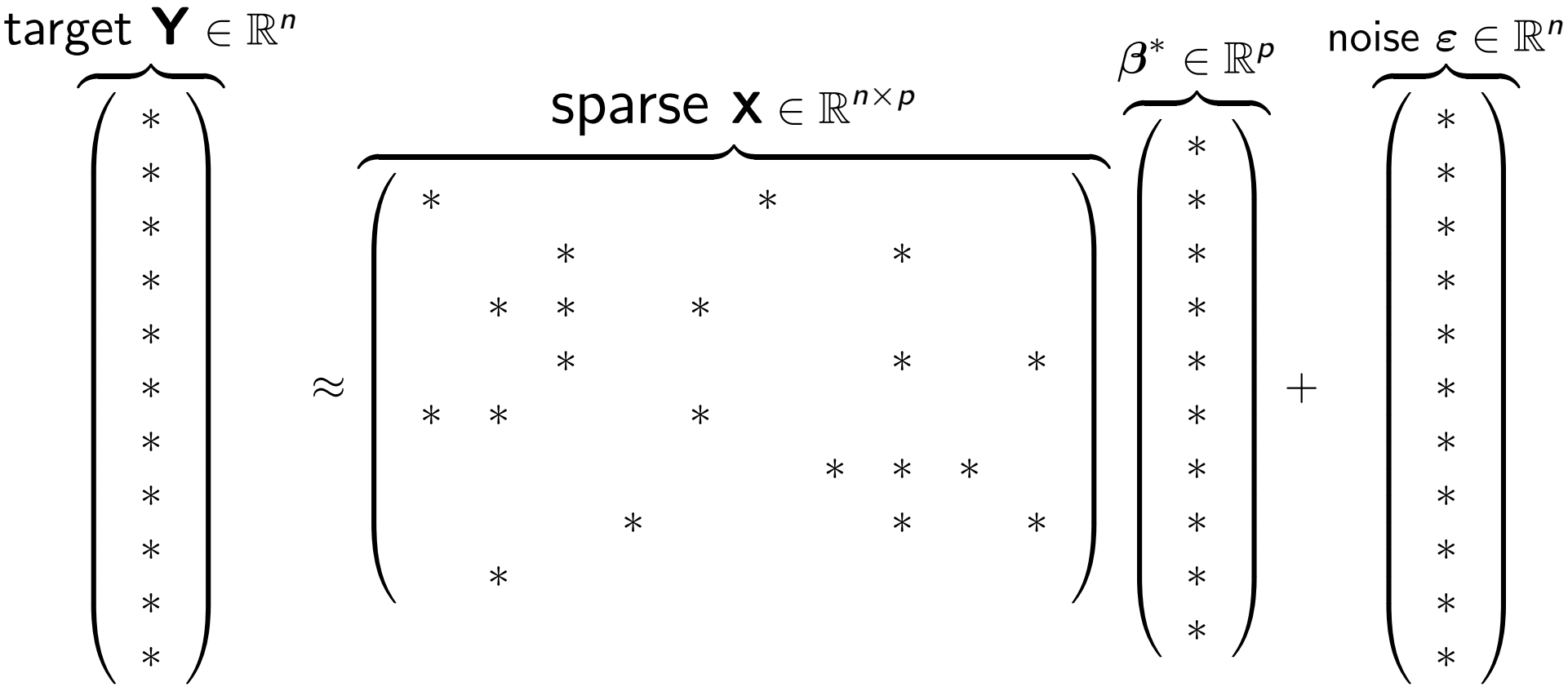
One observation in SVMlight format for the URL example.

```
+1 4:0.033195 5:0.0551724 6:0.0588235 11:0.142857 16:0.1 17:0.916974
19:0.209008 21:0.000494379 22:0.000496032 23:0.000496032 62:1 64:1 66:1
68:1 72:1 82:1 84:1 86:1 88:1 92:1 102:1 104:1 106:1 108:1 112:1 139:1
141:1 143:1 145:1 149:1 263:1 266:1 267:1 270:0.000496032 726:1 731:1
736:1 905:1 906:1 908:1 909:1 910:1 912:1 913:1 914:1 915:1 917:1 1629:1
2401:1 3521:1 3522:1 8197:1 8198:1 8199:1 8200:1 10728:1 155153:1
155154:1 155155:1 155156:1 155157:1 155158:1 155160:1 155161:1 155163:1
155164:1 155165:1 155174:1 155175:1 155176:1 155177:1 155178:1 155179:1
155180:1 155181:1 155182:1 155183:1 155184:1 155185:1 155186:1 155187:1
155188:1 155189:1 155190:1 155191:1 155192:1 155193:1 155194:1 155195:1
155196:1 155197:1 155198:1 155199:1 155200:1 155201:1 155202:1 155203:1
155204:1 155205:1 155206:1 155207:1 155208:1 155209:1 155210:1 155211:1
155212:1 155213:1 155214:1 455430:1 1554204:1 2528316:1
```

Dimension reduction

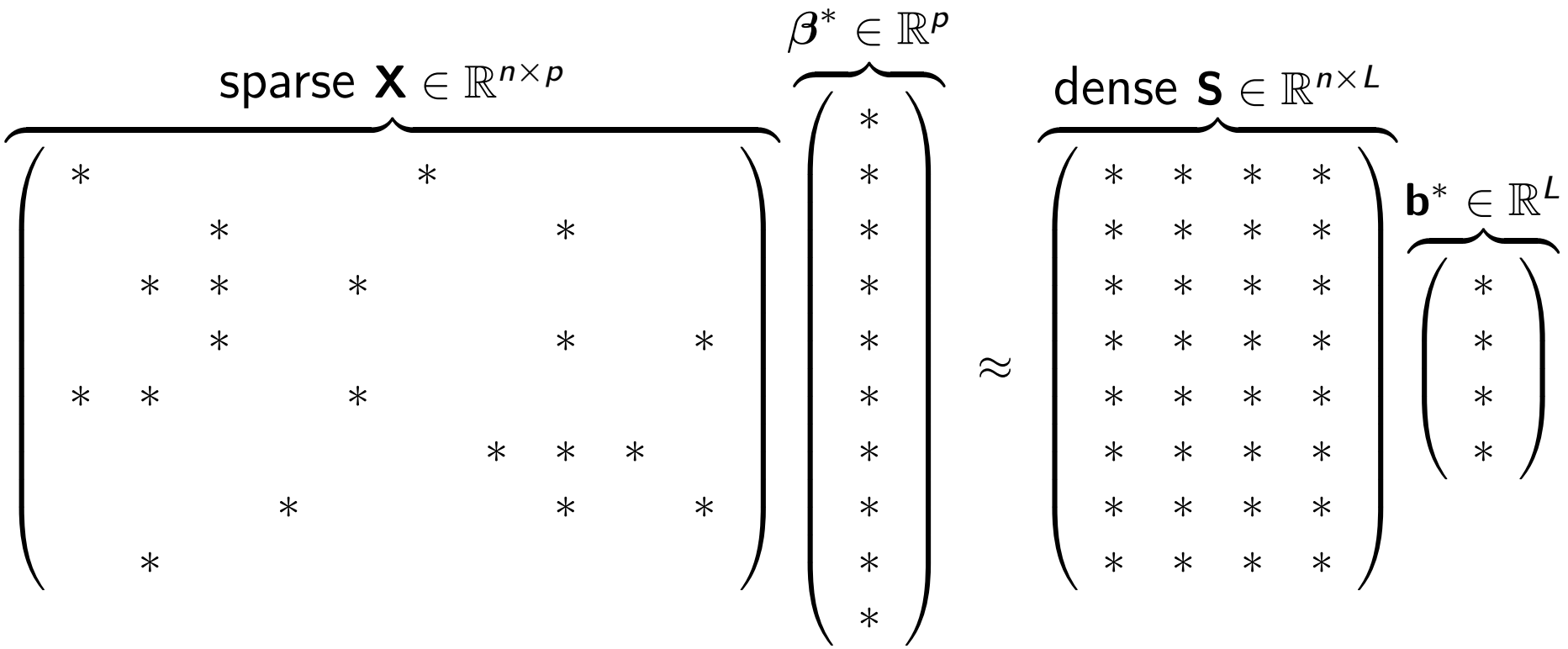
- Consider a regression problem in the large-scale setting, where $\mathbf{X}$ is a sparse $n \times p$ design matrix.
- The sparsity of $\mathbf{X}$ suggests it may be possible to construct a lower-dimensional $n \times L$ matrix $\mathbf{S}$ with $L \ll p$, which contains "most of the information in $\mathbf{X}$".
- We can then perform the regression on $\mathbf{S}$ rather than the larger $\mathbf{X}$.

Linear model with sparse design



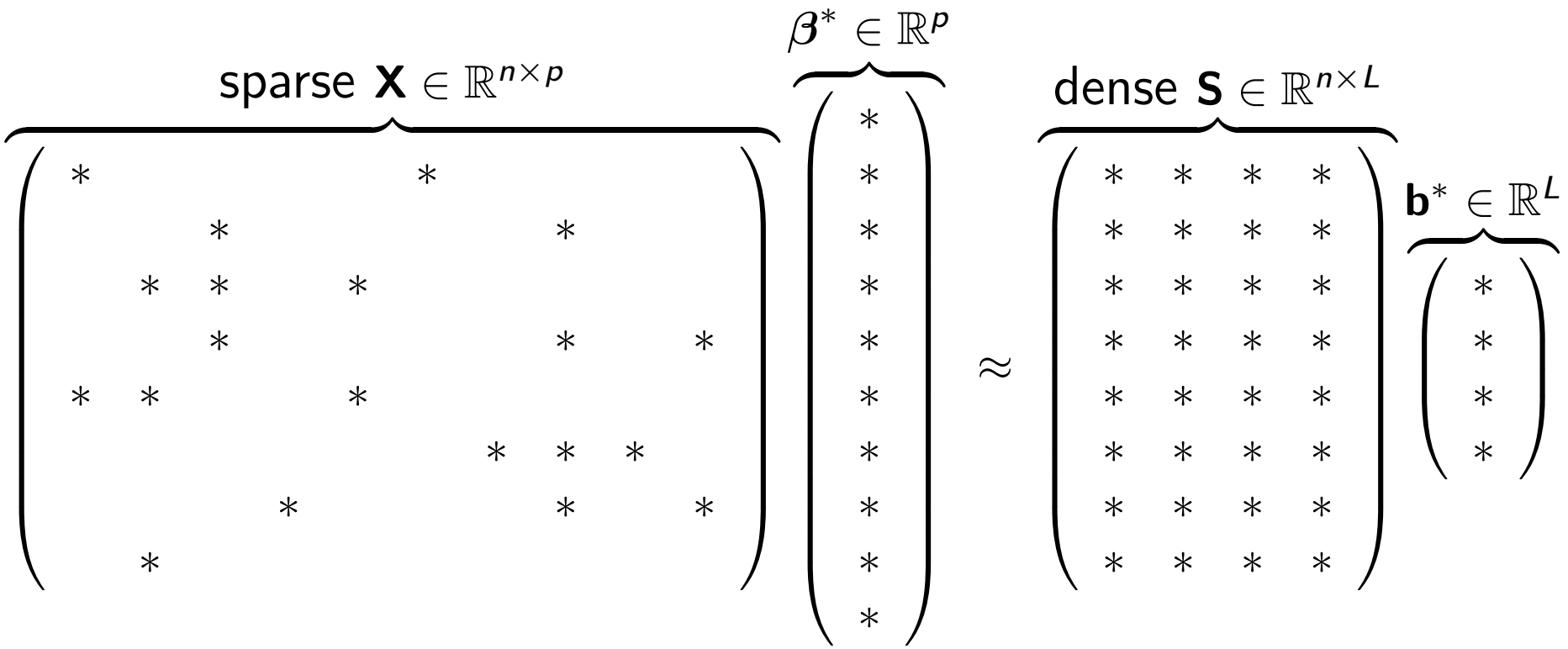
- Non-zero entries are marked with $*$.
- We could also consider a logistic regression model in a similar way.

Can we safely reduce sparse p-dimesional problem to a dense L-dimensional one with $L \ll p$?



- PCA is an obvious choice. However, it may be too computationally expensive to compute.

Can we safely reduce sparse p-dimesional problem to a dense L-dimensional one with $L \ll p$?



- PCA is an obvious choice. However, it may be too computationally expensive to compute.
- The approach we take here is based on *min-wise hashing*, and more specifically a variant called *b-bit min-wise hashing*.

Min-wise hashing (Broder, 1997; Broder *et al.*, 1998)

- Suppose we have sets $\mathbf{z}_1, \ldots, \mathbf{z}_n \subseteq \{1, \ldots, p\}$.
- Min-wise hashing gives estimates of the Jaccard index of every pair of sets $\mathbf{z}_i, \mathbf{z}_j$, given by

$$J(\mathbf{z}_i, \mathbf{z}_j) = \frac{|\mathbf{z}_i \cap \mathbf{z}_j|}{|\mathbf{z}_i \cup \mathbf{z}_j|}.$$

Min-wise hashing (Broder, 1997; Broder *et al.*, 1998)

- Suppose we have sets $\mathbf{z}_1, \ldots, \mathbf{z}_n \subseteq \{1, \ldots, p\}$.
- Min-wise hashing gives estimates of the Jaccard index of every pair of sets $\mathbf{z}_i, \mathbf{z}_j$, given by

$$J(\mathbf{z}_i, \mathbf{z}_j) = \frac{|\mathbf{z}_i \cap \mathbf{z}_j|}{|\mathbf{z}_i \cup \mathbf{z}_j|}.$$

- Let $\pi_1, \ldots, \pi_L$ be random permutations of $\{1, \ldots, p\}$ (in practice all random functions would be implemented by hash functions).
- Let the $n \times L$ matrix $\mathbf{M}$ be given by

$$M_{il} = \min_{k \in \mathbf{z}_i} \pi_l(k).$$

Min-wise hashing (Broder, 1997; Broder *et al.*, 1998)

- Suppose we have sets $\mathbf{z}_1, \ldots, \mathbf{z}_n \subseteq \{1, \ldots, p\}$.
- Min-wise hashing gives estimates of the Jaccard index of every pair of sets $\mathbf{z}_i, \mathbf{z}_j$, given by

$$J(\mathbf{z}_i, \mathbf{z}_j) = \frac{|\mathbf{z}_i \cap \mathbf{z}_j|}{|\mathbf{z}_i \cup \mathbf{z}_j|}.$$

- Let $\pi_1, \ldots, \pi_L$ be random permutations of $\{1, \ldots, p\}$ (in practice all random functions would be implemented by hash functions).
- Let the $n \times L$ matrix $\mathbf{M}$ be given by

$$M_{il} = \min_{k \in \mathbf{z}_i} \pi_l(k).$$

- Then for each i, j, l, $\mathbb{P}(M_{il} = M_{jl}) = J(\mathbf{z}_i, \mathbf{z}_j)$.

Min-wise hashing (Broder, 1997; Broder *et al.*, 1998)

- Suppose we have sets $\mathbf{z}_1, \ldots, \mathbf{z}_n \subseteq \{1, \ldots, p\}$.
- Min-wise hashing gives estimates of the Jaccard index of every pair of sets $\mathbf{z}_i, \mathbf{z}_j$, given by

$$J(\mathbf{z}_i, \mathbf{z}_j) = \frac{|\mathbf{z}_i \cap \mathbf{z}_j|}{|\mathbf{z}_i \cup \mathbf{z}_j|}.$$

- Let $\pi_1, \ldots, \pi_L$ be random permutations of $\{1, \ldots, p\}$ (in practice all random functions would be implemented by hash functions).
- Let the $n \times L$ matrix $\mathbf{M}$ be given by

$$M_{il} = \min_{k \in \mathbf{z}_i} \pi_l(k).$$

- Then for each i, j, l, $\mathbb{P}(M_{il} = M_{jl}) = J(\mathbf{z}_i, \mathbf{z}_j)$.
- In our context, let $\mathbf{X}$ be a sparse binary design matrix and let $\mathbf{z}_i$ record the indices of the non-zero entries of the i^{th} row of $\mathbf{X}$, $\mathbf{x}_i$.

Min-wise hashing matrix $\mathbf{M}$

$$\mathbf{X} = \begin{array}{c} \pi_1 \\ \\ \\ \\ \\ \end{array} \begin{array}{c} 3 \\ \\ \\ * \\ \\ * \end{array} \begin{array}{c} 1 \\ * \\ \\ \\ * \\ * \end{array} \begin{array}{c} 2 \\ \\ * \\ * \\ * \\ \\ \end{array} \begin{array}{c} 4 \\ * \\ * \\ \\ \\ \\ \end{array} \quad \mapsto \quad \mathbf{M} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \\ 1 \end{pmatrix}$$

One column of $\mathbf{M}$ generated by the random permutation π of the variables.

Min-wise hashing matrix $\mathbf{M}$

$$\mathbf{X} = \begin{array}{c} \pi_2 \\ \\ \\ \\ \\ \end{array} \begin{array}{c} \begin{array}{cccc} 2 & 4 & 1 & 3 \end{array} \\ \begin{pmatrix} & * & & * \\ & & * & * \\ * & & * & \\ & * & * & \\ * & * & & \end{pmatrix} \end{array} \mapsto \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$$

- Idea: work with $\mathbf{M}$ instead of sparse $\mathbf{X}$.

Min-wise hashing matrix $\mathbf{M}$

$$\mathbf{X} = \begin{array}{c} \pi_2 \\ \\ \\ \\ \\ \end{array} \begin{array}{c} 2 \\ \\ \\ \\ \\ \end{array} \begin{array}{c} 4 \\ \\ \\ \\ \\ \end{array} \begin{array}{c} 1 \\ \\ \\ \\ \\ \end{array} \begin{array}{c} 3 \\ \\ \\ \\ \\ \end{array} \begin{pmatrix} & * & & * \\ & & * & * \\ * & & * & \\ & * & * & \\ * & * & & \end{pmatrix} \quad \mapsto \quad \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix}$$

- Idea: work with $\mathbf{M}$ instead of sparse $\mathbf{X}$.
- Encode all levels in a column as dummy variables.

b-bit min-wise hashing (Li and König, 2011)

b-bit min-wise hashing stores only the lowest b bits of each entry of $\mathbf{M}$ when expressed in binary (i.e. the residue mod 2), so for $b = 1$,

$$M_{il}^{(1)} \equiv M_{il} \pmod 2.$$

b-bit min-wise hashing (Li and König, 2011)

b-bit min-wise hashing stores only the lowest b bits of each entry of $\mathbf{M}$ when expressed in binary (i.e. the residue mod 2), so for $b = 1$,

$$M_{il}^{(1)} \equiv M_{il} \pmod 2.$$

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & 1 & & \end{pmatrix} \mapsto \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \mapsto \mathbf{M}^{(1)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

b-bit min-wise hashing (Li and König, 2011)

b-bit min-wise hashing stores only the lowest b bits of each entry of $\mathbf{M}$ when expressed in binary (i.e. the residue mod 2), so for $b = 1$,

$$M_{il}^{(1)} \equiv M_{il} \pmod 2.$$

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & 1 & & \end{pmatrix} \mapsto \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \mapsto \mathbf{M}^{(1)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

- Perform regression using binary $n \times L$ matrix $\mathbf{M}^{(1)}$ rather than $\mathbf{X}$.
- When $L \ll p$ this gives large computational savings, and empirical studies report good performance (mostly for classification with SVM's).

MRS mapping

Will study a variant of 1-bit min-wise hashing we call MRS-mapping (**m**in-wise hash **r**andom **s**ign)

- Easier to analyse.
- Deals with sparse design matrices with real-valued entries.
- Allows for the construction of a variable importance measure.

Downside: slightly less efficient to implement.

MRS-mapping

1-bit min-wise hashing: **keep last bit**

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & 1 & & \end{pmatrix} \quad \mapsto \quad \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \quad \mapsto \quad \mathbf{M}^{(1)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

MRS-mapping

1-bit min-wise hashing: **keep last bit**

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & 1 & & \end{pmatrix} \quad \mapsto \quad \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \quad \mapsto \quad \mathbf{M}^{(1)} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}$$

MRS-map: **random sign assigments** $\{1, \ldots, p\} \to \{-1, 1\}$ are chosen independently for all columns $l = 1, \ldots, L$ when going from $\mathbf{M}_l$ to $\mathbf{S}_l$.

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & 1 & & \end{pmatrix} \quad \mapsto \quad \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \quad \mapsto \quad \mathbf{S} = \begin{pmatrix} 1 & -1 \\ -1 & -1 \\ -1 & -1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix}$$

The **H** and **M** matrices

Rather than storing **M**, we can store the "responsible" variables in **H**

$$M_{il} = \min_{k \in \mathbf{z}_i} \pi_l(\mathbf{z}_i)$$

$$H_{il} = \text{argmin}_{k \in \mathbf{z}_i} \pi_l(k)$$

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & 1 & & \end{pmatrix} \quad \mapsto \quad \mathbf{M} = \begin{pmatrix} 1 & 3 \\ 2 & 1 \\ 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{pmatrix} \quad \mapsto \quad \mathbf{S} = \begin{pmatrix} 1 & -1 \\ -1 & -1 \\ -1 & -1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix}$$

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 1 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 1 & 1 & & \end{pmatrix} \quad \mapsto \quad \mathbf{H} = \begin{pmatrix} 2 & 4 \\ 3 & 3 \\ 3 & 3 \\ 2 & 3 \\ 2 & 1 \end{pmatrix} \quad \mapsto \quad \mathbf{S} = \begin{pmatrix} 1 & -1 \\ -1 & -1 \\ -1 & -1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix}$$

Continuous variables

By using $\mathbf{H}$ rather than $\mathbf{M}$, we can handle continuous variables.

$$\mathbf{X} = \begin{pmatrix} & 1 & & 1 \\ & & 4.2 & 1 \\ 1 & & 1 & \\ & 1 & 1 & \\ 7.1 & 1 & & \end{pmatrix} \quad \mapsto \quad \mathbf{H} = \begin{pmatrix} 2 & 4 \\ 3 & 3 \\ 3 & 3 \\ 2 & 3 \\ 2 & 1 \end{pmatrix} \quad \mapsto \quad \mathbf{S} = \begin{pmatrix} 1 & -1 \\ -4.2 & -4.2 \\ -1 & -1 \\ 1 & -1 \\ 1 & 7.1 \end{pmatrix}$$

We get $n \times L$ matrices $\mathbf{H}$, and $\mathbf{S}$ given by

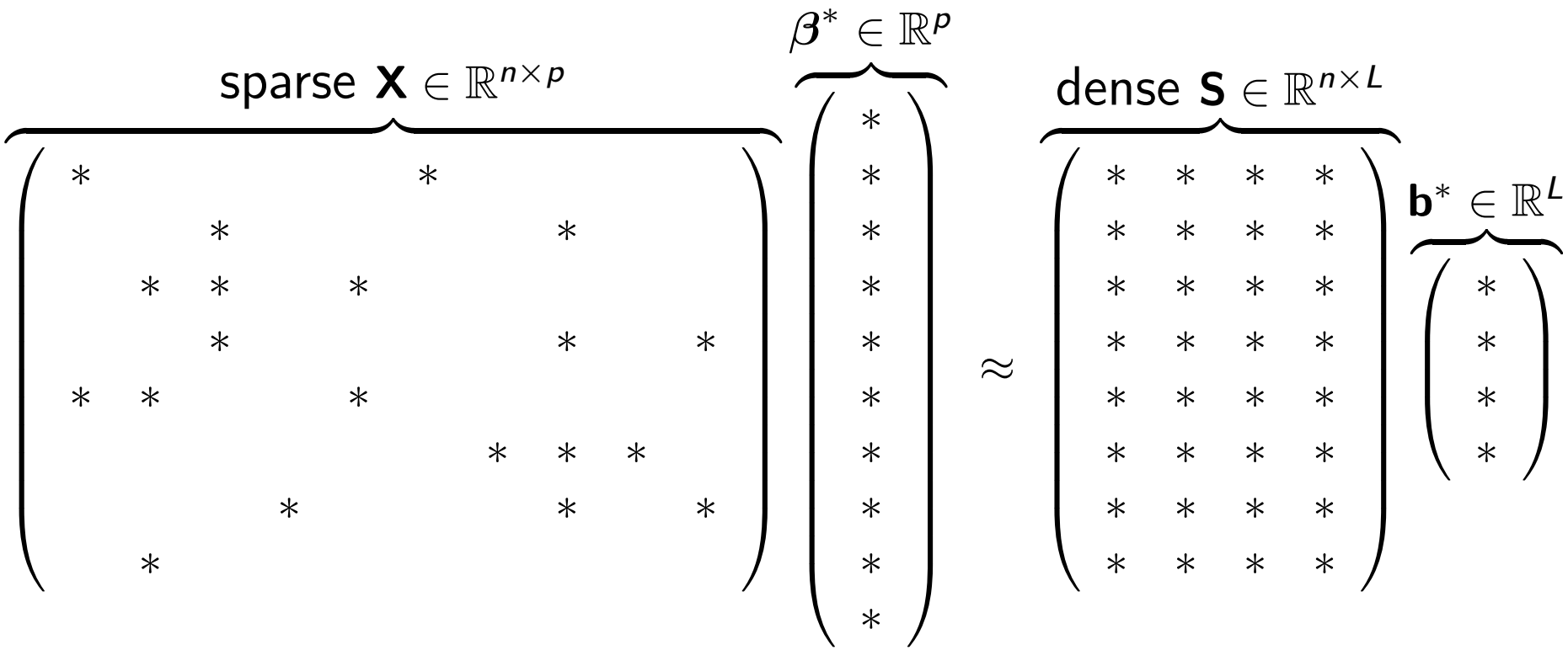
$$\begin{aligned} H_{il} &= \text{argmin}_{k \in \mathbf{z}_i} \pi_l(k) \\ S_{il} &= \Psi_{H_{il} l} X_{i H_{il}}, \end{aligned}$$

where Ψ_{hl} is the random sign of the h-th variable in the l-th permutation.

Approximation error

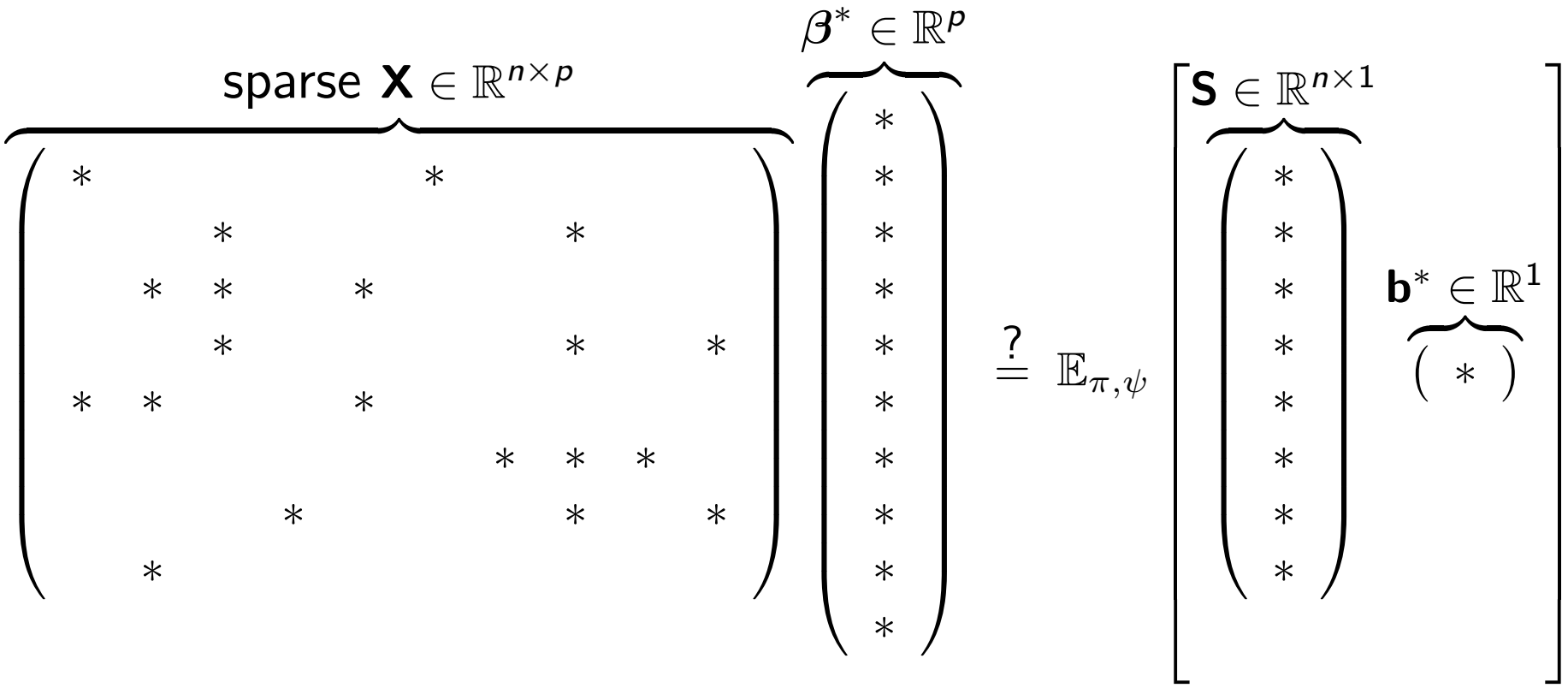
Can we find a $\mathbf{b}^* \in \mathbb{R}^L$ such that $\mathbf{X}\beta^*$ is close to $\mathbf{S}\mathbf{b}^*$ on average?

- Assume that there are $q \leq p$ non-zero entries in each row of $\mathbf{X}$.
- If not, can be dealt with.



Approximation error

Is there a $\mathbf{b}^*$ such that the expected value is unbiased (if averaged over the random permutations and sign assignments)?



Approximation error

Consider one permutation with min-hash value H_i for $i = 1, \ldots, n$ and random signs ψ_k, $k = 1, \ldots, p$.

$$\mathbb{E}_{\pi,\psi}\left[\overbrace{\begin{pmatrix} \psi_{H_1} X_{1H_1} \\ \psi_{H_2} X_{2H_2} \\ \ldots \\ \ldots \\ \ldots \end{pmatrix}}^{\mathbf{S}\in\mathbb{R}^{n\times 1}} \overbrace{\left(q \sum_{k=1}^{p} \beta_k^* \psi_k\right)}^{=:\mathbf{b}^*\in\mathbb{R}^1}\right] =$$

Approximation error

Can we find a $\mathbf{b}^* \in \mathbb{R}^L$ such that $\mathbf{X}\boldsymbol{\beta}^*$ is close to $\mathbf{S}\mathbf{b}^*$ on average?

$$\mathbb{E}_{\pi,\psi}\left[\underbrace{\begin{pmatrix} \psi_{H_1} X_{1H_1} \\ \psi_{H_2} X_{2H_2} \\ \cdots \\ \cdots \\ \cdots \end{pmatrix}}_{\mathbf{S}} \underbrace{\left(q \sum_{k=1}^{p} \beta_k^* \psi_k\right)}_{=:\mathbf{b}^*}\right] = \begin{pmatrix} \sum_{k=1}^{p} X_{1k} \beta_k^* q \mathbb{P}(H_1 = k) \\ \sum_{k=1}^{p} X_{2k} \beta_k^* q \mathbb{P}(H_2 = k) \\ \cdots \\ \cdots \\ \cdots \end{pmatrix}$$

Approximation error

Can we find a $\mathbf{b}^* \in \mathbb{R}^L$ such that $\mathbf{X}\boldsymbol{\beta}^*$ is close to $\mathbf{S}\mathbf{b}^*$ on average?

$$\mathbb{E}_{\pi,\psi}\left[\underbrace{\begin{pmatrix} \psi_{H_1} X_{1H_1} \\ \psi_{H_2} X_{2H_2} \\ \cdots \\ \cdots \\ \cdots \end{pmatrix}}_{\mathbf{S}} \underbrace{\left(q \sum_{k=1}^{p} \beta_k^* \psi_k\right)}_{=:\mathbf{b}^*}\right] = \begin{pmatrix} \sum_{k=1}^{p} X_{1k} \beta_k^* q \mathbb{P}(H_1 = k) \\ \sum_{k=1}^{p} X_{2k} \beta_k^* q \mathbb{P}(H_2 = k) \\ \cdots \\ \cdots \\ \cdots \end{pmatrix}$$

$$= \mathbf{X}\boldsymbol{\beta}^* \text{ (unbiased)}$$

Approximation error

Theorem

Let $\mathbf{b}^* \in \mathbb{R}^L$ *be defined by*

$$b_l^* = \frac{q}{L} \sum_{k=1}^{p} \beta_k^* \Psi_{kl} w_{\pi_l(k)},$$

where $\mathbf{w}$ *is a vector of weights. Then there is a choice of* $\mathbf{w}$*, such that:*

(i) *The approximation is unbiased:* $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{S}\mathbf{b}^*) = \mathbf{X}\boldsymbol{\beta}^*$.

(ii) $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{b}^*\|_2^2) \leq 2q\|\boldsymbol{\beta}^*\|_2^2/L$.

(iii) *If* $\|\|\mathbf{X}\|\|_\infty \leq 1$, *then* $\frac{1}{n}\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{S}\mathbf{b}^* - \mathbf{X}\boldsymbol{\beta}^*\|_2^2) \leq 2q\|\boldsymbol{\beta}^*\|_2^2/L$.

Linear model

- Assume model
$$\mathbf{Y} = \alpha^* \mathbf{1} + \mathbf{X}\beta^* + \boldsymbol{\varepsilon}.$$
- Random noise $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ satisfies $\mathbb{E}(\varepsilon_i) = 0$, $\mathbb{E}(\varepsilon_i^2) = \sigma^2$ and $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$ for $i \neq j$.
- Without loss of generality, assume $\mathbf{X}^T \mathbf{1} = \mathbf{0}$.

Linear model

- Assume model
$$\mathbf{Y} = \alpha^* \mathbf{1} + \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}.$$
- Random noise $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ satisfies $\mathbb{E}(\varepsilon_i) = 0$, $\mathbb{E}(\varepsilon_i^2) = \sigma^2$ and $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0$ for $i \neq j$.
- Without loss of generality, assume $\mathbf{X}^T \mathbf{1} = \mathbf{0}$.
- We will give bounds on a mean-squared prediction error (MSPE) of the form
$$\mathrm{MSPE}(\hat{\mathbf{b}}) := \mathbb{E}_{\boldsymbol{\varepsilon},\boldsymbol{\pi},\boldsymbol{\Psi}}\left(\|\mathbf{X}\boldsymbol{\beta}^* - \mathbf{S}\hat{\mathbf{b}}\|_2^2 \right)/n.$$

Ordinary least squares

Theorem

Let $\hat{\mathbf{b}}$ *be the least squares estimator and let* $L^* = \sqrt{2qn}\|\beta^*\|_2/\sigma$. *We have*

$$\text{MSPE}(\hat{\mathbf{b}}) \leq 2\max\left\{\frac{L}{L^*}, \frac{L^*}{L}\right\}\sigma\sqrt{\frac{2q}{n}}\|\beta^*\|_2 + \frac{\sigma^2}{n}.$$

Ordinary least squares

Theorem

Let $\hat{\mathbf{b}}$ *be the least squares estimator and let* $L^* = \sqrt{2qn}\|\beta^*\|_2/\sigma$. *We have*

$$\text{MSPE}(\hat{\mathbf{b}}) \leq 2\max\left\{\frac{L}{L^*}, \frac{L^*}{L}\right\}\sigma\sqrt{\frac{2q}{n}}\|\beta^*\|_2 + \frac{\sigma^2}{n}.$$

- If the size of the signal is fixed and columns of $\mathbf{X}$ are independent with roughly equal sparsity, then $\sqrt{q}\|\beta^*\|_2 \leq \text{const}\sqrt{p}$ and we have $\text{MSPE}(\hat{\mathbf{b}}) \to 0$ if $p/n \to 0$.
- If the signal $\mathbf{X}\beta^*$ is (partially) replicated in B groups of variables, then we only need $(p/B)/n \to 0$.

Ridge regression

Define

$$\hat{\mathbf{b}}^{\eta} := \arg\min_{\mathbf{b}} \|\mathbf{Y} - \bar{\mathbf{Y}}\mathbf{1} - \mathbf{S}\mathbf{b}\|_2^2 \text{ such that } \|\mathbf{b}\|_2^2 \leq (1+\eta)\frac{2q\|\beta^*\|_2^2}{L}.$$

Theorem

Let

$$\rho := \exp\left(-\frac{L\eta^2}{36q(36+\eta)}\right).$$

Then

$$\text{MSPE}(\hat{\mathbf{b}}^{\eta}) \leq \sqrt{2q}\|\beta^*\|_2\left(\frac{2\sigma\sqrt{1+\eta}+(L^*/L)}{\sqrt{n}}\right) + \frac{\sigma^2}{n} + \rho\frac{\|\mathbf{X}\beta^*\|_2^2}{n}.$$

- Similar result for logistic regression available.

Interaction models

Let $\mathbf{f}^* \in \mathbb{R}^n$ be given by

$$f_i^* = \sum_{k=1}^{p} X_{ik}\theta_k^{*,(1)} + \sum_{k,k_1=1}^{p} X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{k,k_1}^{*,(2)}, \quad i = 1, \ldots, n.$$

Interaction models

Let $\mathbf{f}^* \in \mathbb{R}^n$ be given by

$$f_i^* = \sum_{k=1}^{p} X_{ik}\theta_k^{*,(1)} + \sum_{k,k_1=1}^{p} X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{k,k_1}^{*,(2)}, \quad i = 1, \ldots, n.$$

Assume $\|\|\mathbf{X}\|\|_\infty \leq 1$. Previous results hold if $\|\boldsymbol{\beta}^*\|_2$ is replaced by

$$\ell(\boldsymbol{\Theta}^*) := \|\boldsymbol{\theta}^{*,(1)}\|_2 + 2\left(q\sum_{k,k_1,k_2}\left|\Theta_{kk_1}^{*,(2)}\Theta_{kk_2}^{*,(2)}\right|\right)^{1/2}.$$

Theorem

There exists $\mathbf{b}^* \in \mathbb{R}^L$ *such that*

(i) $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{S}\mathbf{b}^*) = \mathbf{f}^*$;

(ii) $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{S}\mathbf{b}^* - \mathbf{f}^*\|_2^2)/n \leq 2q\ell^2(\boldsymbol{\Theta}^*)/L$.

Interaction models

Let $\mathbf{f}^* \in \mathbb{R}^n$ be given by

$$f_i^* = \sum_{k=1}^{p} X_{ik}\theta_k^{*,(1)} + \sum_{k,k_1=1}^{p} X_{ik}\mathbb{1}_{\{X_{ik_1}=0\}}\Theta_{k,k_1}^{*,(2)}, \quad i = 1, \ldots, n.$$

Assume $\|\|\mathbf{X}\|\|_\infty \leq 1$. Previous results hold if $\|\boldsymbol{\beta}^*\|_2$ is replaced by

$$\ell(\boldsymbol{\Theta}^*) := \|\boldsymbol{\theta}^{*,(1)}\|_2 + 2\left(q\sum_{k,k_1,k_2}\left|\Theta_{kk_1}^{*,(2)}\Theta_{kk_2}^{*,(2)}\right|\right)^{1/2}.$$

Theorem

There exists $\mathbf{b}^* \in \mathbb{R}^L$ *such that*

(i) $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\mathbf{S}\mathbf{b}^*) = \mathbf{f}^*$;

(ii) $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Psi}}(\|\mathbf{S}\mathbf{b}^* - \mathbf{f}^*\|_2^2)/n \leq 2q\ell^2(\boldsymbol{\Theta}^*)/L$.

If there are a finite number of non-zero interaction terms with finite value, the approximation error becomes very small if $L \gg q^2$.

Prediction error

Assume the linear model from before, but with $\mathbf{X}\beta^*$ replaced by $\mathbf{f}^*$.

Theorem

Let $\hat{\mathbf{b}}$ be the least squares estimator and let $L^* = \sqrt{2qn}\,\ell(\mathbf{\Theta}^*)/\sigma$. We have

$$\text{MSPE}(\hat{\mathbf{b}}) \;\leq\; 2\max\left\{\frac{L}{L^*}, \frac{L^*}{L}\right\}\sigma\sqrt{\frac{2q}{n}}\ell(\mathbf{\Theta}^*) + \frac{\sigma^2}{n}.$$

- Consider a situation where there are a fixed number of interaction and main effects of fixed size, so $\ell(\mathbf{\Theta}^*) = O(\sqrt{q})$.

Prediction error

Assume the linear model from before, but with $\mathbf{X}\beta^*$ replaced by $\mathbf{f}^*$.

Theorem

Let $\hat{\mathbf{b}}$ be the least squares estimator and let $L^* = \sqrt{2qn}\,\ell(\mathbf{\Theta}^*)/\sigma$. We have

$$\text{MSPE}(\hat{\mathbf{b}}) \;\leq\; 2\max\left\{\frac{L}{L^*}, \frac{L^*}{L}\right\}\sigma\sqrt{\frac{2q}{n}}\ell(\mathbf{\Theta}^*) + \frac{\sigma^2}{n}.$$

- Consider a situation where there are a fixed number of interaction and main effects of fixed size, so $\ell(\mathbf{\Theta}^*) = O(\sqrt{q})$.
- If n, q and p increase by collecting new data and adding uninformative variables, then in order for the MSPE to vanish asymptotically, we require $q^2/n \to 0$.

Variable importance

Predicted values are

$$\hat{\mathbf{f}} = \mathbf{S}\hat{\mathbf{b}}$$

Let $\hat{\mathbf{f}}^{-(k)}$ be the predictions obtained when setting $\mathbf{X}_k = \mathbf{0}$. If the underlying model in linear and contains only main effects, $\hat{\mathbf{f}} - \hat{\mathbf{f}}^{-(k)} \approx \mathbf{X}_k \beta_k^*$.

Construct $\tilde{\mathbf{S}}$ in exactly the same way as $\mathbf{S}$ but using a matrix $\tilde{\mathbf{H}}$ rather than $\mathbf{H}$, with $\tilde{\mathbf{H}}$ defined by

$$\tilde{H}_{il} := \underset{k \in \mathbf{z}_i \setminus H_{il}}{\arg\min}\, \pi_l(k).$$

Variable importance

Predicted values are

$$\hat{\mathbf{f}} = \mathbf{S}\hat{\mathbf{b}}$$

Let $\hat{\mathbf{f}}^{-(k)}$ be the predictions obtained when setting $\mathbf{X}_k = \mathbf{0}$. If the underlying model in linear and contains only main effects, $\hat{\mathbf{f}} - \hat{\mathbf{f}}^{-(k)} \approx \mathbf{X}_k \beta_k^*$.

Construct $\tilde{\mathbf{S}}$ in exactly the same way as $\mathbf{S}$ but using a matrix $\tilde{\mathbf{H}}$ rather than $\mathbf{H}$, with $\tilde{\mathbf{H}}$ defined by

$$\tilde{H}_{il} := \underset{k \in \mathbf{z}_i \setminus H_{il}}{\arg\min}\, \pi_l(k).$$

Store $n \times L$ matrices $\mathbf{S}, \tilde{\mathbf{S}}$ and $\mathbf{H}$. Then

$$\hat{f}_i - \hat{f}_i^{(-k)} = \sum_{l=1}^{L} (S_{il} - \tilde{S}_{il}) \mathbb{1}_{\{H_{il}=k\}} \hat{b}_l.$$

Variable importance

Predicted values are

$$\hat{\mathbf{f}} = \mathbf{S}\hat{\mathbf{b}}$$

Let $\hat{\mathbf{f}}^{-(k)}$ be the predictions obtained when setting $\mathbf{X}_k = \mathbf{0}$. If the underlying model in linear and contains only main effects, $\hat{\mathbf{f}} - \hat{\mathbf{f}}^{-(k)} \approx \mathbf{X}_k \beta_k^*$.

Construct $\tilde{\mathbf{S}}$ in exactly the same way as $\mathbf{S}$ but using a matrix $\tilde{\mathbf{H}}$ rather than $\mathbf{H}$, with $\tilde{\mathbf{H}}$ defined by

$$\tilde{H}_{il} := \underset{k \in \mathbf{z}_i \setminus H_{il}}{\arg\min}\, \pi_l(k).$$

Store $n \times L$ matrices $\mathbf{S}, \tilde{\mathbf{S}}$ and $\mathbf{H}$. Then

$$\hat{f}_i - \hat{f}_i^{(-k)} = \sum_{l=1}^{L} (S_{il} - \tilde{S}_{il}) \mathbb{1}_{\{H_{il}=k\}} \hat{b}_l.$$

MRS aggregation

- The compressed design matrix $\mathbf{S}$ is generated in a random fashion.
- We can repeat the construction $B > 1$ times to obtain B different $\mathbf{S}$ matrices.

MRS aggregation

- The compressed design matrix $\mathbf{S}$ is generated in a random fashion.
- We can repeat the construction $B > 1$ times to obtain B different $\mathbf{S}$ matrices.
- In the spirit of bagging (Breiman, 1996) we can then aggregate the predictions obtained from the different random mappings by averaging them.

MRS aggregation

- The compressed design matrix $\mathbf{S}$ is generated in a random fashion.
- We can repeat the construction $B > 1$ times to obtain B different $\mathbf{S}$ matrices.
- In the spirit of bagging (Breiman, 1996) we can then aggregate the predictions obtained from the different random mappings by averaging them.
- Using $B > 1$ often gives large improvements and our experience has been that L can be chosen much lower than for $B = 1$ to achieve the same predictive accuracy.

Volatility prediction

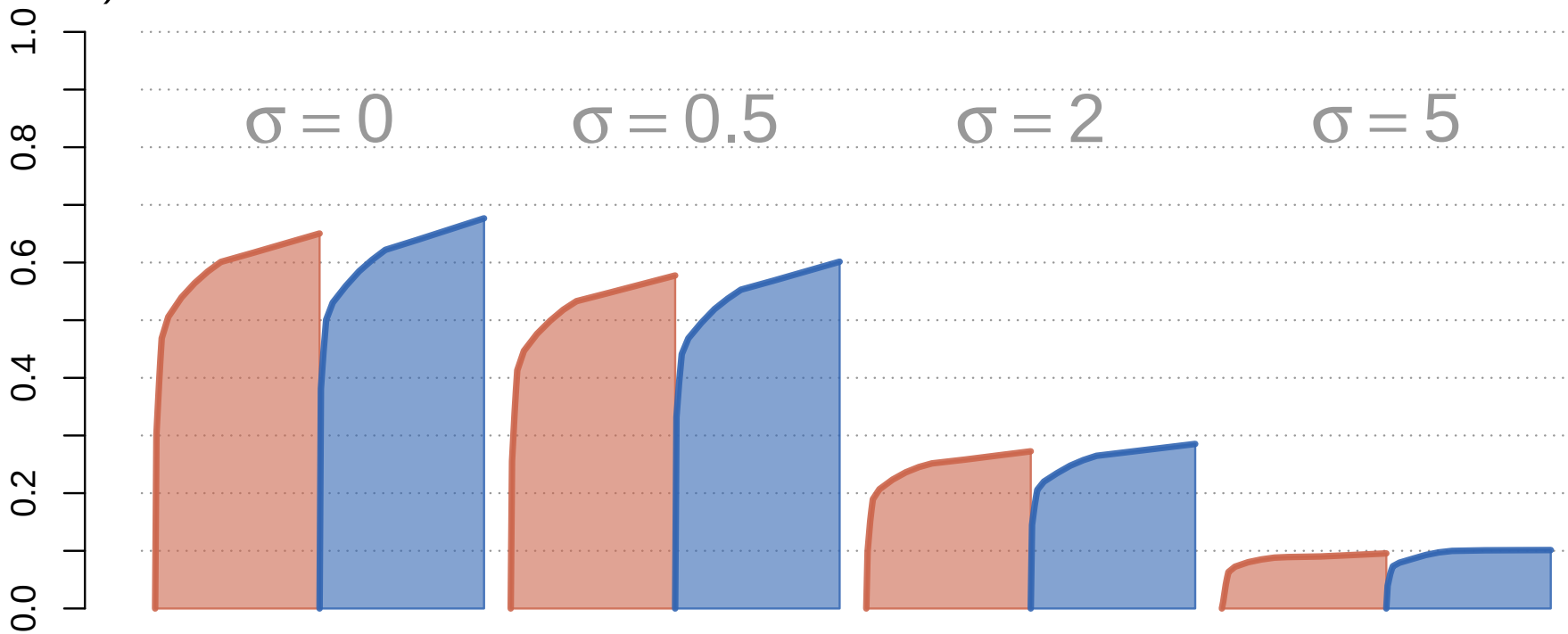
Forecast financial volatility of stocks based on 10-K report filings (Kogan, 2009).

Have $p = 4,272,227$ predictor variables for $n = 16,087$ observations.

Use various targets (volatility after release; a linear model; a non-linear model) and compare prediction accuracy with regression on random projections.

Volatility prediction

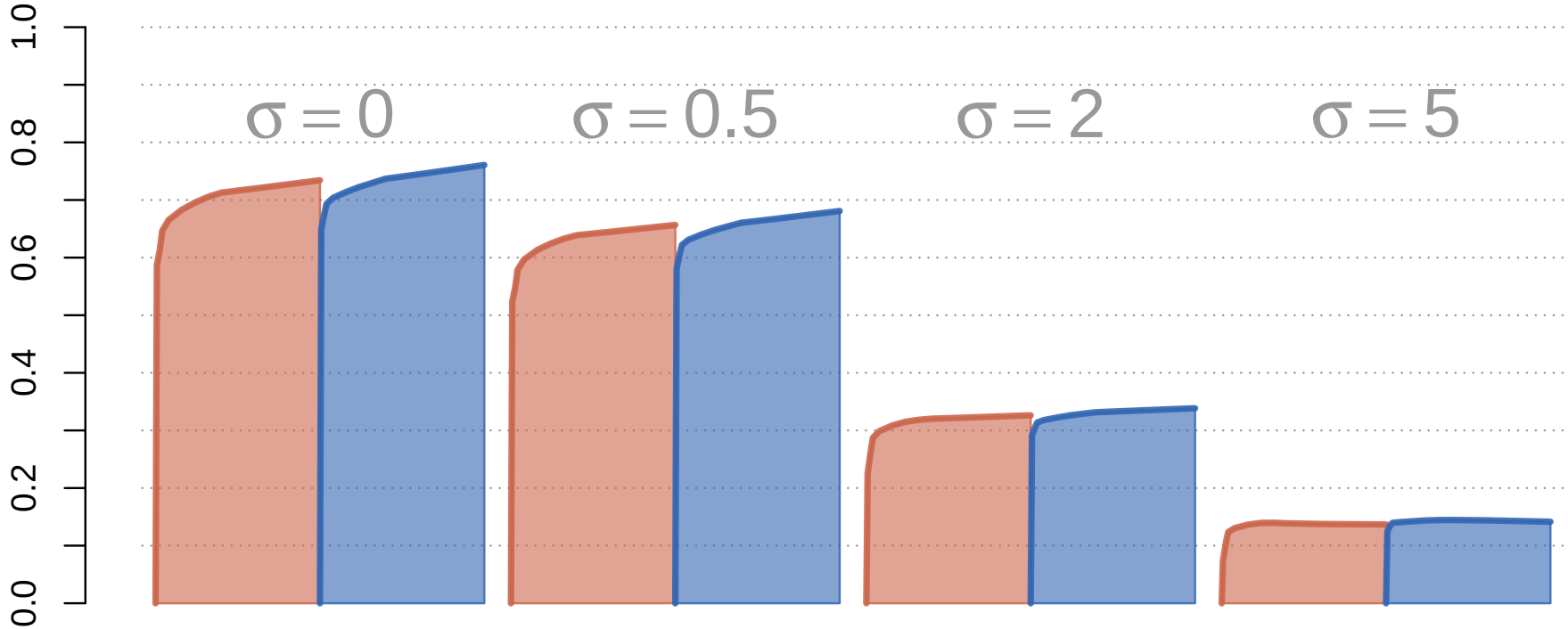
Correlation between prediction and response (volatility in year after release of text).



Red: MRS-mapping. Blue: random projections.

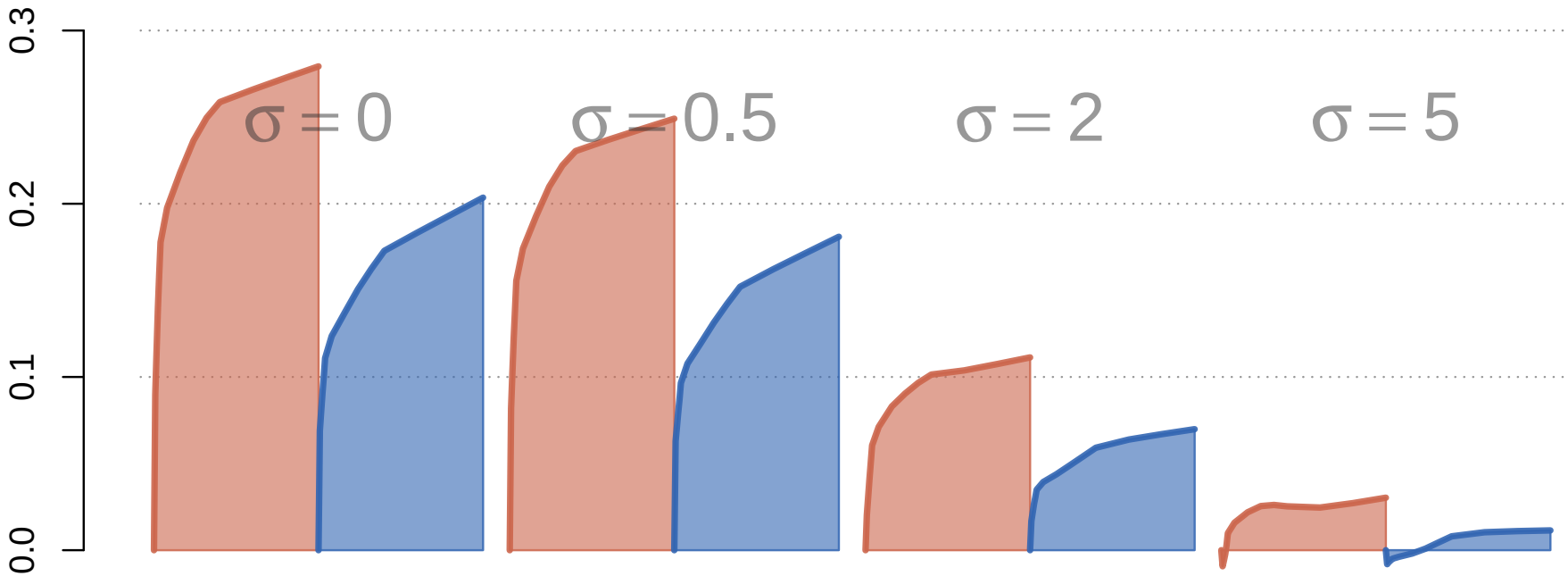
Volatility prediction

Response: linear model in original variables



Volatility prediction

Response: interaction model in original variables



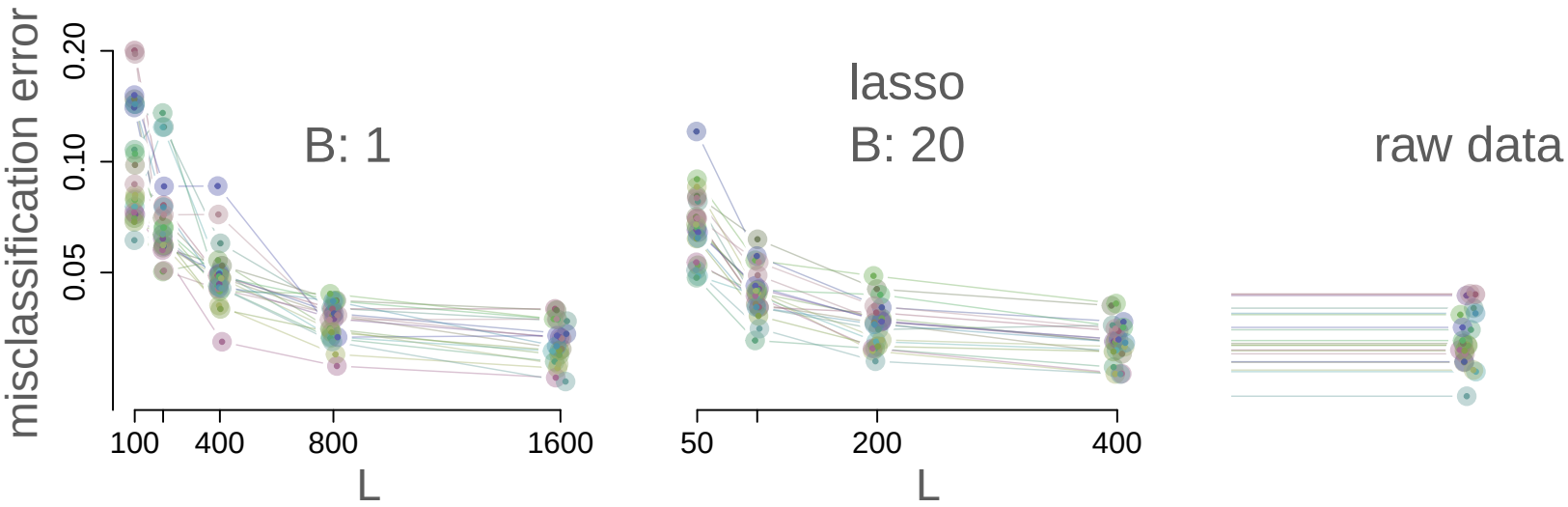
URL identification

Large-scale classification of malicious URLs with
$n \approx 2$ million and $p \approx 3$ million.
Data are ordered into consecutive days.

Response $\mathbf{Y} \in \{0,1\}^n$ is a binary vector where 1 corresponds to a malicious URL.

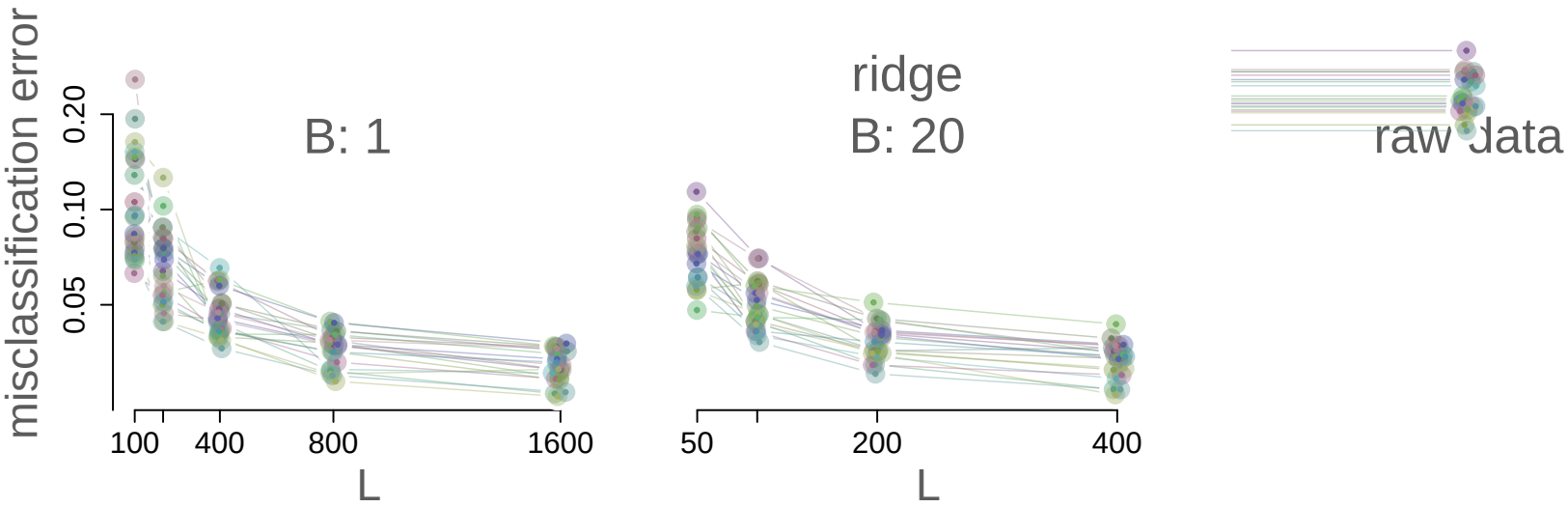
In order to compare MRS-mapping with the Lasso- and ridge-penalised logistic regression, we split the data into the separate days, training on the first half of each day and testing on the second. This gives on average $n \approx 20,000$, $p \approx 100,000$.

URL identification: Lasso regression



Lasso with and without MRS-mapping has similar performance here.

URL identification: Ridge regression



Ridge regression following MRS-mapping performs better than ridge regression applied to the original data.

Discussion

B-bit minwise hashing and closely related *MRS-mapping* are interesting dimensionality reduction techniques for large-scale sparse design matrices.

- Prediction error following compression can be bounded can be bounded with a slow rate (in the absence of assumptions on the design).
- Behaves similar to random projections (or ridge regression) if only main effects are present.
- Linear resgression using the compressed, dense, low-dimensional matrix can capture interactions among the large number of original sparse variables.