Minwise hashing for large-scale regression and classification with sparse data

Nicolai Meinshausen (Seminar für Statistik, ETH Zürich)
joint work with Rajen Shah (Statslab, University of Cambridge)

Simons Institute
18 September 2013

Large-scale sparse regression

Prediction problems with large-scale sparse predictors:

1. **Medical risk prediction/drug surveillance** (OMOP project). $n \approx 100,000$ patients with $p \approx 30,000$ indicator variables about medication history and symptoms.
   With interactions of second order, $p \approx 450$ million.
   With third order $p \approx 4.5$ trillion.
2. **Text data regression or classification**. Binary word indicator variables for approximately $p \approx 20,000$ words. Bi-grams and N-grams of higher order lead to hundreds of millions of variables.
3. **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.

Sparse linear model

Ignoring interactions (for now), can write regression model as:

$$
\overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\text{target } \mathbf{Y} \in \mathbb{R}^n}
\approx
\overbrace{\begin{pmatrix}
* & & & & & * & & & & \\
& & * & & & & & * & & \\
& * & * & & * & & & & & \\
& & * & & & & & * & & * \\
* & * & & & * & & & & & \\
& & & & & & * & * & * & \\
& & & * & & & & * & & * \\
& * & & & & & & & &
\end{pmatrix}}^{\text{sparse } \mathbf{X} \in \mathbb{R}^{n \times p}}
\overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\boldsymbol{\beta}^* \in \mathbb{R}^p}
+
\overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\text{noise } \boldsymbol{\varepsilon} \in \mathbb{R}^n}
$$

Non-zero entries are marked with $*$.

Classification model (logistic regression) analogous.

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

$$\overbrace{\begin{pmatrix} * & & & & & * & & & & \\ & & * & & & & & * & & \\ & * & * & & * & & & & & \\ & & * & & & & & * & & * \\ * & * & & & * & & & & & \\ & & & & & & * & * & * & \\ & & & * & & & & * & & * \\ & * & & & & & & & & \end{pmatrix}}^{\text{sparse } \mathbf{X} \in \mathbb{R}^{n \times p}} \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\boldsymbol{\beta}^* \in \mathbb{R}^p} \approx \overbrace{\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}}^{\text{dense } \mathbf{S} \in \mathbb{R}^{n \times L}} \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \end{pmatrix}}^{\mathbf{b}^* \in \mathbb{R}^L}$$

Here: dimensionality reduction with *b-bit minwise hashing* (Li and Koenig, 2011) and a closely related idea.

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 implemented by hash functions).
- Let the $n \times L$ matrix $\mathbf{M}$ be given by $M_{il} = \min \pi_l(\mathbf{z}_i)$.

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

Min-wise hashing matrix **M**

$$\begin{array}{cc} & \begin{array}{cccc} 3 & 1 & 2 & 4 \end{array} \\ \pi & \\ \mathbf{X} = & \begin{pmatrix} & * & & * \\ & & * & * \\ * & & * & \\ & * & * & \\ * & * & & \end{pmatrix} \end{array} \quad \Rightarrow \quad \mathbf{M} = \begin{pmatrix} 1 \\ 2 \\ 2 \\ 1 \\ 1 \end{pmatrix}$$

One column of **M** generated by the random permutation π of the variables.

Min-wise hashing matrix **M**

Can repeat L times to build **M** with repeated (pseudo-) random permutations π.

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

Work with **M** instead of sparse **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 $b = 1$,

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

Perform regression using binary $n \times L$ matrix $\mathbf{M}^{(1)}$ rather than $\mathbf{X}$.

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

When $L \ll p$ this gives large computational savings, and empirical studies report good performance (mostly for classification with SVM's).

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 and avoids choice of number of bits b to keep.
- 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**

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

MRS-mapping

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

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

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

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

H&M

Equivalent to storing **M**, we can store the "responsible" variables in **H**

$$
\begin{aligned}
M_{il} &= \min \pi_l(\mathbf{z}_i) \\
H_{il} &= \text{argmin}_{k \in \mathbf{z}_i} \pi_l(k)
\end{aligned}
$$

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

$$
\mathbf{X} = \begin{pmatrix} & \color{blue}{1} & & 1 \\ & & \color{red}{1} & 1 \\ 1 & & \color{red}{1} & \\ & \color{blue}{1} & \color{red}{1} & \\ 1 & \color{blue}{1} & & \end{pmatrix} \quad \Rightarrow \quad \mathbf{H} = \begin{pmatrix} \color{blue}{2} & 4 \\ \color{red}{3} & \color{red}{3} \\ \color{red}{3} & \color{red}{3} \\ \color{blue}{2} & \color{red}{3} \\ \color{blue}{2} & 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{S} = \begin{pmatrix} 1 & 1 \\ -1 & -1 \\ -1 & -1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix}
$$

Continuous variables

Can handle continuous variables

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

We get $n \times L$ matrices **H**, and **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.

$$\overbrace{\begin{pmatrix} * & & & & & * & & & & \\ & & * & & & & & * & & \\ & * & * & & * & & & & & \\ & & * & & & & & * & & * \\ * & * & & & * & & & & & \\ & & & & & & * & * & * & \\ & & & * & & & & * & & * \\ & * & & & & & & & & \end{pmatrix}}^{\text{sparse } \mathbf{X} \in \mathbb{R}^{n \times p}} \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\beta^* \in \mathbb{R}^p} \approx \overbrace{\begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}}^{\text{dense } \mathbf{S} \in \mathbb{R}^{n \times L}} \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \end{pmatrix}}^{\mathbf{b}^* \in \mathbb{R}^L}$$

Approximation error

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

$$\overbrace{\begin{pmatrix} * & & & & & * & & & & \\ & & * & & & & & * & & \\ & * & * & & * & & & & & \\ & & * & & & & & * & & * \\ * & * & & & * & & & & & \\ & & & & & & * & * & * & \\ & & & * & & & & * & & * \\ & * & & & & & & & & \end{pmatrix}}^{\text{sparse } \mathbf{X} \in \mathbb{R}^{n \times p}} \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\boldsymbol{\beta}^* \in \mathbb{R}^p} \stackrel{?}{=} \mathbb{E}_{\pi,\psi} \left[\overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\mathbf{S} \in \mathbb{R}^{n \times 1}} \overbrace{\begin{pmatrix} * \end{pmatrix}}^{\mathbf{b}^* \in \mathbb{R}^1} \right]$$

Approximation error

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

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

Approximation error

Can we find a $\mathbf{b}^* \in \mathbb{R}^L$ such that $\mathbf{X}\beta^*$ is close to $\mathbf{S}\mathbf{b}^*$ on average?
Example: binary $\mathbf{X}$ with one permutation with min-hash value H_i for $i = 1, \ldots, n$ and random signs ψ_k, $k = 1, \ldots, p$.

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

Can we find a $\mathbf{b}^* \in \mathbb{R}^L$ such that $\mathbf{X}\beta^*$ is close to $\mathbf{S}\mathbf{b}^*$ on average?
Example: binary $\mathbf{X}$ with one permutation with min-hash value H_i for $i = 1, \ldots, n$ and random signs ψ_k, $k = 1, \ldots, p$.

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

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

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

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:* $\mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\psi}}(\mathbf{S}\mathbf{b}^*) = \mathbf{X}\beta^*$.

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

Linear model

Assume model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta}^* + \boldsymbol{\varepsilon}.$$

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 $i \neq j$.

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*

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

- If the size of the signal is fixed and columns of $\mathbf{X}$ are independent with roughly equal sparsity, then $\sqrt{q}\|\beta^*\|_2 \leq \mathrm{const}\sqrt{p}$ and we have $\mathrm{MSPE}(\hat{\mathbf{b}}) \to 0$ 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

Can also estimate with ridge regression. Very similar results to OLS.

- The dimension L of the projection can be chosen arbitrarily large (from a statistical point of view).
- Ridge penalty parameter is then the relevant tuning parameter

Ridge regression

Can also estimate with ridge regression. Very similar results to OLS.

- The dimension L of the projection can be chosen arbitrarily large (from a statistical point of view).
- Ridge penalty parameter is then the relevant tuning parameter

Similar results for logistic regression available.

Interactions

Linear model:

$$\overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\text{target } \mathbf{Y} \in \mathbb{R}^n} \approx \overbrace{\begin{pmatrix} * & & & & & * & & & & \\ & & * & & & & & * & & \\ & * & * & & * & & & & & \\ & & * & & & & & * & & * \\ * & * & & & * & & & & & \\ & & & & & & * & * & * & \\ & & & * & & & & * & & * \\ & * & & & & & & & & \end{pmatrix}}^{\text{sparse } \mathbf{X} \in \mathbb{R}^{n \times p}} \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\boldsymbol{\beta}^* \in \mathbb{R}^p} + \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\text{noise } \boldsymbol{\varepsilon} \in \mathbb{R}^n}$$

Can we also fit pair-wise interactions if $p \geq 10^6$?

Interactions

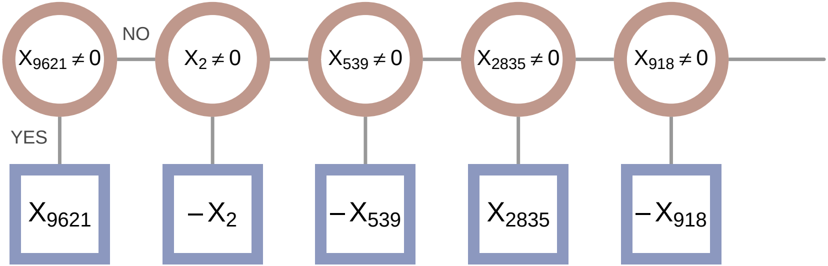
Linear model:

$$\underbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\text{target } \mathbf{Y} \in \mathbb{R}^n} \approx \overbrace{\begin{pmatrix} * & & & & & * & & & & \\ & & * & & & & & * & & \\ & * & * & & * & & & & & \\ & & * & & & & & * & & * \\ * & * & & & * & & & & & \\ & & & & & & * & * & * & \\ & & & * & & & & * & & * \\ & * & & & & & & & & \end{pmatrix}}^{\text{sparse } \mathbf{X} \in \mathbb{R}^{n\times p}} \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\boldsymbol{\beta}^* \in \mathbb{R}^p} + \overbrace{\begin{pmatrix} * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \\ * \end{pmatrix}}^{\text{noise } \boldsymbol{\varepsilon} \in \mathbb{R}^n}$$

Can we also fit pair-wise interactions if $p \geq 10^6$?

$\Rightarrow$ Min-wise hashing does it (almost) for free.

Minwise hash as a tree



Can view minwise hashing operation as a tree-type operation.

Interaction models

Let $\|\mathbf{X}\|_\infty \le 1$ and let $\mathbf{f}^* \in \mathbb{R}^n$ be given by

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

Theorem

Define

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

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

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

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

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 $\mathbf{f}^*$.
- Previous results hold if $\|\beta^*\|_2$ is replaced by $\ell(\mathbf{\Theta}^*)$.

For example:

Theorem

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

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

Advantages

Using MRS-maps for interaction fitting

- requires only fit of a linear model

Advantages

Using MRS-maps for interaction fitting

- requires only fit of a linear model
- does not require interactions to be created explicitly

Advantages

Using MRS-maps for interaction fitting

- requires only fit of a linear model
- does not require interactions to be created explicitly
- has a complexity saving factor of $(q/p)^2$ over the brute force approach.

Does require a larger number L of minwise hashing operations than fitting main effect models.

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 contains only main effects, $\hat{\mathbf{f}} - \hat{\mathbf{f}}^{-(k)} \approx \mathbf{X}_k \beta_k^*$.

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 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 use second-smallest instead of smallest active variable in the random permutation.

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 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 use second-smallest instead of smallest active variable in the random permutation.
Store $n \times L$ matrices $\mathbf{S}, \tilde{\mathbf{S}}$ and $\mathbf{H}$. Then

$$\hat{\mathbf{f}}^{-(k)} = \big(\mathbf{S} \circ \mathbb{1}_{\{\mathbf{H} \neq k\}} + \tilde{\mathbf{S}} \circ \mathbb{1}_{\{\mathbf{H} = k\}}\big)\hat{\mathbf{b}}.$$

Numerical results

Some observations from numerical simulations:

- Scheme becomes more competitive when repeating many times and aggregating.

Numerical results

Some observations from numerical simulations:

- Scheme becomes more competitive when repeating many times and aggregating.
- Predictive accuracy can decrease if we make L too large.
- In the absence of interactions: similar performance to ridge/random projections
- With interactions: performance between linear model (with ridge penalty or random projections) and Random Forest (Breiman, 01).

Volatility prediction

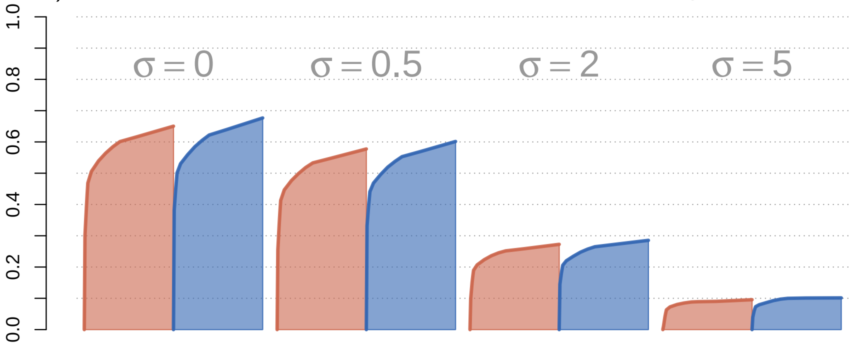
Forecast financial volatility of stocks based on 10-K report filings (Kogan, 2009).

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

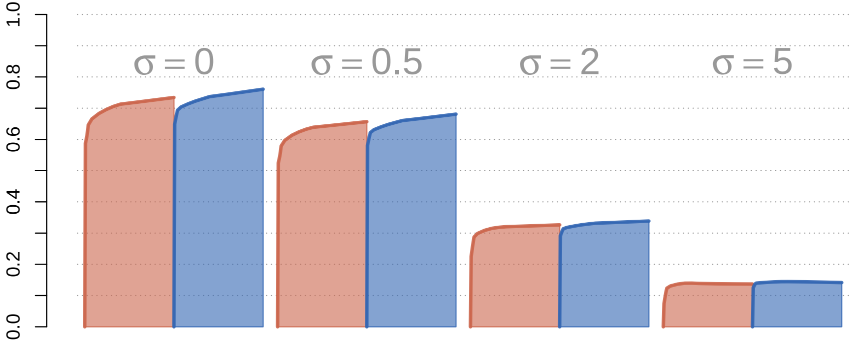
Correlation between prediction and response (volatility in year after release of text). Added additional noise with variance σ^2 to the reponse.



Red: MRS-mapping. Blue: random projections
(as functions of L up to 500)

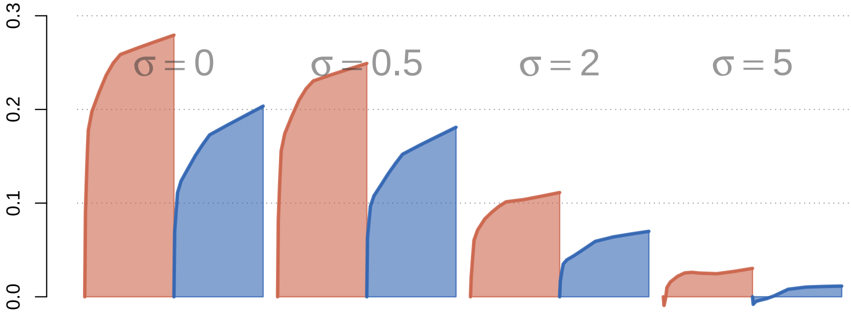
Volatility prediction

Response: linear model in original variables



Volatility prediction

Response: interaction model in original variables



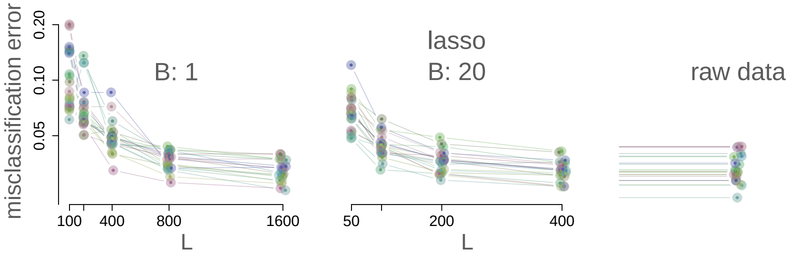
URL identification

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.

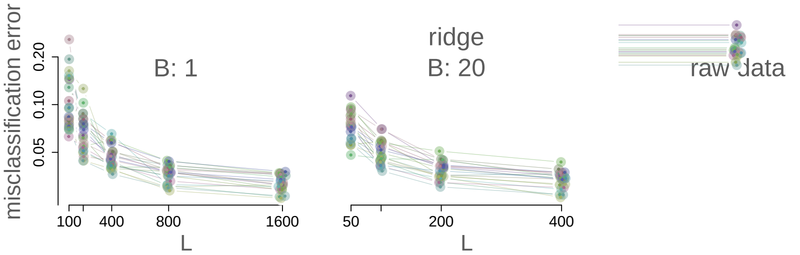
In order to compare MRS-mapping with the Lasso- and ridge-penalised logistic regression, we split the data into the separate days, training on the first half of each day and testing on the second. This gives on average $n \approx 20,000$, $p \approx 100,000$.

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-maps* interesting technique for dimensionality reduction for large-scale sparse design matrices.

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