A simple Bayesian algorithm for feature ranking in high dimensional regression problems

Enes Makalic and Daniel F. Schmidt

Centre for MEGA Epidemiology
The University of Melbourne

24th Australasian Joint Conference on Artificial Intelligence
2011

Outline

Outline

Problem Description (1)

- We have observed data (n samples, p predictors)
  - Targets $\mathbf{y} = (y_1, \ldots, y_n)'$, $y_i \in \mathbb{R}$
  - Matrix of p feature vectors $\mathbf{X} = (\mathbf{x}_1', \ldots, \mathbf{x}_n')'$, $\mathbf{x}_j \in \mathbb{R}^p$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$$

- We wish to explain the data using the features
  - Often p is very large!
  - Task: select best features to explain data

Problem Description (1)

- We have observed data (n samples, p predictors)
  - Targets $\mathbf{y} = (y_1, \ldots, y_n)'$, $y_i \in \mathbb{R}$
  - Matrix of p feature vectors $\mathbf{X} = (\mathbf{x}_1', \ldots, \mathbf{x}_n')'$, $\mathbf{x}_j \in \mathbb{R}^p$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$$

- We wish to explain the data using the features
  - Often p is very large!
  - Task: select best features to explain data

Problem Description (2)

- Concentrate on linear models

$$\mu_i = \sum_{j=1}^{p} \beta_j x_{i,j}$$

- μ_i is the linear predictor
  - Regression, $y_i \in \mathbb{R}$

$$y_i = \mu_i + \varepsilon_i$$

    with ε_i a random disturbance
  - Classification, $y_i \in \{0, 1\}$

$$\mathbb{P}\{y_i\} = \eta(\mu_i)$$

    where $\eta : \mathbb{R} \to (0, 1)$
- Determine which features to include

Some Existing Approaches

- Marginal feature selection
  - Rank features by marginal increase in utility
  - Example, Hall & Miller (2009) (combined with bootstrap)

  $\Rightarrow$ ignores joint information in features

- Penalized regression
  - Produce a "path" of feature subsets
  - Example, LASSO (ℓ_1 penalisation)

  $\Rightarrow$ does not take into account uncertainty in estimates

- Our method designed to overcome both drawbacks

Outline

Algorithm (1)

- Focus on regression
- Assume:
  - Covariates are standardised

$$\sum_{i=1}^{n} x_{i,j} = 0, \quad \sum_{i=1}^{n} x_{i,j}^2 = 1$$

  - There exists $B > 0$ samples

$$\{\beta_1, \ldots, \beta_B\}$$

from posterior $p(\beta|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\beta, \mathbf{X})\pi(\beta)$

Algorithm (1)

- Focus on regression
- Assume:
  - Covariates are standardised

$$\sum_{i=1}^{n} x_{i,j} = 0, \quad \sum_{i=1}^{n} x_{i,j}^2 = 1$$

  - There exists $B > 0$ samples

$$\{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_B\}$$

from posterior $p(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\beta}, \mathbf{X})\pi(\boldsymbol{\beta})$

Algorithm (2)

- Let $\mathbf{R} \in \mathbb{N}^{B \times p}$ be matrix of rankings
  - $R_{k,j}$ is the ranking of j-th feature in k-th sample
- For each sample β_k, $k = 1, \ldots, B$
  - Rank each feature in descending order of $|\beta_{k,j}|$
  - Example:

β_k	2.3	-6.3	-3.5	0.2	-0.8	3.1	1.6	-2.1
Ranking	4	1	2	8	7	3	6	5

- Percentiles of $\mathbf{R}$ yield confidence intervals on feature ranks
- See paper for details

Algorithm (2)

- Let $\mathbf{R} \in \mathbb{N}^{B \times p}$ be matrix of rankings
  - $R_{k,j}$ is the ranking of j-th feature in k-th sample
- For each sample $\boldsymbol{\beta}_k$, $k = 1, \ldots, B$
  - Rank each feature in descending order of $|\beta_{k,j}|$
  - Example:

$\boldsymbol{\beta}_k$	2.3	-6.3	-3.5	0.2	-0.8	3.1	1.6	-2.1
Ranking	4	1	2	8	7	3	6	5

- Percentiles of $\mathbf{R}$ yield confidence intervals on feature ranks
- See paper for details

Algorithm (2)

- Let $\mathbf{R} \in \mathbb{N}^{B \times p}$ be matrix of rankings
  - $R_{k,j}$ is the ranking of j-th feature in k-th sample
- For each sample $\boldsymbol{\beta}_k$, $k = 1, \ldots, B$
  - Rank each feature in descending order of $|\beta_{k,j}|$
  - Example:

$\boldsymbol{\beta}_k$	2.3	-6.3	-3.5	0.2	-0.8	3.1	1.6	-2.1
Ranking	4	1	2	8	7	3	6	5

- Percentiles of $\mathbf{R}$ yield confidence intervals on feature ranks
- See paper for details

Algorithm (3)

- Motivation:
  - In a linear model with parameters $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$

$$\beta_j^2 \left(\sum_{i=1}^{n} x_{i,j}^2 \right)$$

    is the variance explained by feature j
- We standardise features to unit length
  $\Rightarrow$ Variance explained reduces to β_j^2
- Ranking by $|\beta_j|$ is equivalent to ranking by explained variance
- Using samples from posterior incorporates uncertainty of estimates into ranking

Experiments (1)

- Compared our method (BFR) against:
  - Hall & Miller marginal algorithm (HM)
  - Breiman's random forests (RF)
- Synthetic data: we know which features are "true"
- Use `TopX` metric
  - The rank below which all true features are included
- For example, `TopX` of 15 indicates all true features are included in first 15 features, as determined by their ranking

Experiments (2)

- Three test functions; $p = 100$
  - Function I: 6 non-noise features, uncorrelated features
  - Function II: 4 non-noise features, correlated features
  - Function III: 5 non-noise features, correlated features
- Two levels of signal-to-noise ratio (SNR) (1 and 8)
- For each function and SNR level
  - Generate 100 data sets of size $n = 50$
  - Run each algorithm and obtain `TopX` score
  - Box-plots of `TopX` scores for each algorithm

Experiments (3)

- Bayesian ridge regression
- Hierarchy:

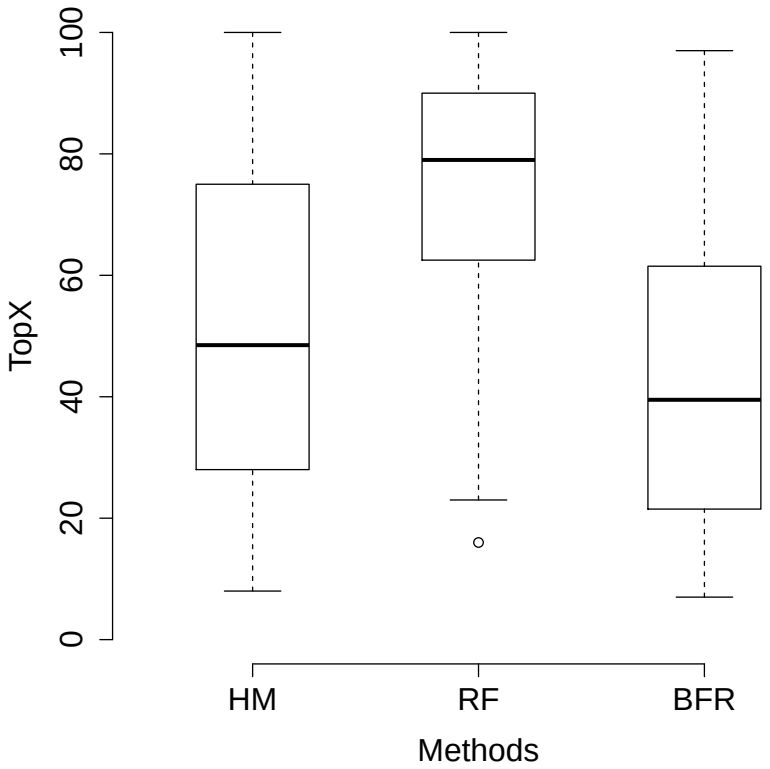
$$
\begin{aligned}
\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}, \sigma^2 &\sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n) \\
\boldsymbol{\beta} &\sim N_p(\mathbf{0}_p, \sigma^2/\lambda^2\mathbf{I}_p) \\
\sigma^2 &\sim \sigma^{-2}d\sigma^2 \\
\lambda &\sim \text{Gamma}(1, 0.01)
\end{aligned}
$$

where

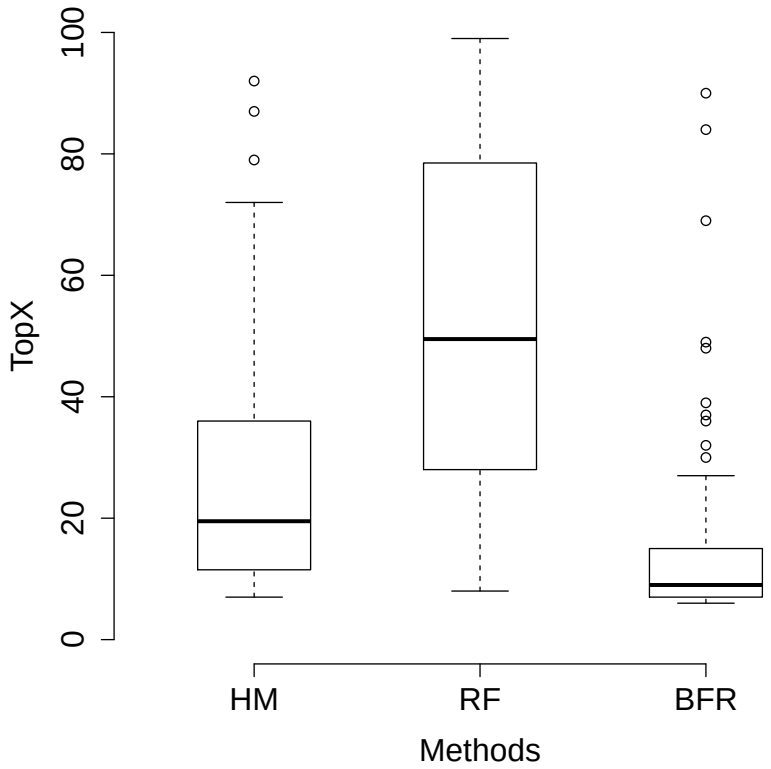
- $\boldsymbol{\beta}$ are the regression coefficients
- σ^2 is the noise variance
- λ is the regularisation parameter

Results (1)

Function I:



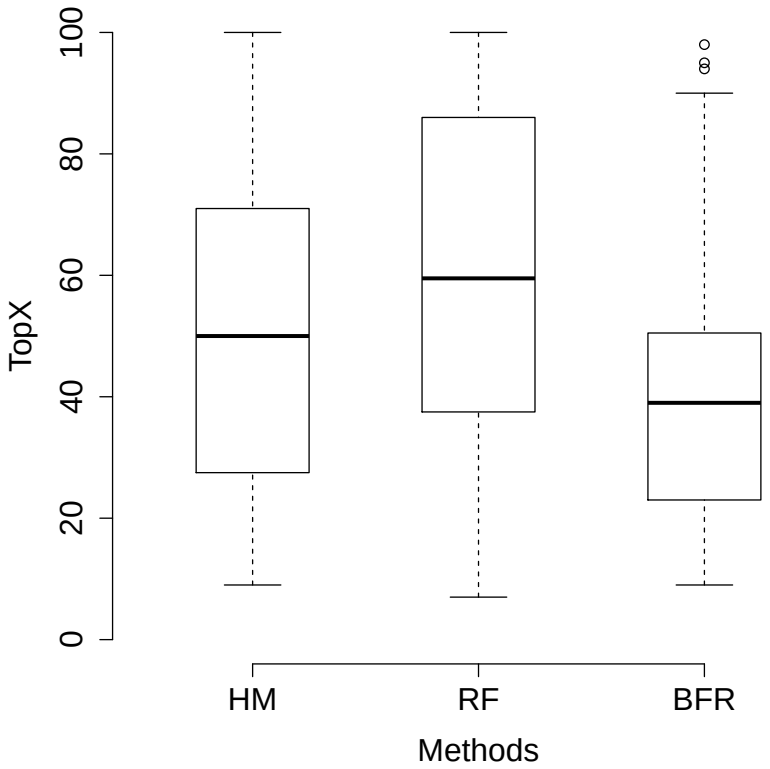
(a) SNR = 1



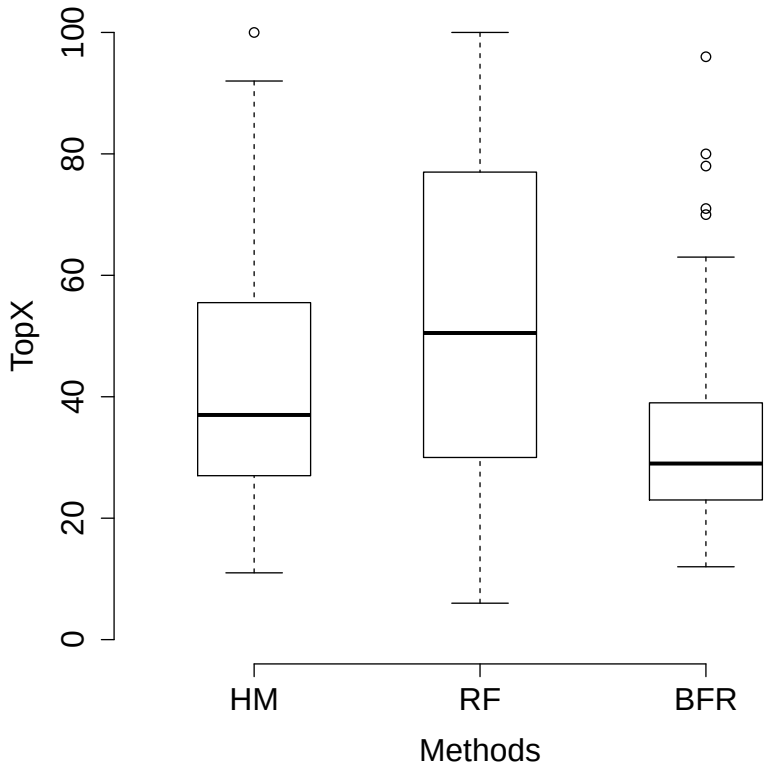
(b) SNR = 8

Results (2)

Function II:



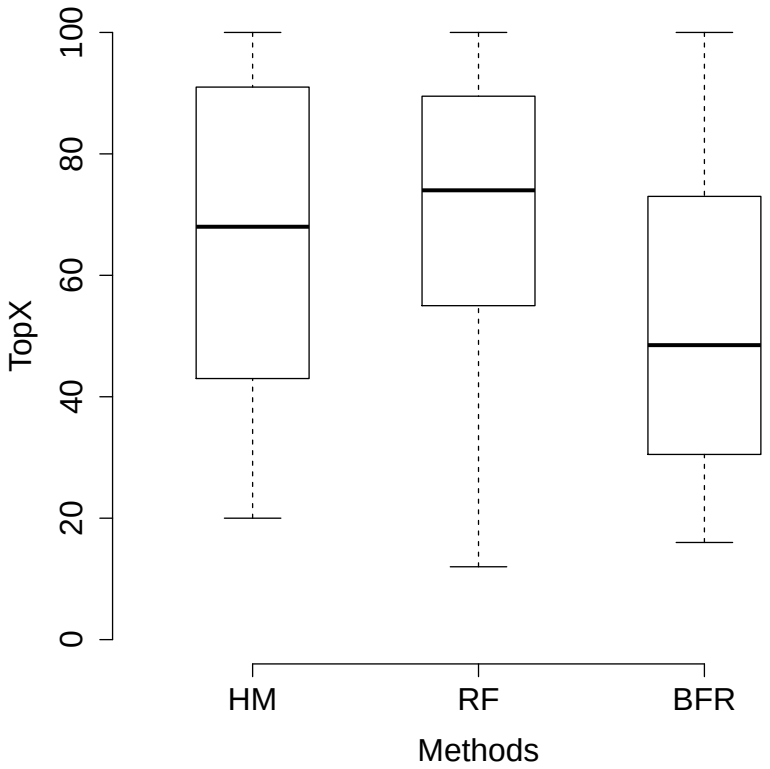
(c) SNR = 1



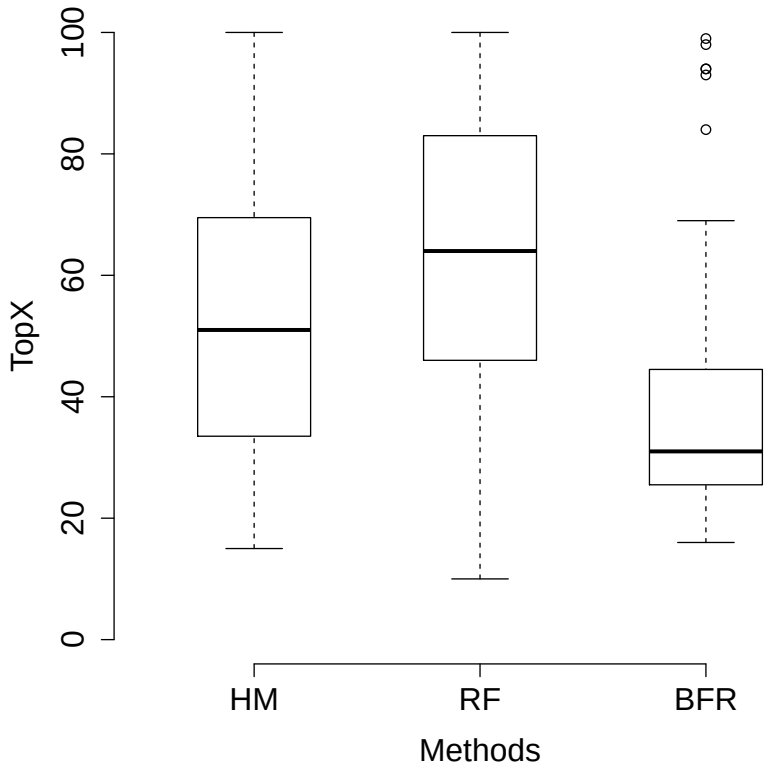
(d) SNR = 8

Results (3)

Function III:



(e) SNR = 1



(f) SNR = 8

Results (4)

- Our algorithm worked well, especially with correlated features
  - Smallest median `TopX`
  - Generally lower variance of `TopX`
- Real data experiments:
  - Diabetes data set $(n = 442, p = 10)$
  - Crime & Communities data set $(n = 319, p = 123)$
- Details in paper

Future Work

- Other regularisation schemes
- Experiment on classification problems
- Extend to non-linear transformations of features
  - More powerful test of feature association

- Thank you – questions?

References

- Breiman, L. "Random forests". *Machine Learning*, 2001, Vol. 45, pp. 5–32
- Hall, P., Miller, H. "Using generalized correlation to effect variable selection in very high dimensional problems". *Journal of Computational and Graphical Statistics*, 2009, Vol. 18, pp. 533–550
- Fan, J., Samworth, R., Wu, Y. "Ultrahigh dimensional feature selection: Beyond the linear model". *Journal of Machine Learning Research*, 2009, Vol. 10, pp. 2013–2038
- Park, T., Casella, G. "The Bayesian lasso". *Journal of the American Statistical Association*, 2008, Vol. 103, pp. 681–686