An Adaptive Forward/Backward Greedy Algorithm for Learning Sparse Representations

Tong Zhang

Statistics Department
Rutgers University, NJ

Learning with large number of features

- Consider learning problems with large number of features
- Sparse target
  - linear combination of small number of features
- This talk: how to solve sparse learning problem
  - directly solve L_0 regularization: approximate path following
  - provably effective under appropriate conditions

Notations

- Basis functions $\mathbf{f}_1, \ldots, \mathbf{f}_d \in R^n$; Observation $\mathbf{y} \in R^n$
- $d \gg n$
- Cost function $R(\cdot)$:
  - e.g., least squares problem: $R(\mathbf{f}) = \|\mathbf{f} - \mathbf{y}\|_2^2/n$
- Given $\mathbf{w} \in R^d$, linear prediction function $f(\mathbf{w}) = \sum_j \mathbf{w}_j \mathbf{f}_j$
- Empirical risk minimization:

$$R(f(\mathbf{w})).$$

Sparse Regularization

- $d \gg n$: ill-posed
  - what if only a few relevant features.
- Learning method: L_0 regularization

$$\hat{\mathbf{w}}_{FS} = \arg\min_{\mathbf{w}} R(f(\mathbf{w})), \quad \text{subject to } \|w\|_0 \leq k.$$

$$\|w\|_0 = |\{j : w_j \neq 0\}|$$

- Combinatorial problem: find $k \ll n$ features with smallest prediction error.
  - C_d^k possible feature combinations: exponential in k (NP-hard).
- This talk: how to solve L_0 using greedy algorithm.

Statistical model for sparse least squares regression

- Linear prediction model: $\boldsymbol{Y = \sum_j \bar{w}_j f_j + \epsilon}$
  - $\epsilon \in R^n$ are n independent zero-mean noise with variance $\leq \sigma^2$.
- Assumption: sparse model achieves good performance
  - $\bar{w}$ has only k nonzero components: $k \ll n \ll d$.
  - or approximately sparse: $\bar{w}$ can be approximated by sparse vector.
- Compressed sensing is special case: noise $\sigma = 0$ with least squares loss.

Efficient Sparse Learning and Feature Selection Methods

- Traditional Methods:
  - convex relaxation: L_1-regularization.
  - simple greedy algorithms:
    * forward (greedy) feature selection: boosting.
    * backward (greedy) feature selection.
  - provably effective only under restrictive assumptions.

- A new method: adaptive forward/backward greedy algorithm: FoBa
  - solve L_0 directly: remedy problems in traditional methods.
  - theoretically: better statistical behavior under less restrictive assumptions.

Some Assumptions

- sub-Gaussian noise: σ is noise level
- basis are normalized: $\|\mathbf{f}_j\|_2 = 1$ ($j = 1, \ldots, d$)
- sparse-eigenvalue conditions: any small number of basis functions are linearly independent for small k ($f(\mathbf{w}) = \sum_j \mathbf{w}_j \mathbf{f}_j$)

$$\rho(k) = \inf \left\{ \frac{1}{n} \|f(\mathbf{w})\|_2^2 / \|\mathbf{w}\|_2^2 : \|\mathbf{w}\|_0 \leq k \right\} > 0,$$

and for all $\bar{F} \subset \{1, \ldots, d\}$, let

$$\lambda(\bar{F}) = \sup \left\{ \frac{1}{n} \|f(\mathbf{w})\|_2^2 / \|\mathbf{w}\|_2^2 : \mathrm{support}(\mathbf{w}) \subset \bar{F} \right\}.$$

L_1-regularization and its Problems

- Closest convex relaxation of L_0-regularization (feature selection):

$$\hat{w}_{L_1} = \arg\min_{w} R(\mathbf{w}), \quad \text{subject to } \|\mathbf{w}\|_1 \leq k.$$

  replace L_0-regularization $\|w\|_0 \leq k$.

- Practical: not good approximation to L_0 regularization

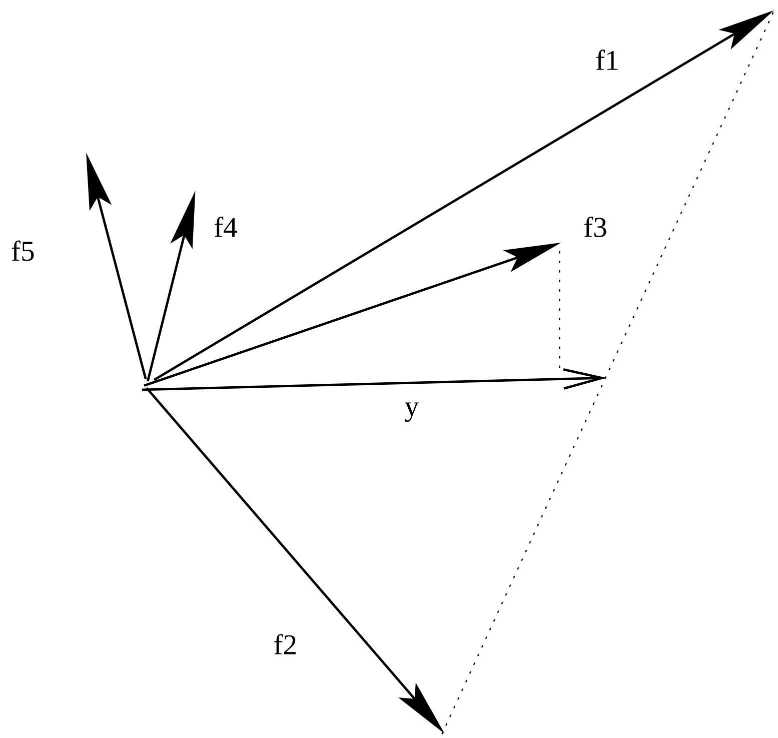
- Theoretical: analysis exists
  - requires relatively strong conditions
  - inferior sparse learning method when noise is present: bias

Forward Greedy Algorithm

- Initialize feature set $F^k = \emptyset$ at $k = 0$
- Iterate
  - find best feature j to add to F^k with most significant cost reduction
  - $k++$ and $F^k = F^{k-1} \cup \{j\}$

Problem of Forward Greedy Feature Selection

- Can make error in early stage that cannot be corrected.
  - correct basis functions: f_1 and f_2, but f_3 closer to y
  - forward greedy algorithm output: $f_3, f_1, f_2, \ldots$



Backward Greedy Algorithm

- Initialize feature set $F^k = \{1, \ldots, d\}$ at $k = d$
- Iterate
  - find best feature $j \in F^k$ to remove with least significant cost increase
  - $F^{k-1} = F^k - \{j\}$ and $k - -$

Problems of Backward Greedy Feature Selection

- Computationally very expensive.
- The naive version overfits the data when $d \gg n$: $R(F^d) = 0$.
  - fails if $R(F^d - \{j\}) = 0$ for all $j \in F_t$.
  - cannot effectively eliminate bad features
- Works only when $n \gg d$ (insignificant overfitting).
  - when $n \ll d$: have to regularize the naive version to prevent overfitting
  - how to regularize?

Idea: Combine Forward/Backward Algorithms

- Forward greedy
  - pros: computationally efficient; doesn't overfit
  - cons: error made in early stage doesn't get corrected later
- Backward greedy
  - pros: can correct error by looking at the full model
  - cons: need to start with sparse/non-overfited model
- Combination: adaptive forward/backward greedy
  - computationally efficient; doesn't overfit; error made in early stage can be corrected by backward greedy step later
  - key design issue: when to take a backward step?

Greedy method for Direct L_0 minimization

- Optimize objective function greedily:

$$\min_{w}[R(\mathbf{w}) + \lambda\|\mathbf{w}\|_0].$$

- Two types of greedy operations to reduce L_0 regularized objective
  - feature addition (forward): $R(\mathbf{w})$ decreases, $\lambda\|\mathbf{w}\|_0$ increases by λ
  - feature deletion (backward): $R(\mathbf{w})$ increases, $\lambda\|\mathbf{w}\|_0$ decreases by λ

- First idea: alternating with addition/deletion to reduce objective
  - "local" solution: a fixed point of the procedure
  - problem: ineffective deletion with small λ: overfitting like backward greedy

- Key modification: track a sparse solution path
  - L_0 path-following: λ decreases from ∞ to 0.

FoBa (conservative): Adaptive Forward/Backward Greedy Algorithm

- Iterate
  - forward step
    - find best feature j to add
    - $k++$ and $F^k = F^{k-1} \cup \{j\}$
    - $\delta_k =$ forward step square error reduction
    - if ($\delta_k < \epsilon$) terminate the loop.
  - backward step
    - find best feature $j \in F^k$ to remove
    - if (backward square error increase $\leq 0.5\delta_k$)
      - $F_{k-1} = F_k - \{j\}$ and $k--$
      - repeat the backward step.

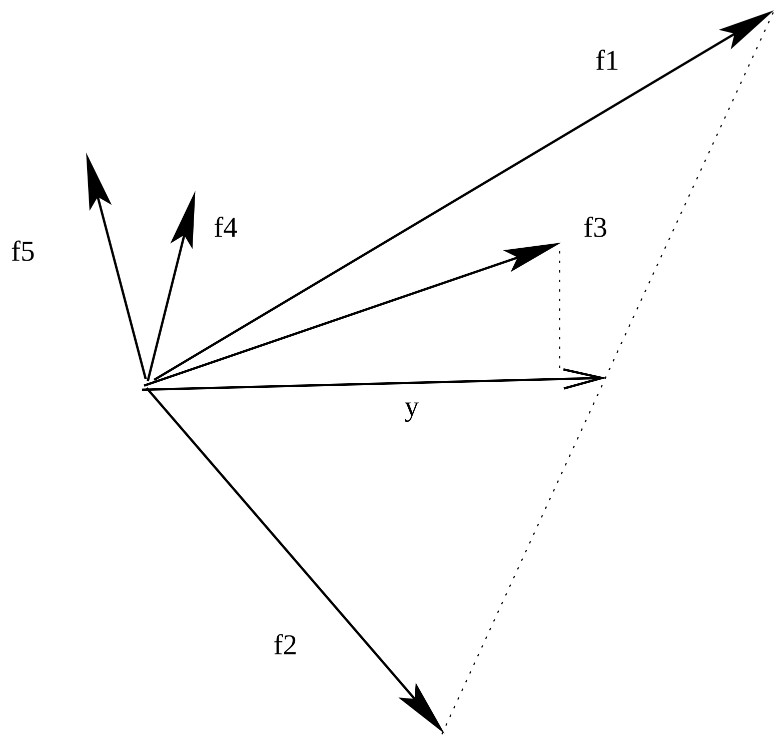
- L_0 path-following: replace 0.5 by a shrinkage factor $\nu \to 1$

Computational Efficiency

- Assume $R(\mathbf{w}) \geq 0$ for all $\mathbf{w} \in R^d$
- Given stopping criterion $\epsilon > 0$
  - ϵ: should be set to noise level
- FoBa terminates after at most $2R(0)/\epsilon$ forward iterations.
- The algorithm approximately follows an L_0 local solution path
  - statistically as effective as global L_0 under appropriate conditions.

Forward Greedy Failure Example Revisited

- FoBa can correct errors made in early forward stages
  - correct basis functions: f_1 and f_2, but f_3 is closer to y
  - FoBa output: $f_3, f_1, f_2, -f_3 \ldots$



Learning Theory: FoBa with Sparse Target

Theorem 1. *Assume also that the target is sparse: there exists* $\bar{\mathbf{w}} \in R^d$ *such that* $\bar{\mathbf{w}}^T \mathbf{x}_i = \mathbf{E} y_i$ *for* $i = 1, \ldots, n$, *and* $\bar{F} = \mathrm{support}(\bar{\mathbf{w}})$. *Let* $\bar{k} = |\bar{F}|$, *and assume that for some* $s > 0$, *we have* $\bar{k} \leq 5s\rho(s)^2(32+5\rho(s)^2)^{-1}$. *Given any* $\eta \in (0, 1/3)$, *and choose* ϵ *that satisfies the condition* $\epsilon \geq 64\rho(s)^{-2}\sigma^2 \ln(2d/\eta)/n$. *If* $\min_{j \in \mathrm{support}(\bar{\mathbf{w}})} |\bar{\mathbf{w}}_j|^2 \geq \frac{64}{25}\rho(s)^{-2}\epsilon$, *then with probability larger than* $1 - 3\eta$:

- *When the algorithm terminates, we have* $F^k = \mathrm{support}(\bar{\mathbf{w}})$, *and the solution*

$$\|\mathbf{w}^k - \bar{\mathbf{w}}\|_2 \leq \sigma \sqrt{\bar{k}/(n\rho(\bar{k}))} \left[1 + \sqrt{20 \ln(1/\eta)}\right].$$

- *The algorithm terminates after at most* $\frac{7\lambda(\bar{F})\|\bar{\mathbf{w}}\|_2^2}{\rho(s)^2 \min_{j \in \bar{F}} |\bar{\mathbf{w}}_j|^2}$ *forward-backward iterations.*

Approximate Sparse Target for FoBa

- Let $\epsilon \geq 64\rho(s)^{-2}\sigma^2 \ln(2d/\eta)/n$.

- $\bar{k} = |\bar{F}|$: $\bar{F} = \mathrm{support}(\bar{\mathbf{w}})$
  - $\bar{\mathbf{w}}$: approximate target parameter

- $k(\epsilon) = \left|\{j \in \bar{F} : |\bar{\mathbf{w}}_j|^2 \leq 12\epsilon/\rho(s)^2\}\right|$
  - $k(\epsilon)$ can be much smaller than $\bar{k}$
  - features with small weights that cannot be reliably selected by any algorithm (up to a constant in threshold)

- Learning Theory Bounds
  - Optimal feature selection and parameter estimation accuracy

– Feature selection:

$$\max(|\bar{F} - F^{(k)}|, |F^{(k)} - \bar{F}|) = O(k(\epsilon) + \|\mathbf{E}\mathbf{y} - f(\bar{\mathbf{w}})\|_2/(n\epsilon))$$

– Estimation error bound of $\|\mathbf{w}^{(k)} - \bar{\mathbf{w}}\|_2$: (better than L_1)

$$O\left(\underbrace{\sigma\sqrt{\frac{\bar{k}\ln(1/\eta)}{n}}}_{O(\text{parametric})} + \underbrace{\sigma\sqrt{k(\epsilon)\ln(d/\eta)/n}}_{\sqrt{k(\epsilon)\epsilon}} + \underbrace{\|\mathbf{E}\mathbf{y} - f(\bar{\mathbf{w}})\|_2/n}_{\text{approximation error}}\right).$$

– Compare to L_1: needs stronger condition for feature selection, and gives error

$$O\left(\sigma\sqrt{\bar{k}\ln(d/\eta)/n} + \underbrace{\|\mathbf{E}\mathbf{y} - f(\bar{\mathbf{w}})\|_2/n}_{\text{approximation error}}\right).$$

Artificial data experiment: feature selection/parameter estimation

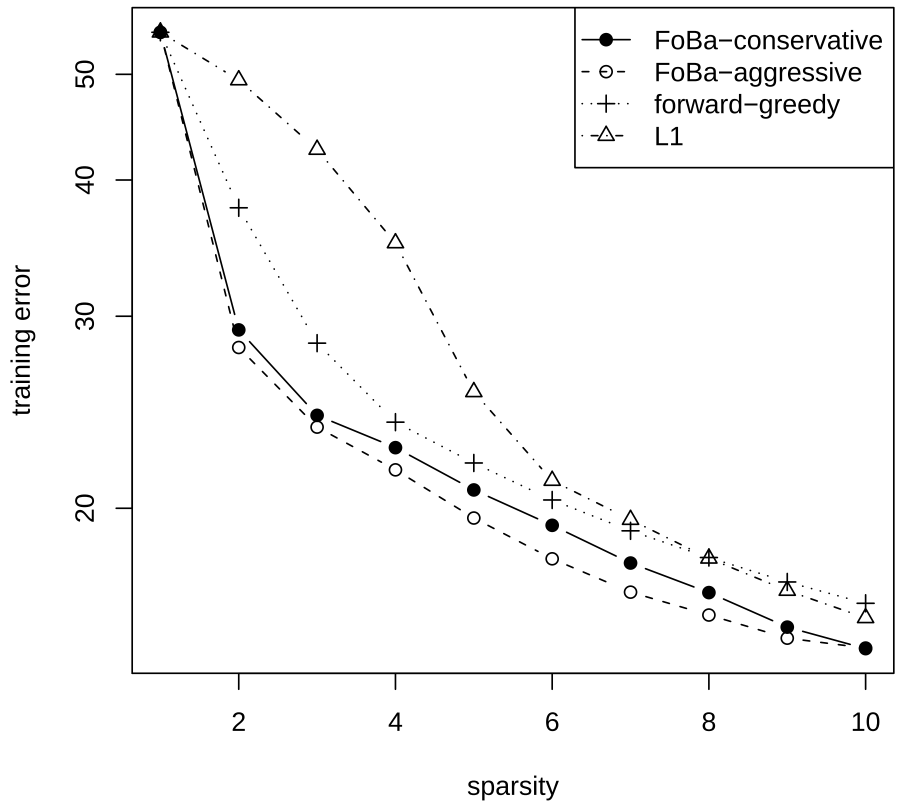
- $d = 500$, $n = 100$, noise $\sigma = 0.1$, moderately correlated design matrix
- exact sparse weight with $\bar{k} = 5$ and weights uniform $0 - 10$
- 50 random runs, resulting results for top five features

	FoBa-conservative	forward-greedy	L_1
least squares training error	0.093 ± 0.02	0.16 ± 0.089	0.25 ± 0.14
parameter estimation error	0.057 ± 0.2	0.52 ± 0.82	1.1 ± 1
feature selection error	0.76 ± 0.98	1.8 ± 1.1	3.2 ± 0.77

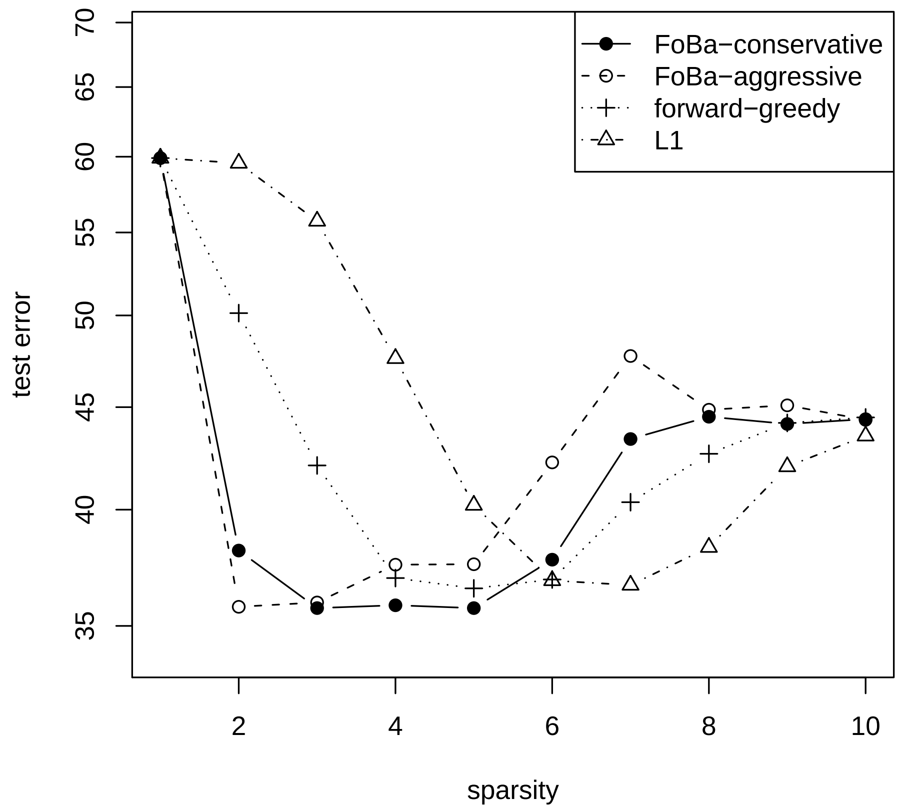
Real data experiment: Boston Housing

- least squares regression: 13 features + 1 constant feature,
- 506 data points: random 50 as training, remaining as test data ($n \gg d$)
- Example forward-greedy steps:
  - 6 13 4 8 2 3 10 1 7 11
- Example FoBa (conservative) steps:
  - 6 13 4 8 -4 2 4 3 -4 4 10 -4 -3 4 1 7
- Example L_1 steps (lars):
  - 6 2 13 4 8 10 3 11 7 12 5 9 1 -3 14 3

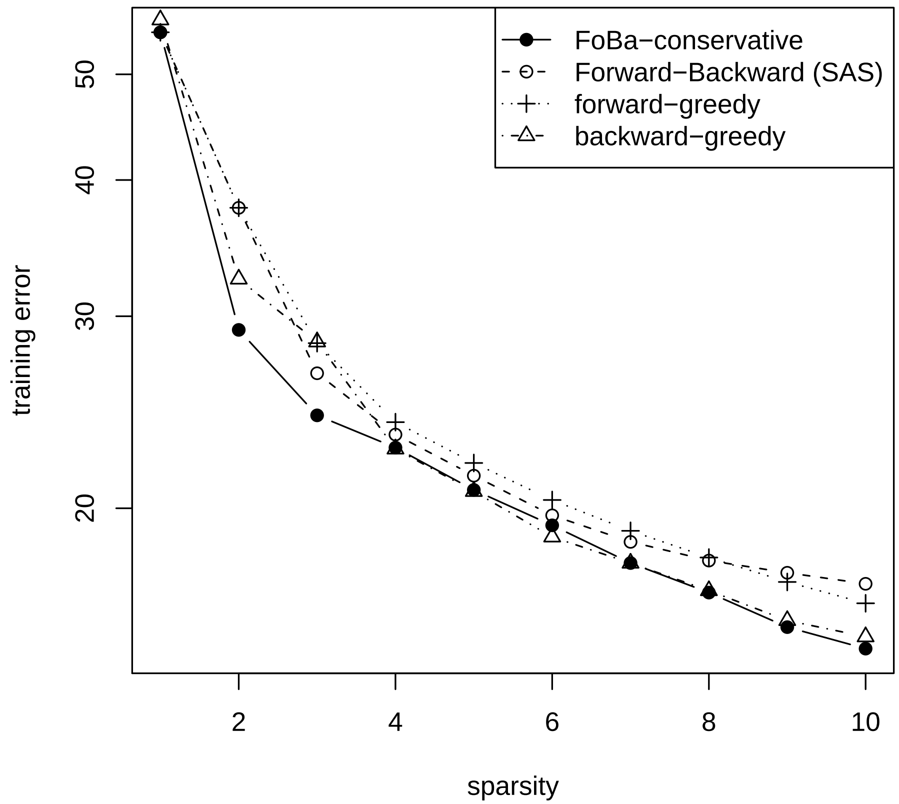
Training error



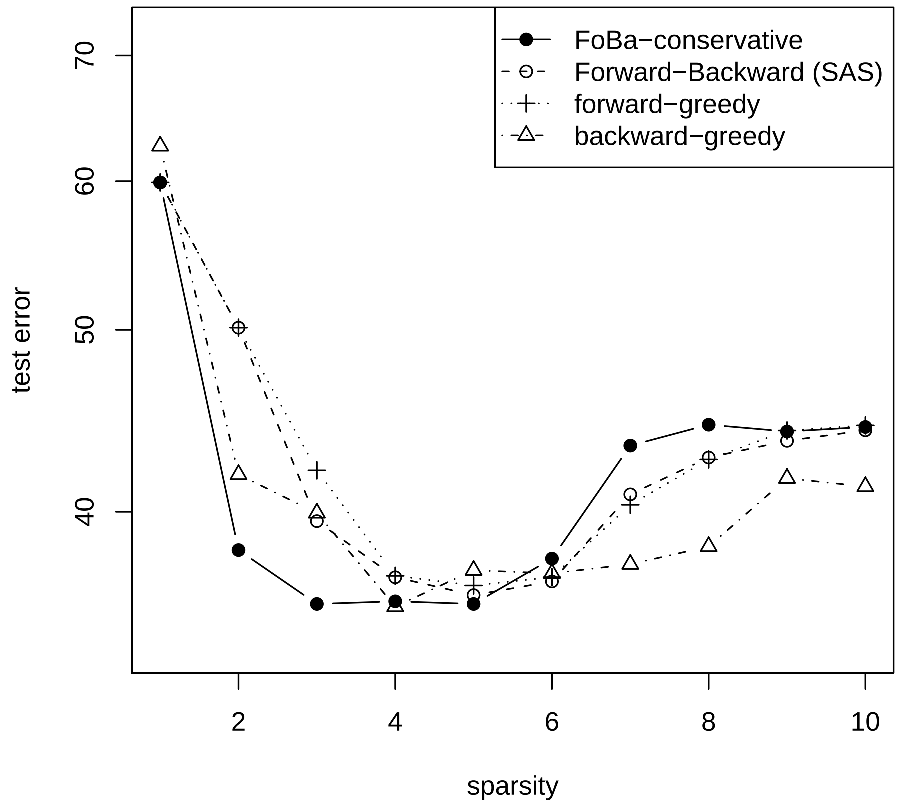
Test error


FoBa−conservative
FoBa−aggressive
forward−greedy
L1
test error
sparsity
70
65
60
55
50
45
40
35
2
4
6
8
10

Training error (additional comparisons)



Test error (additional comparisons)



Summary

- Traditional approximation methods for L_0 regularization
  - L_1 relaxation (bias: need non-convexity)
  - forward selection (not good for feature selection)
  - backward selection (cannot start with overfitted model)
- FoBa: combines the strength of forward backward selection
  - approximate path-following algorithm to directly solve L_0
  - theoretically: more effective than earlier algorithms
  - practically: closer to L_0 than forward-greedy and L_1
- A Final Remark: L_0 (sparsity) does not always lead to better prediction performance in practice (unstable for certain problems)