Poster T036

Learning Sparse Representations of High Dimensional Data on Large Scale Dictionaries

Zhen James Xiang, Hao Xu, Peter J. Ramadge
Dept. of EE, Princeton University

1. The Problem

The computational challenge of learning large scale sparse representations.

2. Lasso Screening

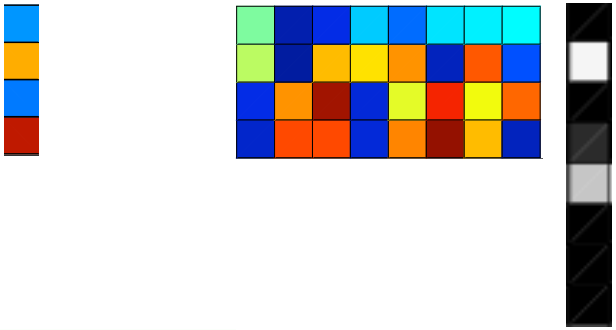
Solve lasso problems faster and obtain same solution.

3. Hierarchical Dictionary

Efficient algorithm framework to learn tree structured multi-layer dictionaries.

Sparse Representations: Why?

$$\mathbf{x} \approx \mathbf{B}\mathbf{w}$$



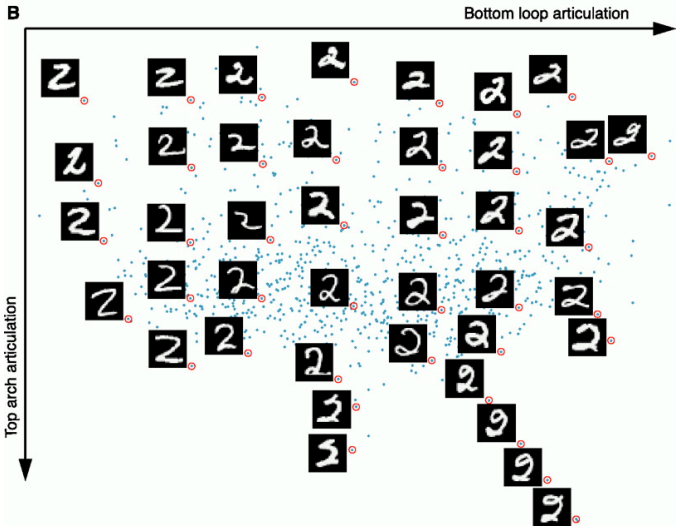


Image credit: Tenenbaum et al.

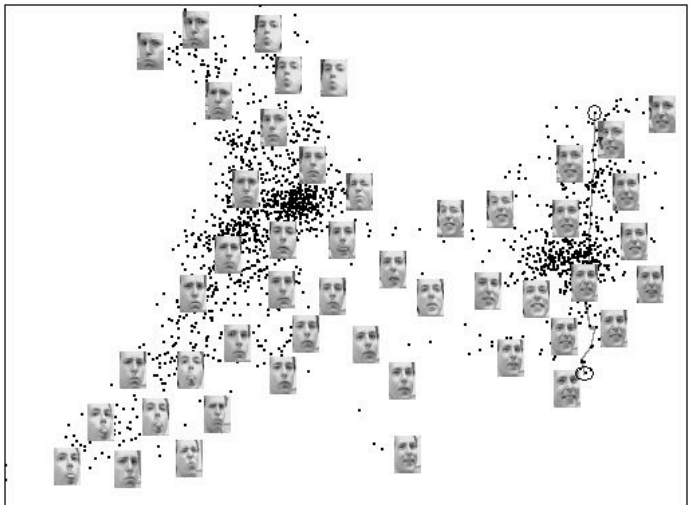
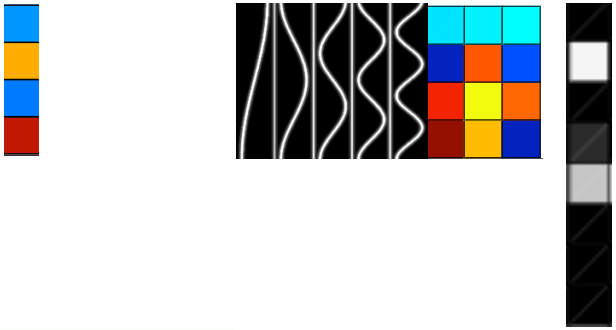


Image credit: He and Niyogi.

Sparse Representations: Why?

$$\mathbf{x} \approx \mathbf{B}\mathbf{w}$$



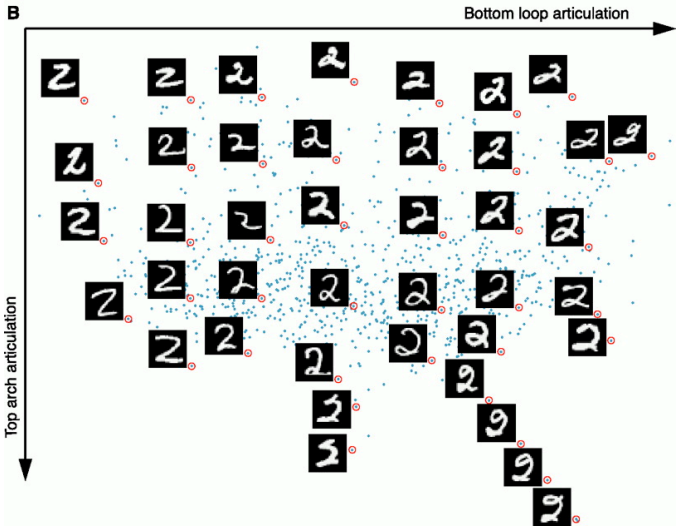


Image credit: Tenenbaum et al.

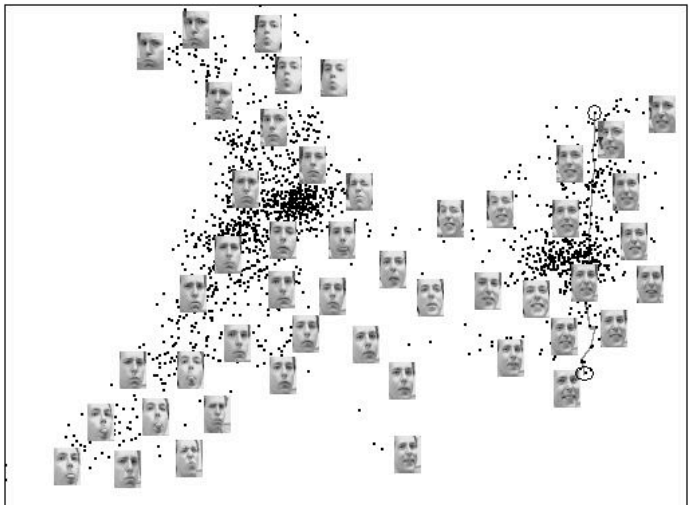
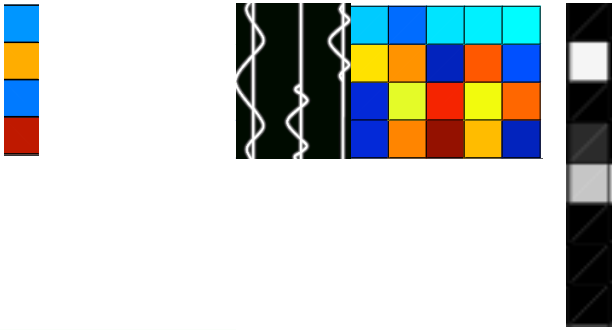


Image credit: He and Niyogi.

Sparse Representations: Why?

$$\mathbf{x} \approx \mathbf{B}\mathbf{w}$$



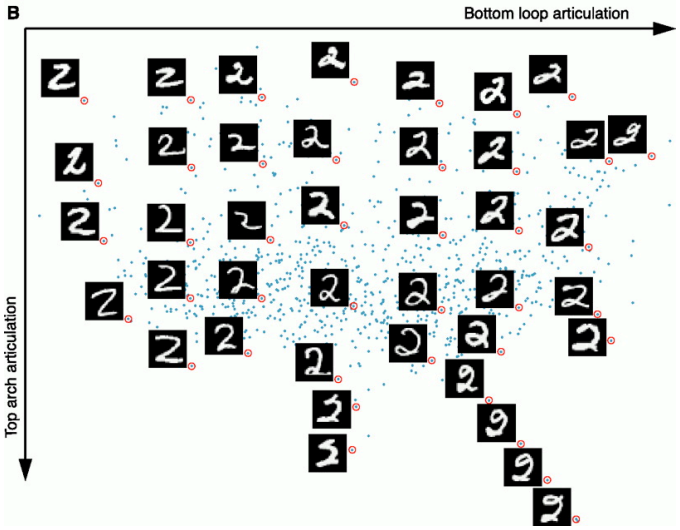


Image credit: Tenenbaum et al.

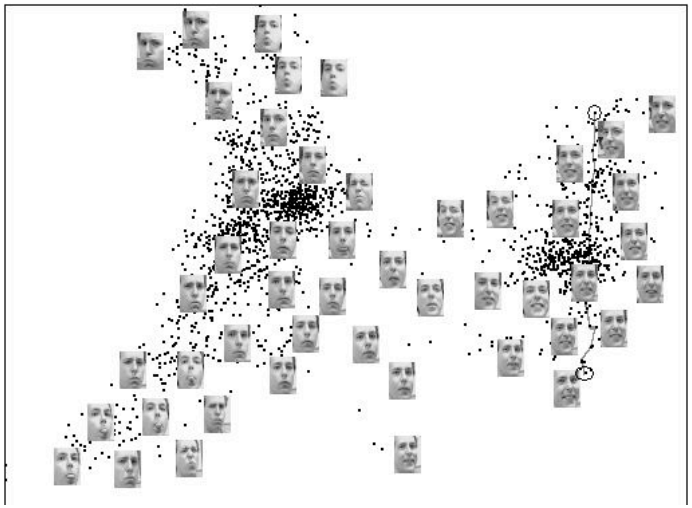
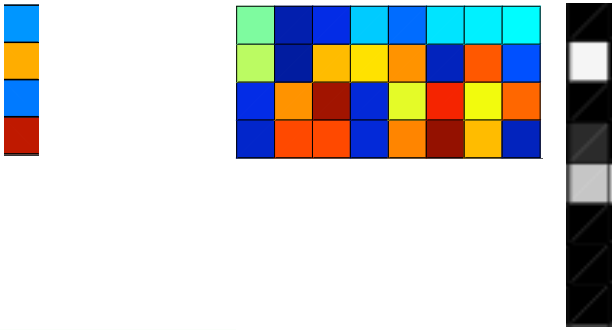


Image credit: He and Niyogi.

Sparse Representations: Why?

$$\mathbf{x} \approx \mathbf{B}\mathbf{w}$$



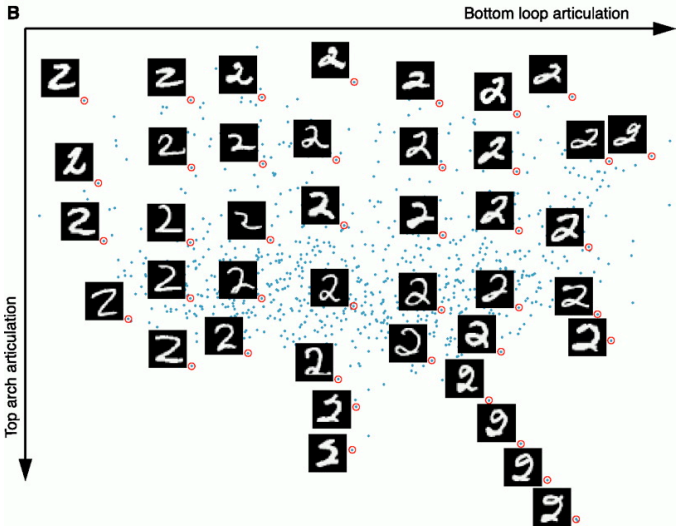


Image credit: Tenenbaum et al.

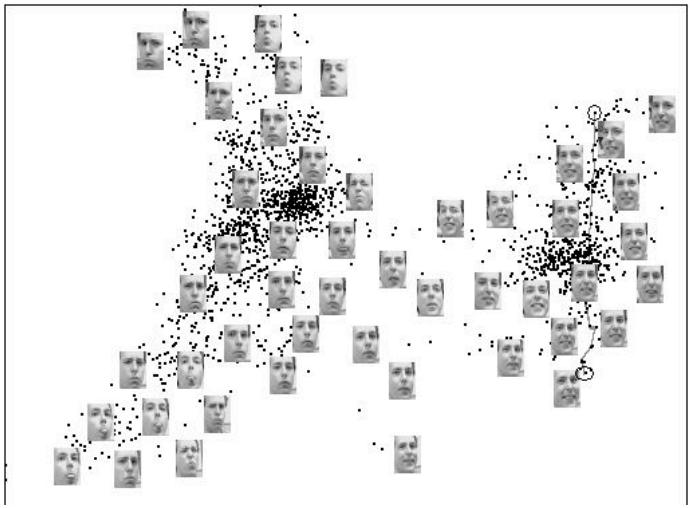
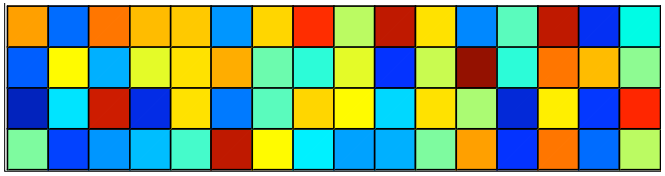
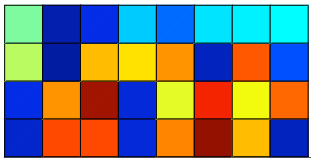


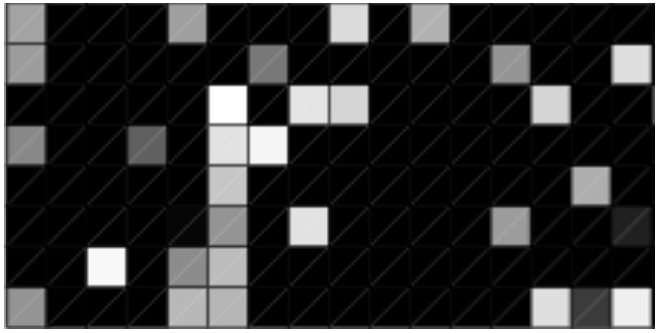
Image credit: He and Niyogi.

Sparse Representations: Why?

$$\mathbf{X} \approx \mathbf{BW}$$







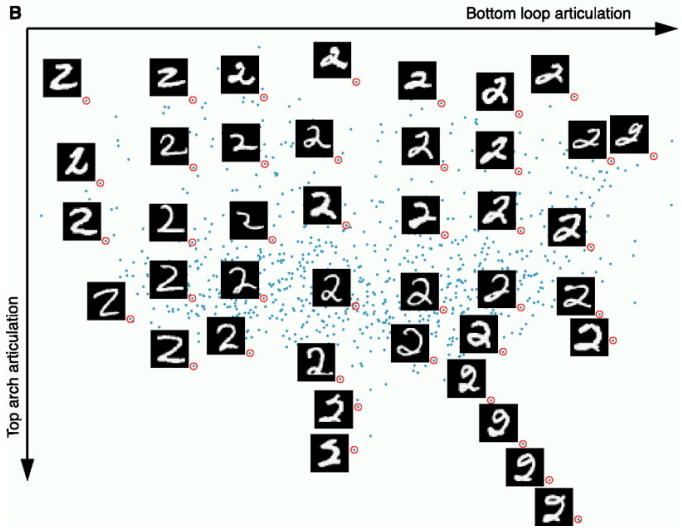


Image credit: Tenenbaum et al.

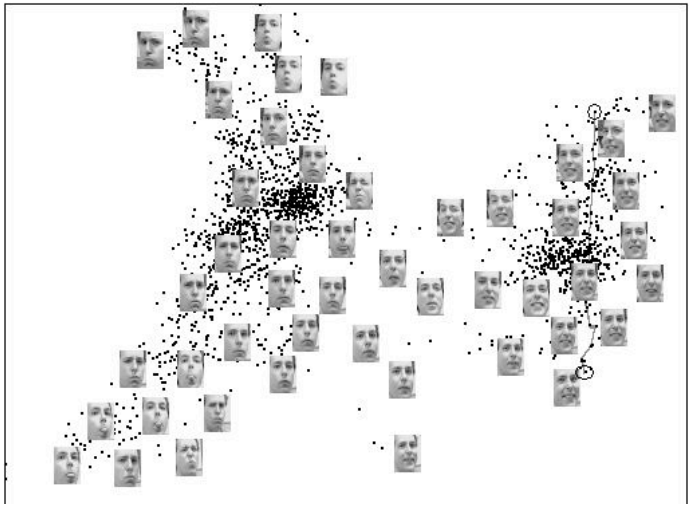
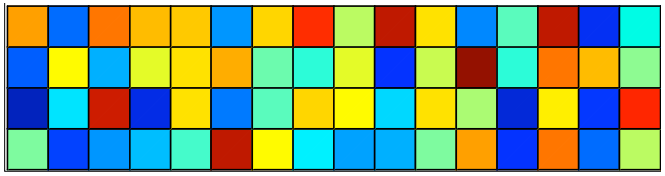
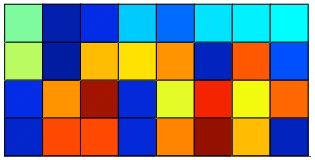


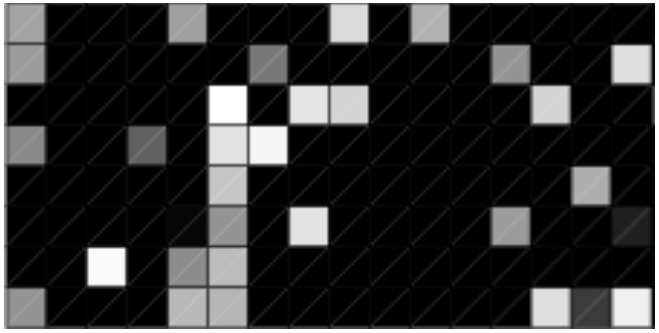
Image credit: He and Niyogi.

Sparse Representations: How?

$$\mathbf{X} \approx \mathbf{BW}$$

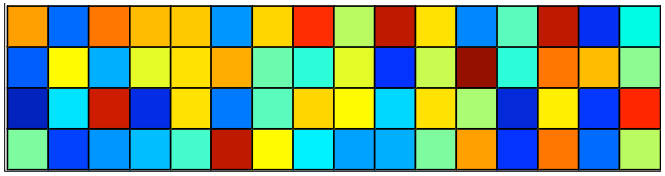


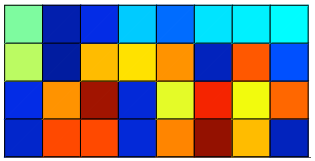


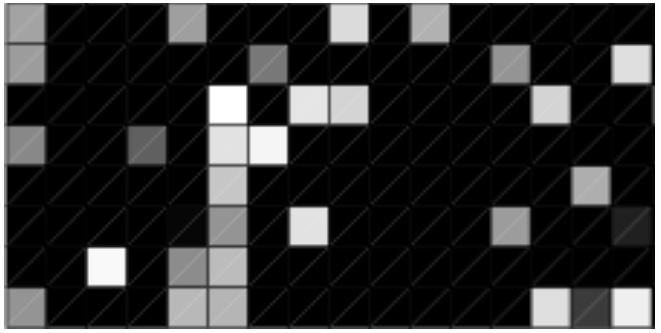


Sparse Representations: How?

$$\mathbf{X} \approx \mathbf{BW}$$





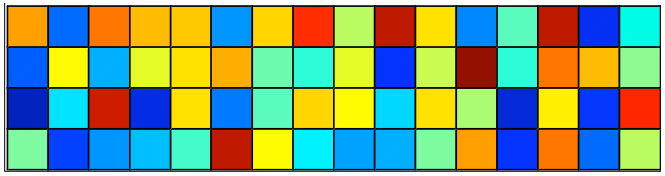
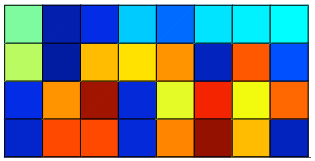
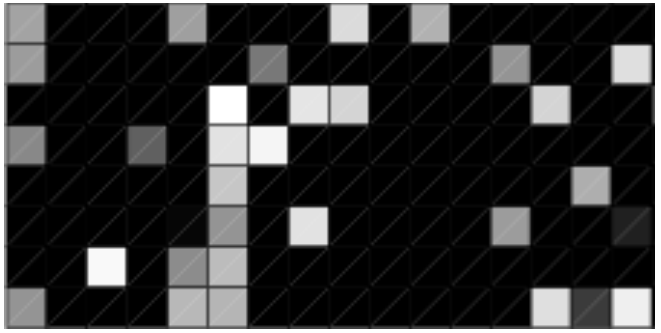


$$\min_{\mathbf{B},\mathbf{W}} \quad \frac{1}{2}\|\mathbf{X}-\mathbf{BW}\|_F^2 + \lambda\|\mathbf{W}\|_1$$

$$\text{s.t.} \quad \|\mathbf{b}_i\|_2^2 \le 1, \quad 1 \le i \le m.$$

Sparse Representations: How?

$$\mathbf{X} \approx \mathbf{BW}$$

$$\min_{\mathbf{B},\mathbf{W}} \quad \frac{1}{2}\|\mathbf{X}-\mathbf{BW}\|_F^2 + \lambda\|\mathbf{W}\|_1$$

$$\text{s.t.} \quad \|\mathbf{b}_i\|_2^2 \le 1, \quad 1 \le i \le m.$$

Updating **W**: one lasso problem per data point:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{x}-\mathbf{Bw}\|_2^2 + \lambda\|\mathbf{w}\|_1.$$

Updating **B**: a constrained least square problem.

Examples

Face Recognition: (Wright et al., 2009 PAMI; Wagner et al., 2011 PAMI)

- Dimension: ~32K (192x168 for YALE B Extended);
- Dictionary size: 30~40 times the number of subjects;
- No dictionary iteration yet.

Examples

Face Recognition: (Wright et al., 2009 PAMI; Wagner et al., 2011 PAMI)

- Dimension: ~32K (192x168 for YALE B Extended);
- Dictionary size: 30~40 times the number of subjects;
- No dictionary iteration yet.

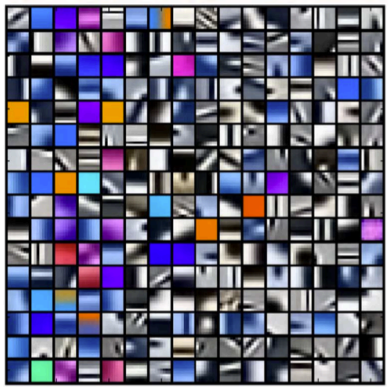


Image Restoration: (Mairal et al., 2008 TIP; Mairal et al., 2009 ICCV)

- Dimension: 192 (8*8*3), 1200 (20*20*3), ….
- Dictionary size: 256;
- Nearly millions of data points;
- At least hundreds of iterations.

Examples

Face Recognition: (Wright et al., 2009 PAMI; Wagner et al., 2011 PAMI)

- Dimension: ~32K (192x168 for YALE B Extended);
- Dictionary size: 30~40 times the number of subjects;
- No dictionary iteration yet.

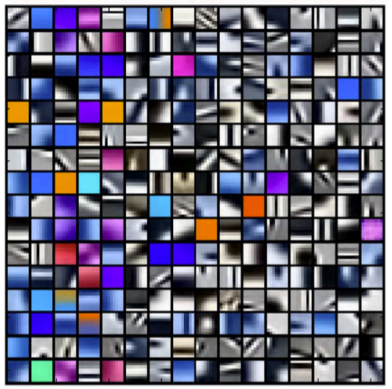


Image Restoration: (Mairal et al., 2008 TIP; Mairal et al., 2009 ICCV)

- Dimension: 192 (8*8*3), 1200 (20*20*3), ….
- Dictionary size: 256;
- Nearly millions of data points;
- At least hundreds of iterations.

This Conference:

- Anna Gilbert, *Sparsity: algorithms, approximations, and analysis*, Invited Talk;
- Morioka and Satoh, *Generalized lasso based approximation of sparse coding for visual recognition,* Poster T027;
- Szlam et al., *Structured sparse coding via lateral inhibition,* Poster W045;
- This talk: Poster T036.

1. The Problem

The computational challenge of learning large scale sparse representations.

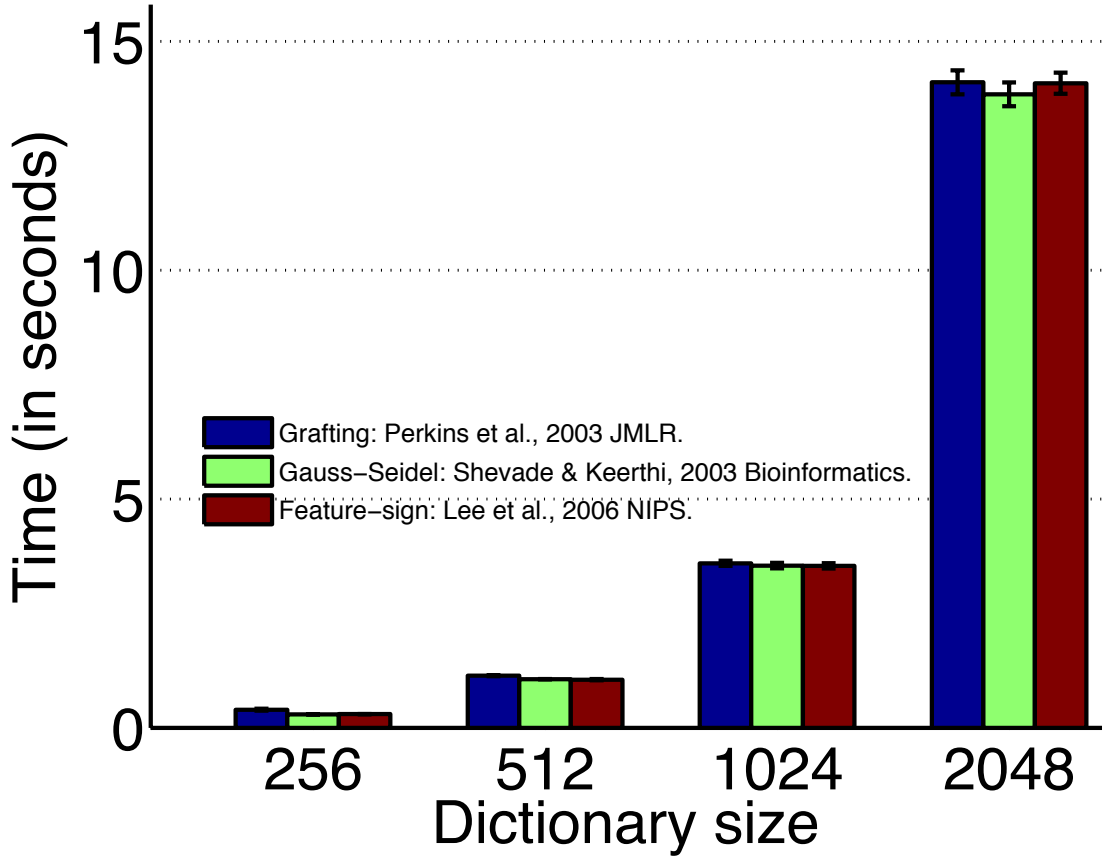
2. Lasso Screening

Solve lasso problems faster and obtain same solution.

3. Hierarchical Dictionary

Efficient algorithm framework to learn tree structured multi-layer dictionaries.

Solving One Lasso Problem

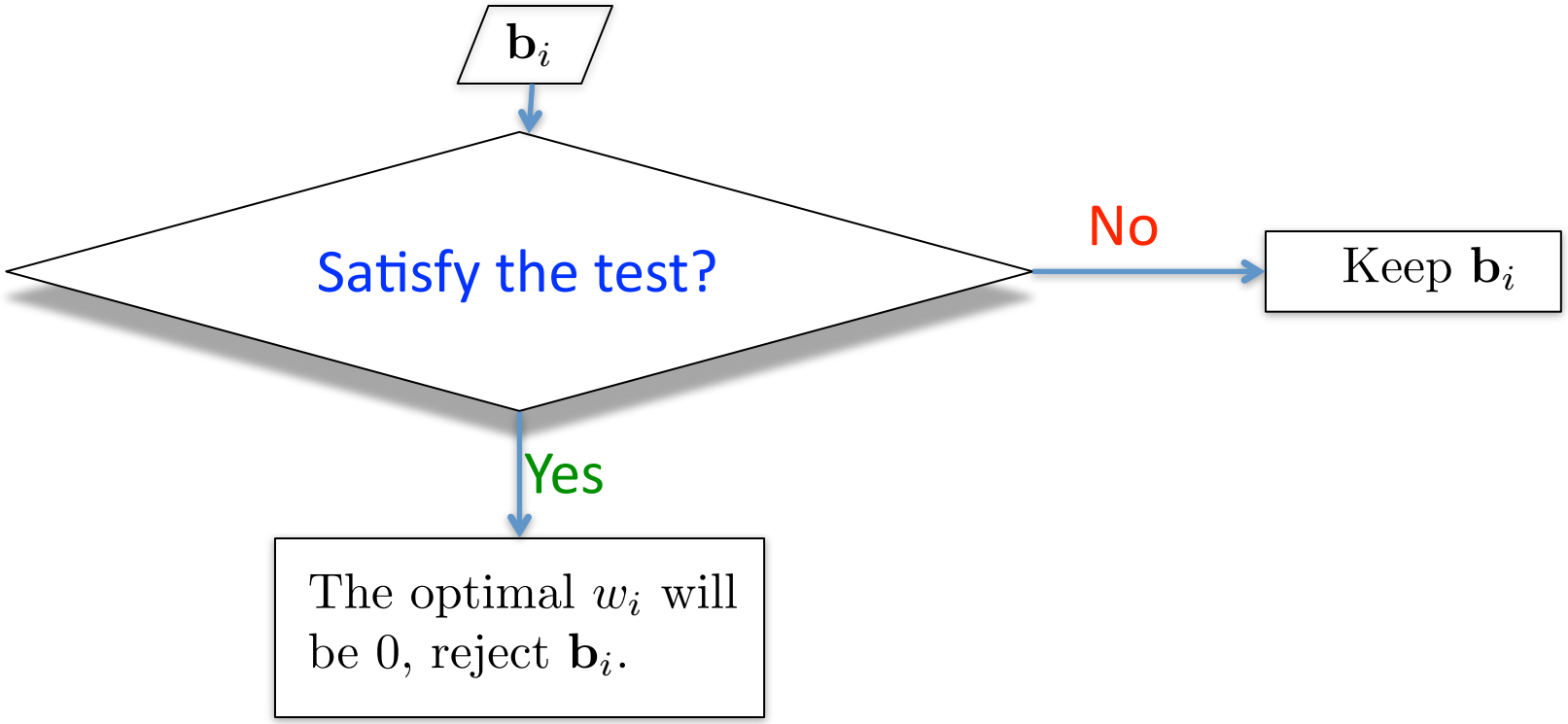


- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.5;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Screening Tests

Lasso Problem: $\min_{w_1,w_2,\ldots,w_m} \frac{1}{2}\|\mathbf{x} - \sum_{i=1}^{m} w_i \mathbf{b}_i\|_2^2 + \lambda \sum_{i=1}^{m} |w_i|.$

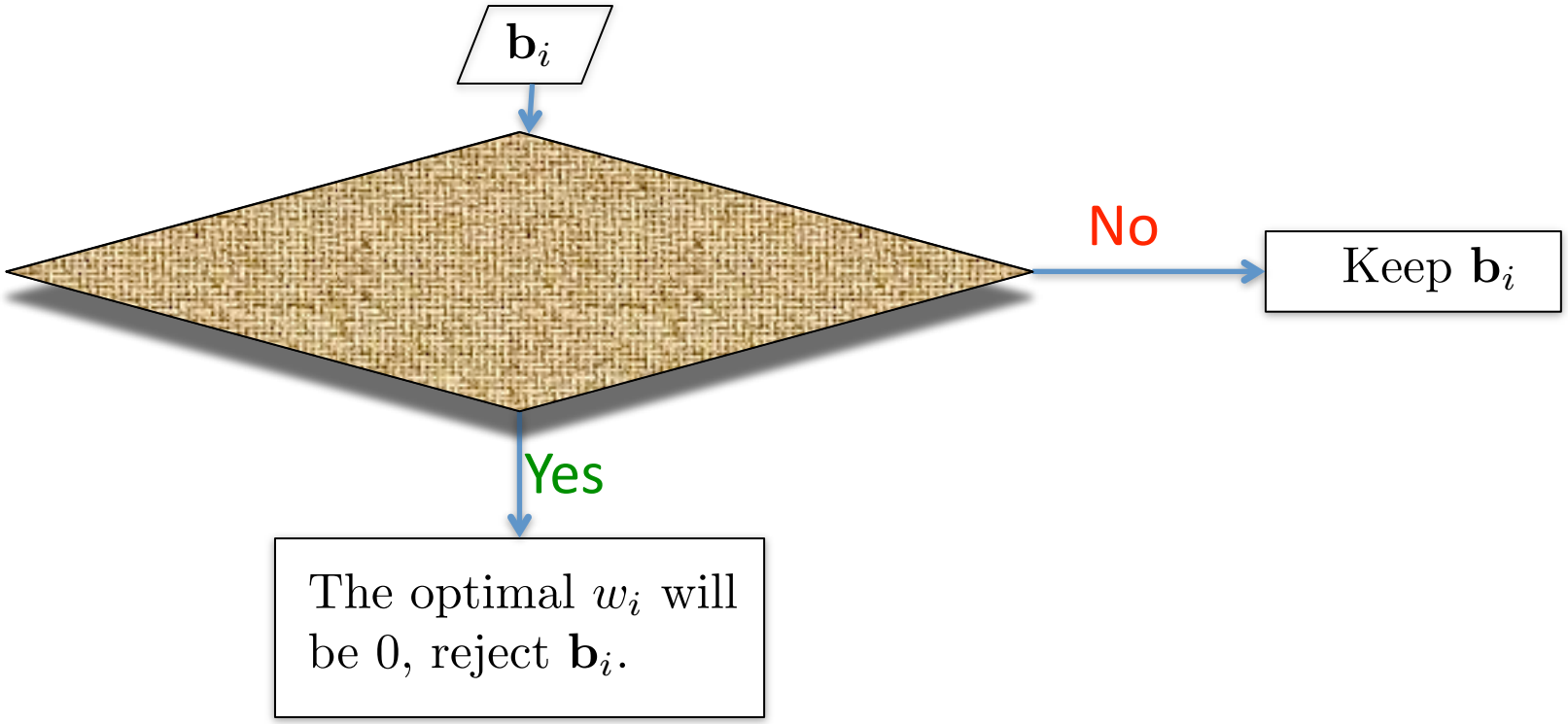
Assume $\|\mathbf{x}\|_2 = \|\mathbf{b}_i\|_2 = 1.$



Screening Tests

Lasso Problem: $\min_{w_1,w_2,\ldots,w_m} \frac{1}{2}\|\mathbf{x} - \sum_{i=1}^{m} w_i \mathbf{b}_i\|_2^2 + \lambda \sum_{i=1}^{m} |w_i|.$

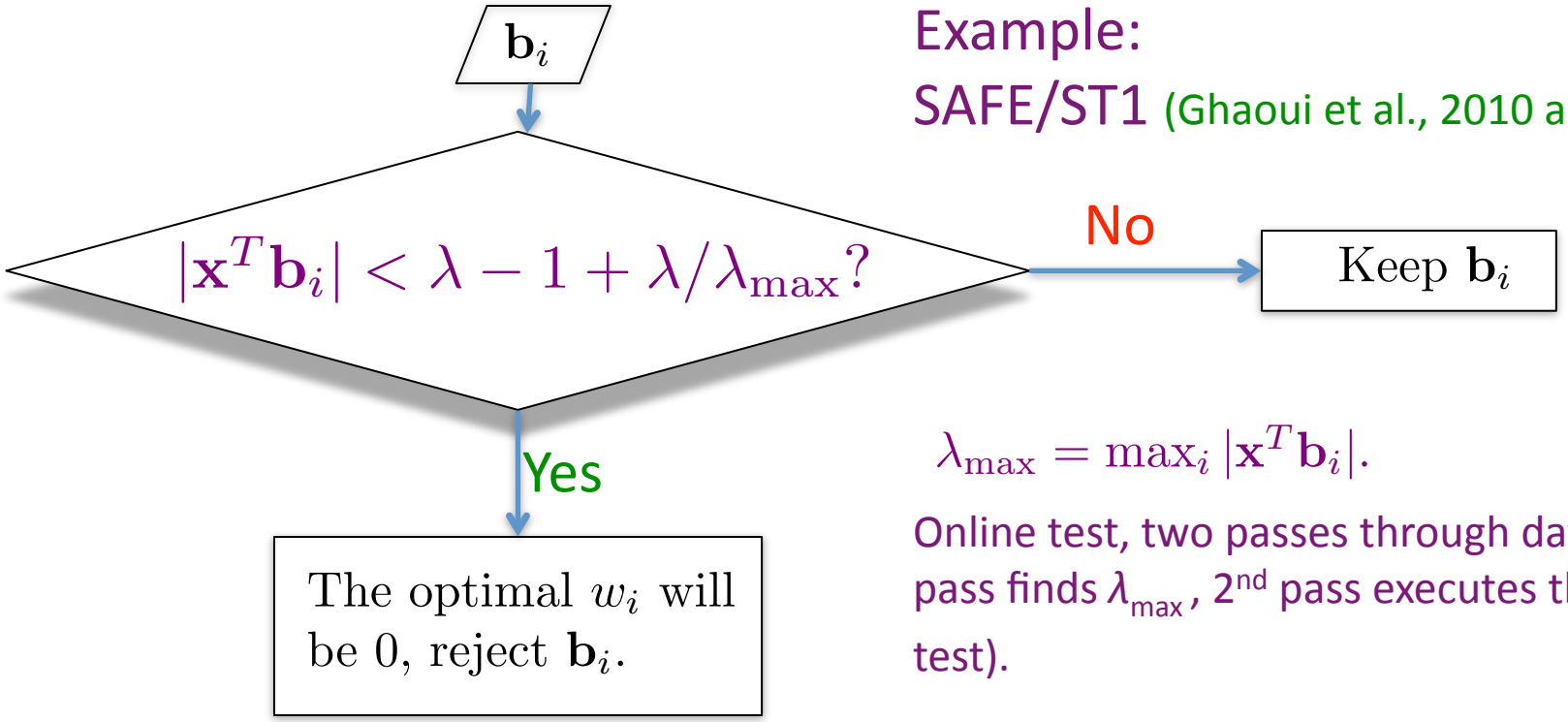
Assume $\|\mathbf{x}\|_2 = \|\mathbf{b}_i\|_2 = 1.$



Screening Tests

Lasso Problem: $\min_{w_1,w_2,\ldots,w_m} \frac{1}{2}\|\mathbf{x} - \sum_{i=1}^{m} w_i \mathbf{b}_i\|_2^2 + \lambda \sum_{i=1}^{m} |w_i|.$

Assume $\|\mathbf{x}\|_2 = \|\mathbf{b}_i\|_2 = 1.$

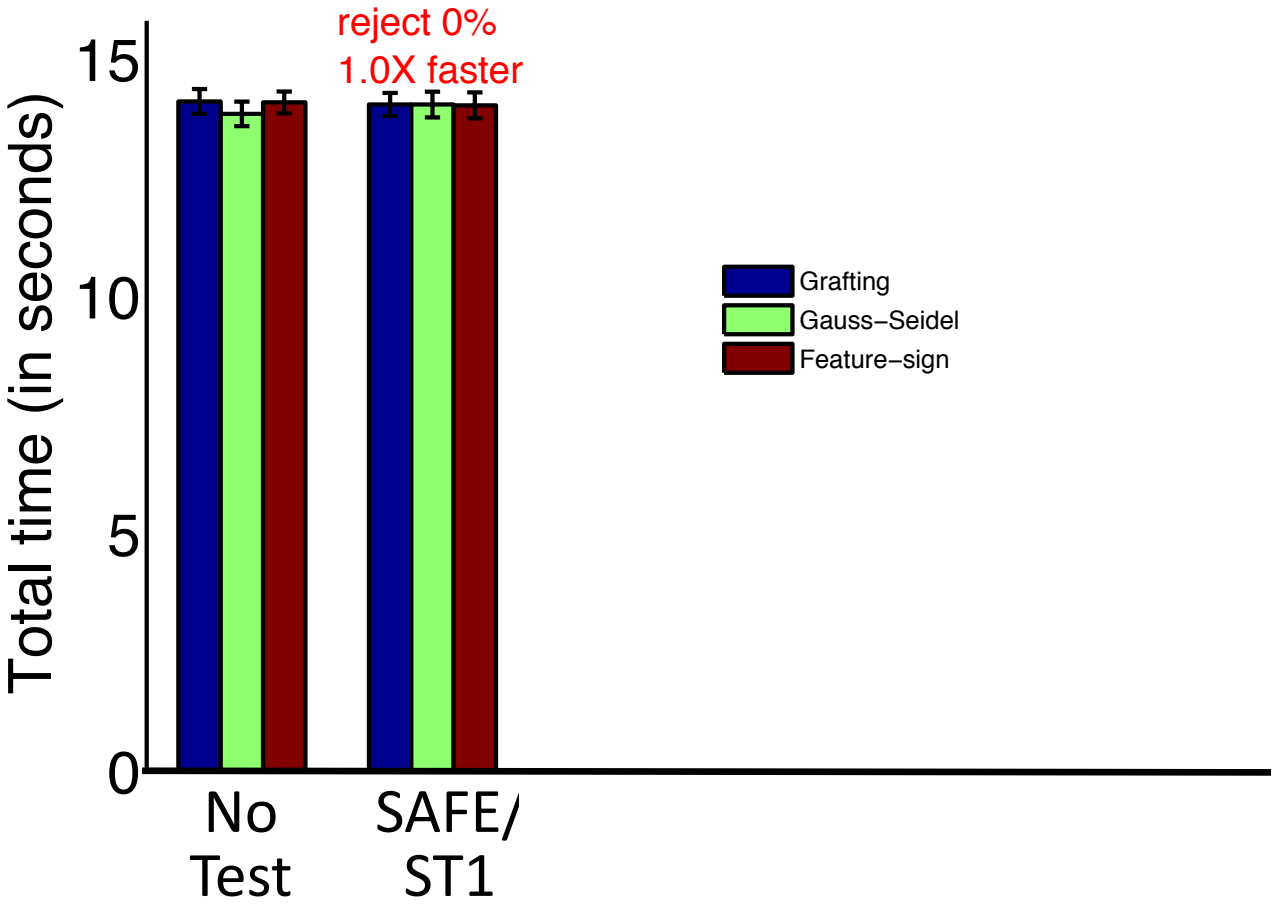


Example:
SAFE/ST1 (Ghaoui et al., 2010 arXiv).

$\lambda_{\max} = \max_i |\mathbf{x}^T\mathbf{b}_i|.$

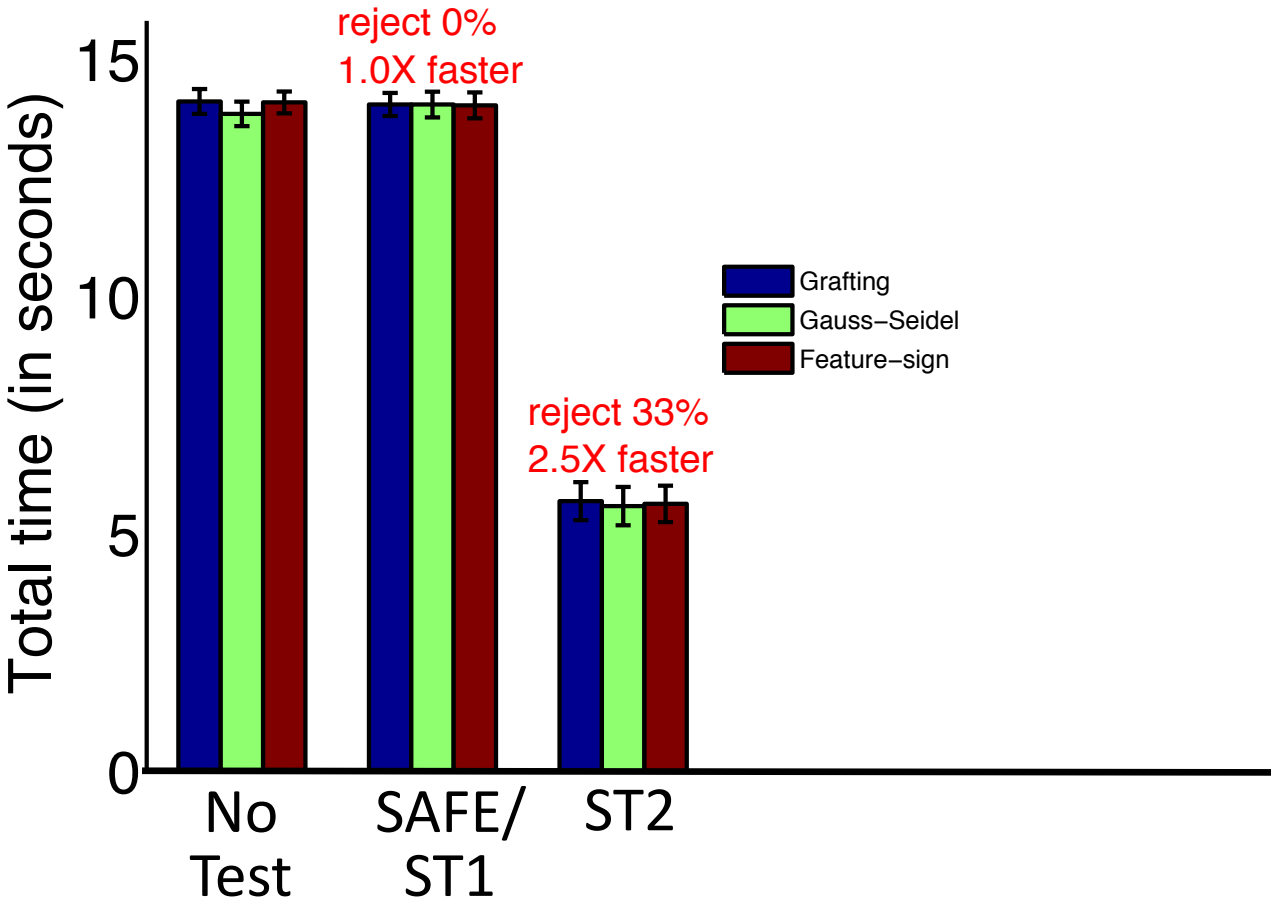
Online test, two passes through data (1st pass finds λ_{max}, 2nd pass executes the test).

Speed Up Lasso



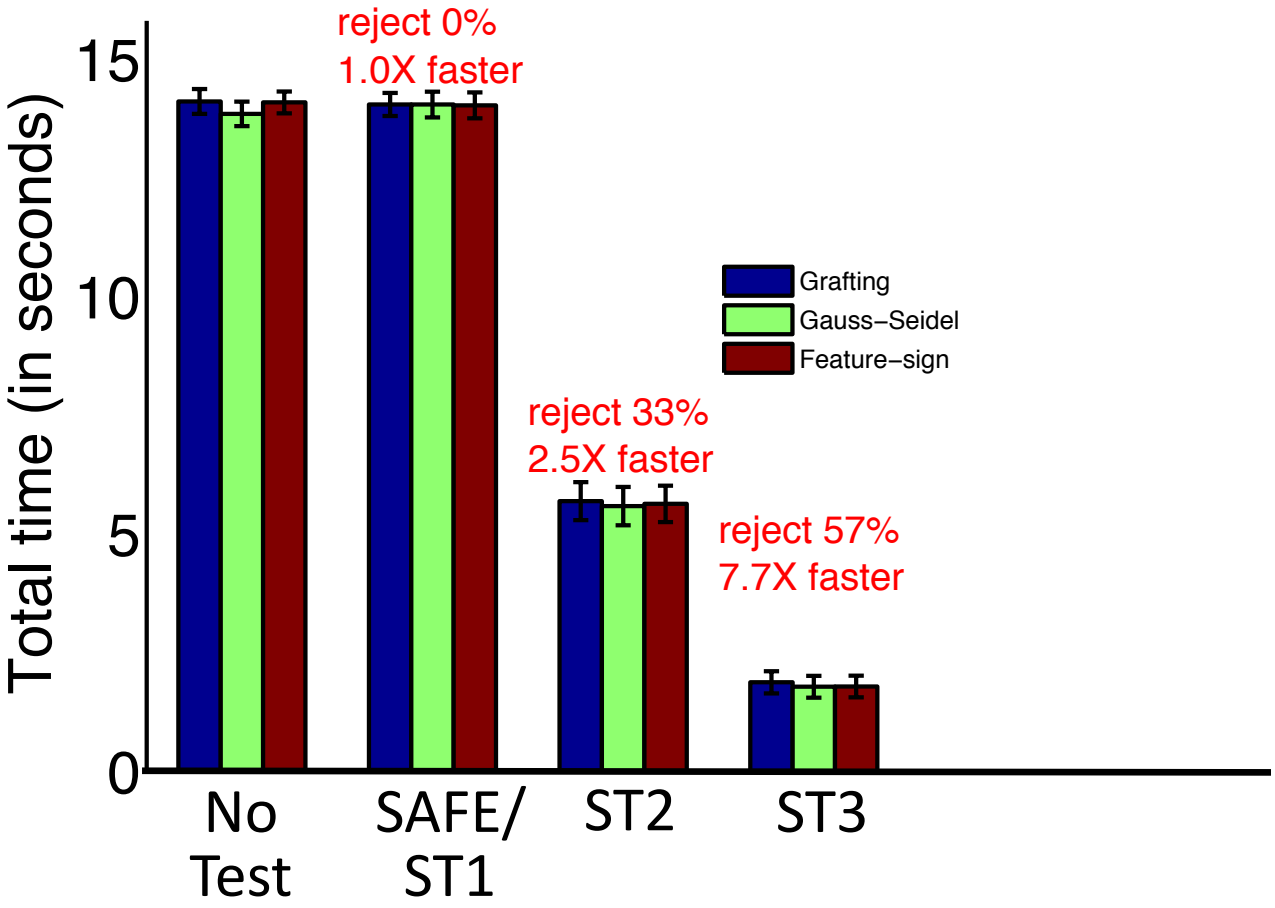
- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.5; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Speed Up Lasso



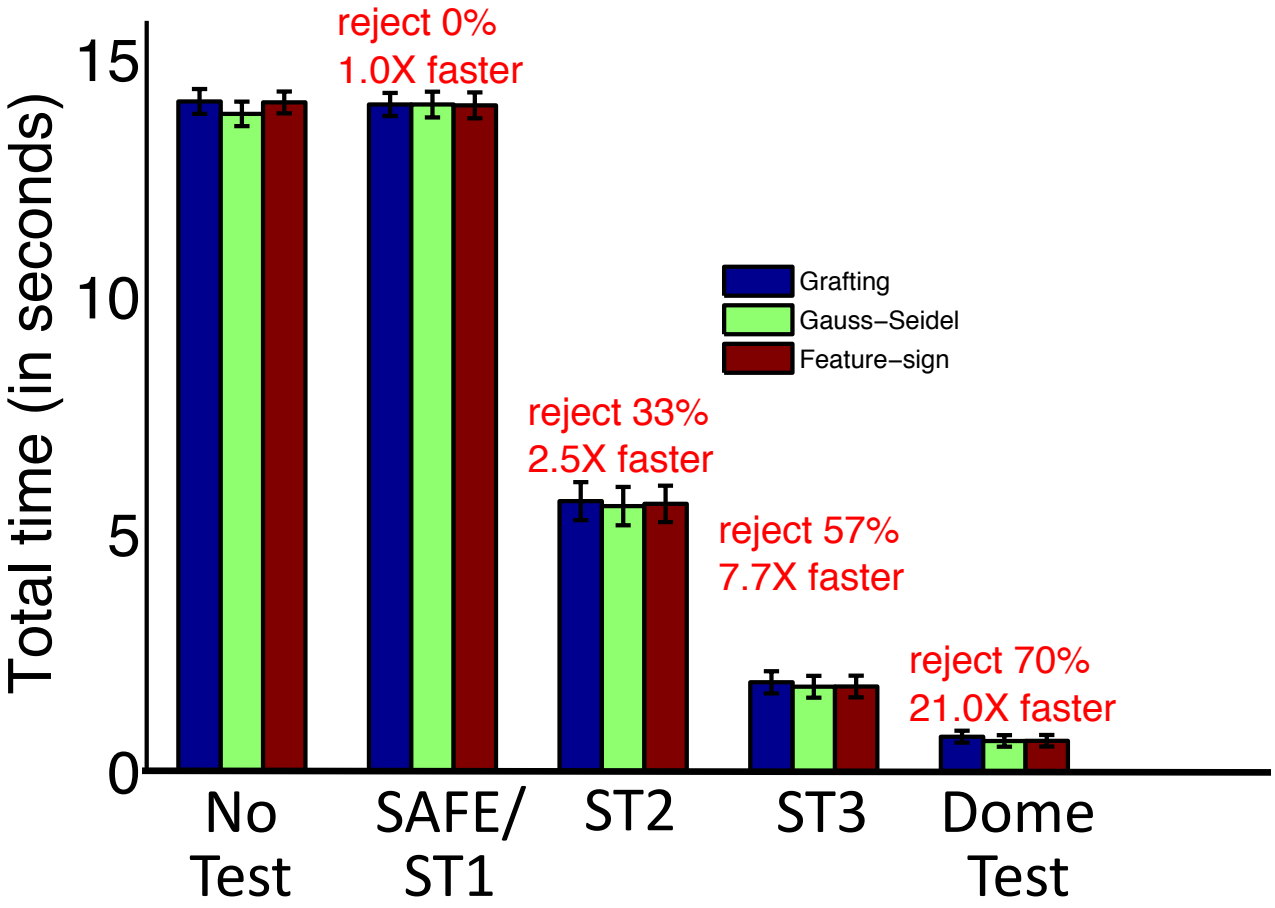
- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.5; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Speed Up Lasso



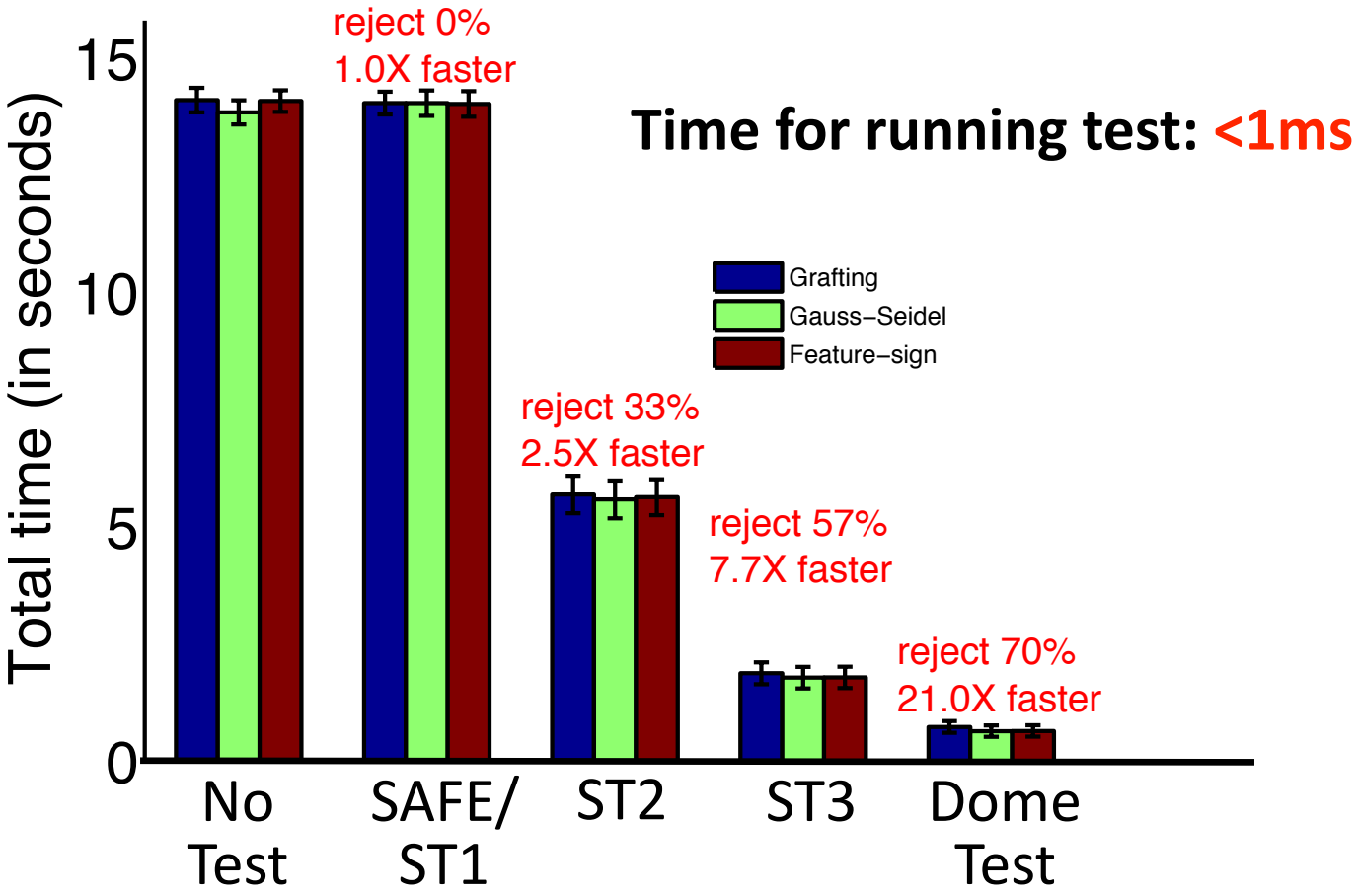
- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.5; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Speed Up Lasso



- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.5; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Speed Up Lasso



- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.5; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Fast and Online

- SAFE/ST1 (Ghaoui et al.): $|\mathbf{x}^T\mathbf{b}_i| < \lambda - 1 + \lambda/\lambda_{\max}$.
- ST2: $|\mathbf{x}^T\mathbf{b}_i| < \lambda_{\max}(1 - 2r_3)$.
- ST3: $|\mathbf{x}^T\mathbf{b}_i - (\lambda_{\max} - \lambda)\mathbf{b}_*^T\mathbf{b}_i| < \lambda(1 - r_3)$.

$$\lambda_{\max} = \max_i |\mathbf{x}^T\mathbf{b}_i|,\ \mathbf{b}_* = \arg\max_{\mathbf{b}\in\{\pm\mathbf{b}_i\}} \mathbf{x}^T\mathbf{b},\quad r_3 = \sqrt{1/\lambda_{\max}^2 - 1}(\lambda_{\max}/\lambda - 1).$$

Fast and Online

- SAFE/ST1 (Ghaoui et al.): $|\mathbf{x}^T\mathbf{b}_i| < \lambda - 1 + \lambda/\lambda_{\max}$.
- ST2: $|\mathbf{x}^T\mathbf{b}_i| < \lambda_{\max}(1 - 2r_3)$.
- ST3: $|\mathbf{x}^T\mathbf{b}_i - (\lambda_{\max} - \lambda)\mathbf{b}_*^T\mathbf{b}_i| < \lambda(1 - r_3)$.

$$\lambda_{\max} = \max_i |\mathbf{x}^T\mathbf{b}_i|,\ \mathbf{b}_* = \arg\max_{\mathbf{b}\in\{\pm\mathbf{b}_i\}} \mathbf{x}^T\mathbf{b},\quad r_3 = \sqrt{1/\lambda_{\max}^2 - 1}(\lambda_{\max}/\lambda - 1).$$

All tests are online tests:

- Two passes;
- Memory footprint = 3 codewords.

Fast and Online

- SAFE/ST1 (Ghaoui et al.): $|\mathbf{x}^T\mathbf{b}_i| < \lambda - 1 + \lambda/\lambda_{\max}$.
- ST2: $|\mathbf{x}^T\mathbf{b}_i| < \lambda_{\max}(1 - 2r_3)$.
- ST3: $|\mathbf{x}^T\mathbf{b}_i - (\lambda_{\max} - \lambda)\mathbf{b}_*^T\mathbf{b}_i| < \lambda(1 - r_3)$.

$$\lambda_{\max} = \max_i |\mathbf{x}^T\mathbf{b}_i|, \ \mathbf{b}_* = \arg\max_{\mathbf{b}\in\{\pm\mathbf{b}_i\}} \mathbf{x}^T\mathbf{b}, \quad r_3 = \sqrt{1/\lambda_{\max}^2 - 1}(\lambda_{\max}/\lambda - 1).$$

All tests are online tests:

- Two passes;
- Memory footprint = 3 codewords.

Here is the fine print:

- ST2,ST3 and Dome Test are only powerful when $\lambda_{\max}$ is large.
- Luckily, high correlation is common in real world data sets (Wright et al., 2010).

Fast and Online

- SAFE/ST1 (Ghaoui et al.): $|\mathbf{x}^T\mathbf{b}_i| < \lambda - 1 + \lambda/\lambda_{\max}$.
- ST2: $|\mathbf{x}^T\mathbf{b}_i| < \lambda_{\max}(1 - 2r_3)$.
- ST3: $|\mathbf{x}^T\mathbf{b}_i - (\lambda_{\max} - \lambda)\mathbf{b}_*^T\mathbf{b}_i| < \lambda(1 - r_3)$.

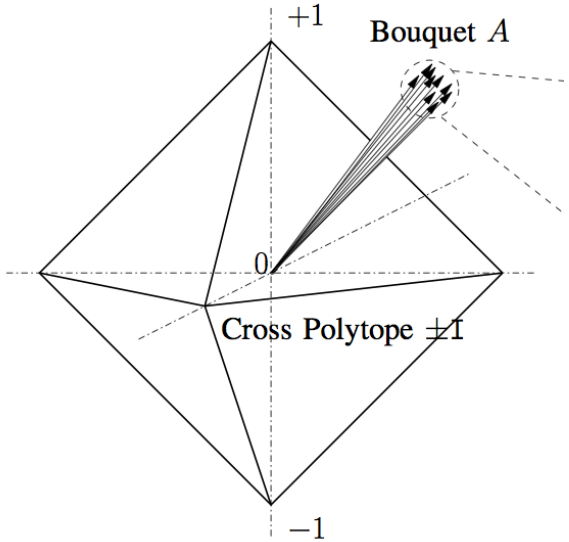
$$\lambda_{\max} = \max_i |\mathbf{x}^T\mathbf{b}_i|, \ \mathbf{b}_* = \arg\max_{\mathbf{b}\in\{\pm\mathbf{b}_i\}} \mathbf{x}^T\mathbf{b}, \quad r_3 = \sqrt{1/\lambda_{\max}^2 - 1}(\lambda_{\max}/\lambda - 1).$$

All tests are online tests:

- Two passes;
- Memory footprint = 3 codewords.

Here is the fine print:

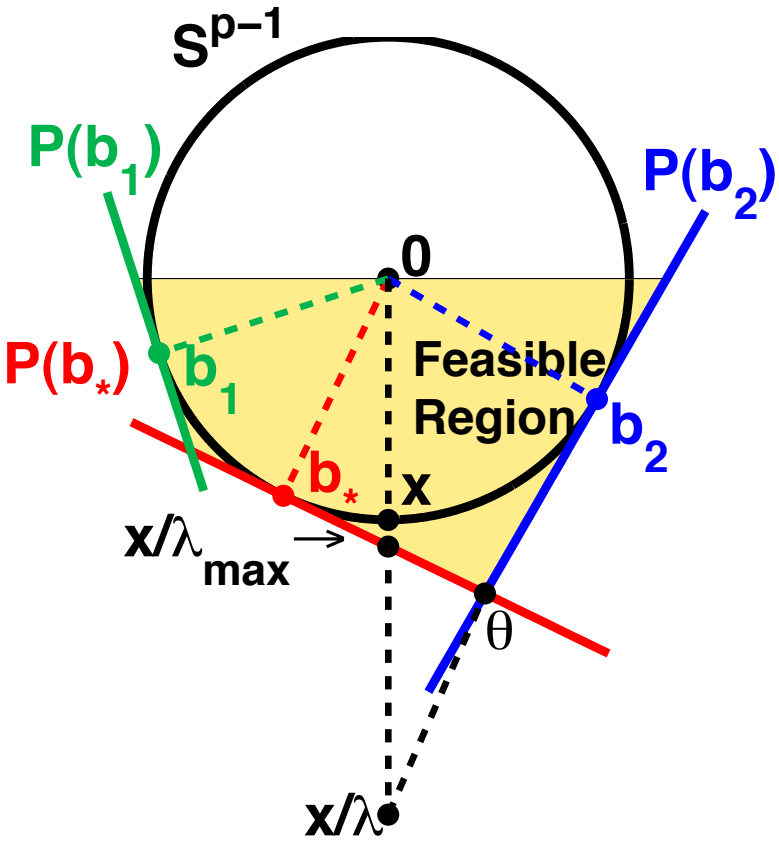
- ST2,ST3 and Dome Test are only powerful when $\lambda_{\max}$ is large.
- Luckily, high correlation is common in real world data sets (Wright et al., 2010).



The Math Behind the Tests

Dual Problem:

$$\max_{\boldsymbol{\theta}} \quad \frac{1}{2}\|\mathbf{x}\|_2^2 - \frac{\lambda^2}{2}\|\boldsymbol{\theta} - \frac{\mathbf{x}}{\lambda}\|_2^2$$
$$\text{s.t.} \quad |\boldsymbol{\theta}^T \mathbf{b}_i| \leq 1 \quad \forall i = 1, 2, \ldots, m.$$



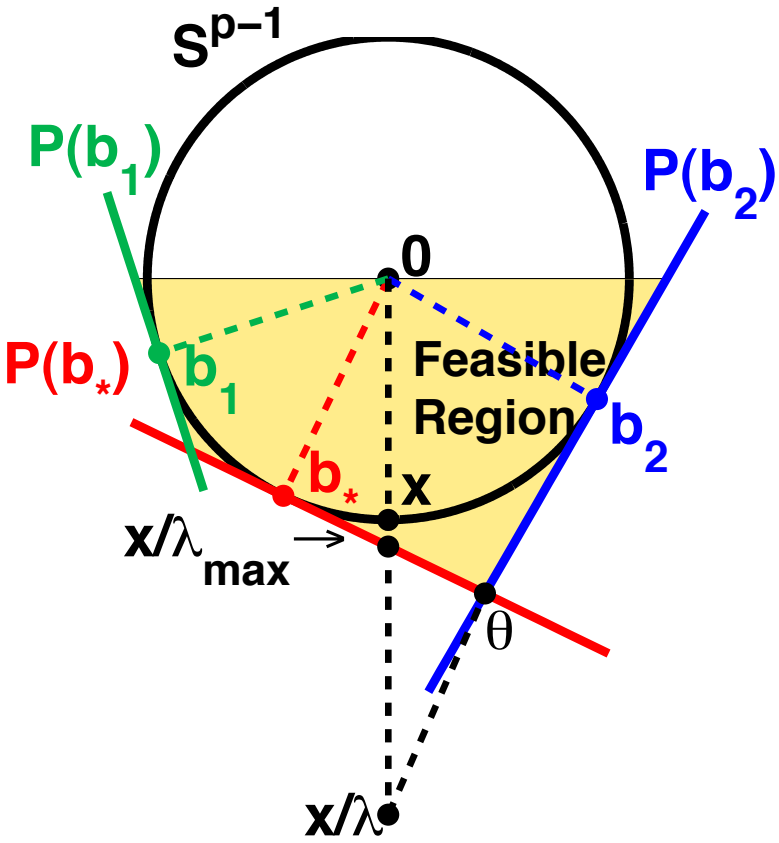
The Math Behind the Tests

Dual Problem:

$$\max_{\boldsymbol{\theta}} \quad \frac{1}{2}\|\mathbf{x}\|_2^2 - \frac{\lambda^2}{2}\|\boldsymbol{\theta} - \frac{\mathbf{x}}{\lambda}\|_2^2$$
$$\text{s.t.} \quad |\boldsymbol{\theta}^T \mathbf{b}_i| \leq 1 \quad \forall i = 1, 2, \ldots, m.$$

Core Rejection Test:

$$|\tilde{\boldsymbol{\theta}}^T \mathbf{b}_i| < 1 \Rightarrow \tilde{w}_i = 0.$$



The Math Behind the Tests

Dual Problem:

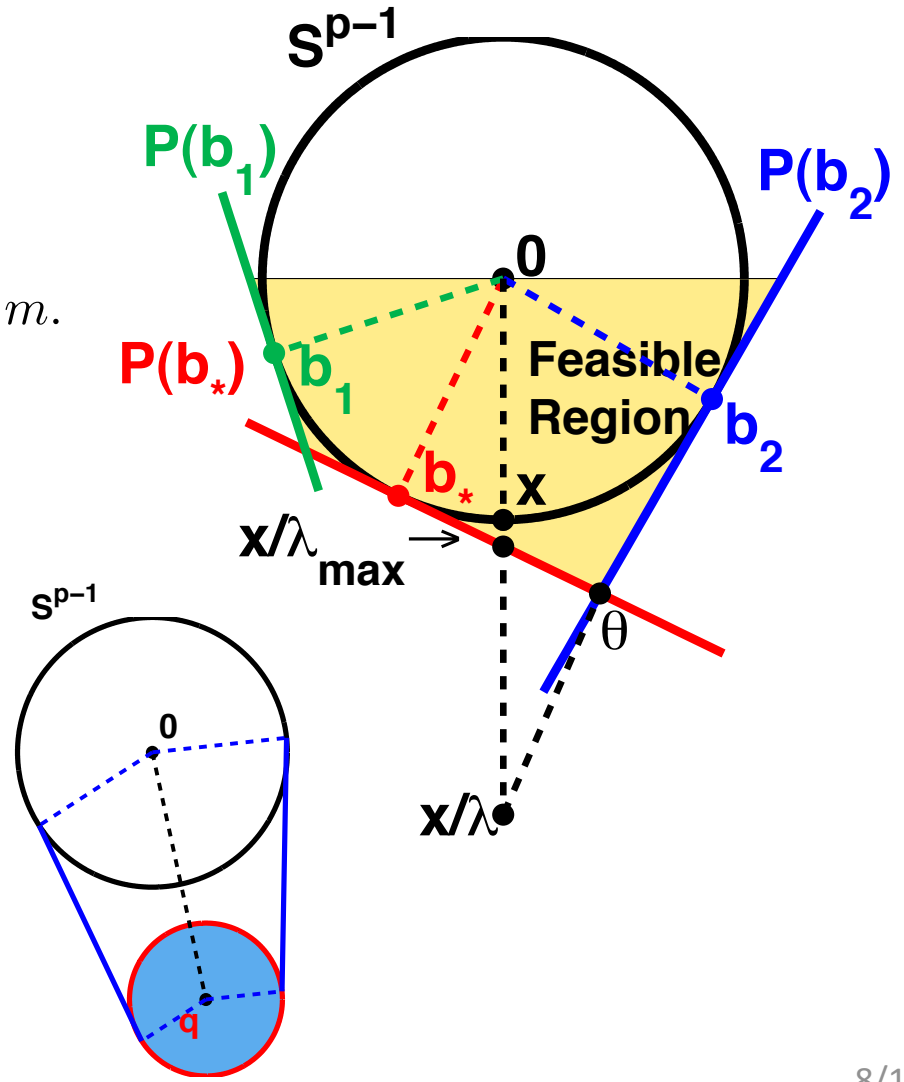
$$\max_{\boldsymbol{\theta}} \quad \frac{1}{2}\|\mathbf{x}\|_2^2 - \frac{\lambda^2}{2}\|\boldsymbol{\theta} - \frac{\mathbf{x}}{\lambda}\|_2^2$$
$$\text{s.t.} \quad |\boldsymbol{\theta}^T \mathbf{b}_i| \leq 1 \quad \forall i = 1, 2, \ldots, m.$$

Core Rejection Test:

$$|\tilde{\boldsymbol{\theta}}^T \mathbf{b}_i| < 1 \Rightarrow \tilde{w}_i = 0.$$

Sphere Test:

If $\tilde{\boldsymbol{\theta}}$ satisfies $\|\tilde{\boldsymbol{\theta}} - \mathbf{q}\|_2 \leq r$,
then $|\mathbf{q}^T \mathbf{b}_i| < (1 - r) \Rightarrow \tilde{w}_i = 0.$



1. The Problem

The computational challenge of learning large scale sparse representations.

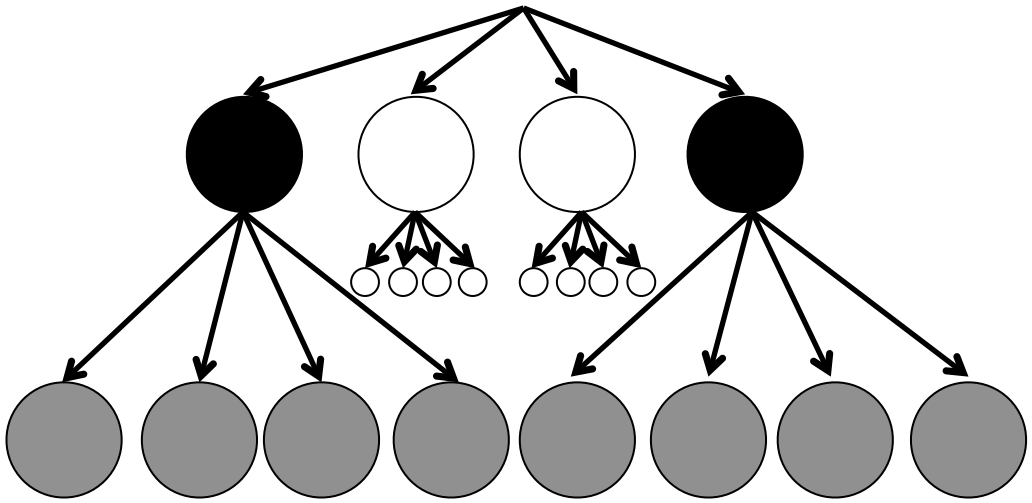
2. Lasso Screening

Solve lasso problems faster and obtain same solution.

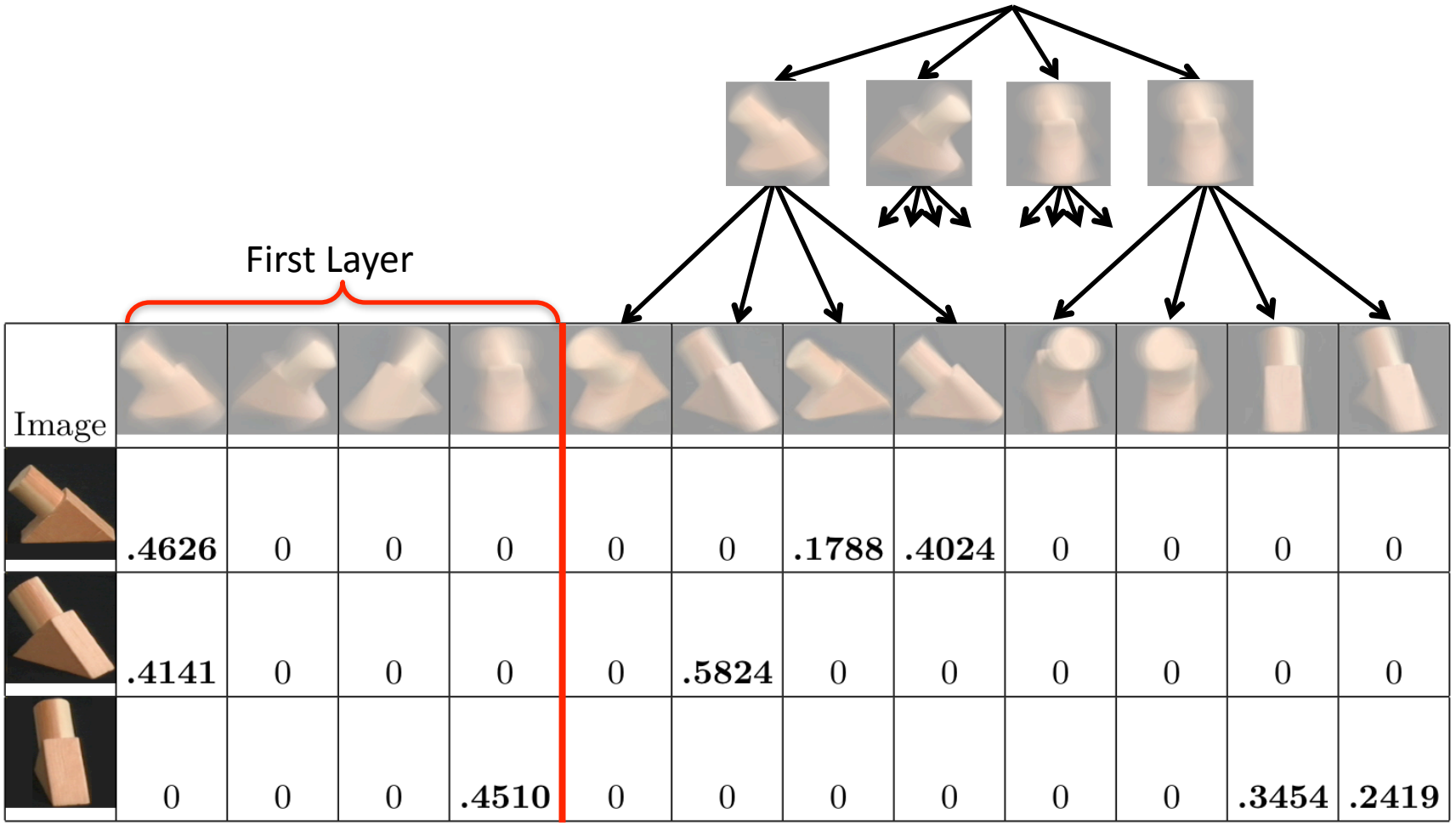
3. Hierarchical Dictionary

Efficient algorithm framework to learn tree structured multi-layer dictionaries.

Tree Structured Dictionaries



Tree Structured Dictionaries

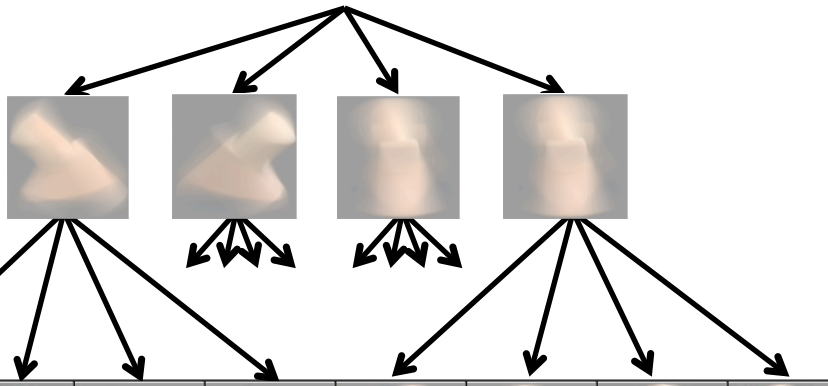


COIL data set, object #80. Train dictionaries using 72 images with rotation angle 5° apart.

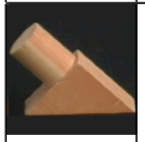
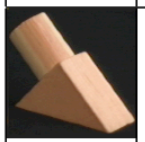
Tree Structured Dictionaries

Two layers better than one:

- Deep belief nets (Hinton et al., 2006).
- Deep coding network (Lin et al., 2010 NIPS).
- Proximal methods (Jenatton et al., 2010 ICML).



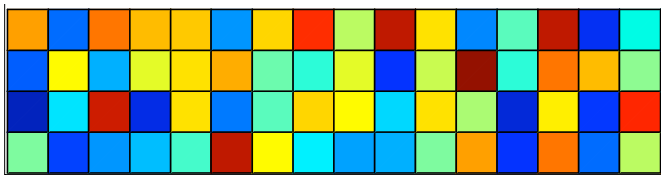
First Layer

Image											
	**.4626**	0	0	0	0	0	**.1788**	**.4024**	0	0	0	0
	**.4141**	0	0	0	0	**.5824**	0	0	0	0	0	0
	0	0	0	**.4510**	0	0	0	0	0	0	**.3454**	**.2419**

COIL data set, object #80. Train dictionaries using 72 images with rotation angle 5° apart.

Random Projections

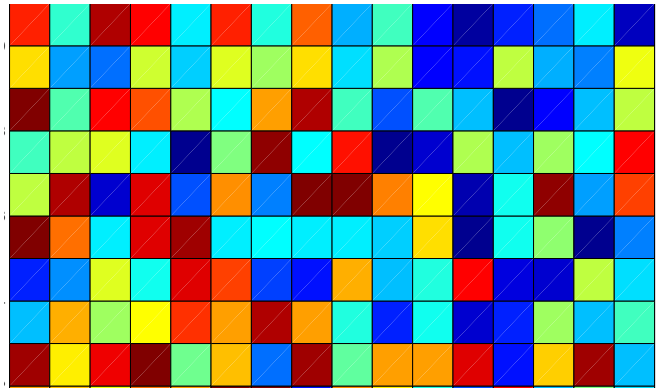
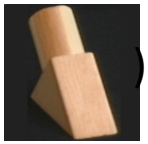
First layer uses fewer random projections:


× vector(

)

$$\mathbf{T}_1\mathbf{X} \approx \mathbf{B}_1\mathbf{W}_1$$

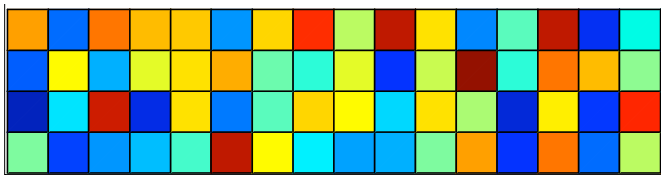
Second layer uses more random projections:


× vector(

)

$$\mathbf{T}_2\mathbf{X} \approx \mathbf{B}_2\mathbf{W}_2$$

Random Projections

First layer uses fewer random projections:

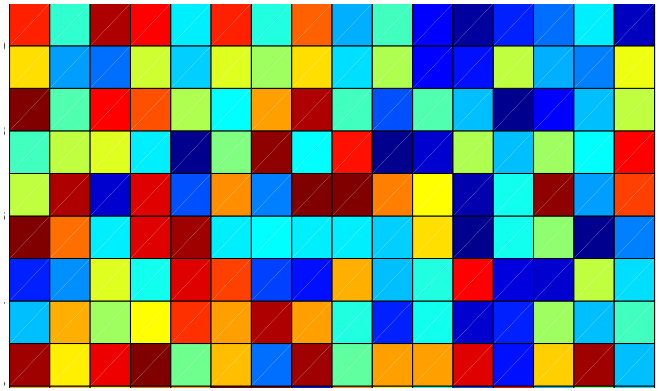


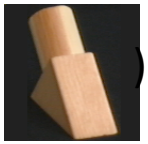
× vector( )

$$\mathbf{T}_1\mathbf{X} \approx \mathbf{B}_1\mathbf{W}_1$$

Use incremental, orthogonal random projections to control the information flow.

Second layer uses more random projections:

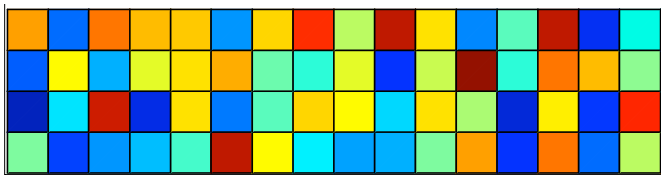


× vector()

$$\mathbf{T}_2\mathbf{X} \approx \mathbf{B}_2\mathbf{W}_2$$

Random Projections

First layer uses fewer random projections:



× vector(

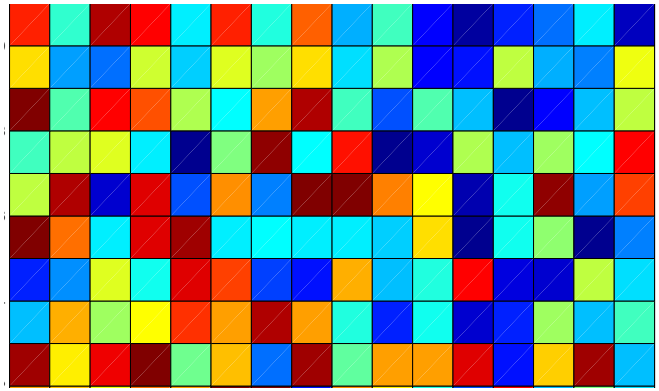
)

$$\mathbf{T}_1\mathbf{X} \approx \mathbf{B}_1\mathbf{W}_1$$

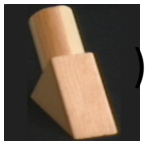
Use incremental, orthogonal random projections to control the information flow.

Previous layers inform the coding of later layers.

Second layer uses more random projections:



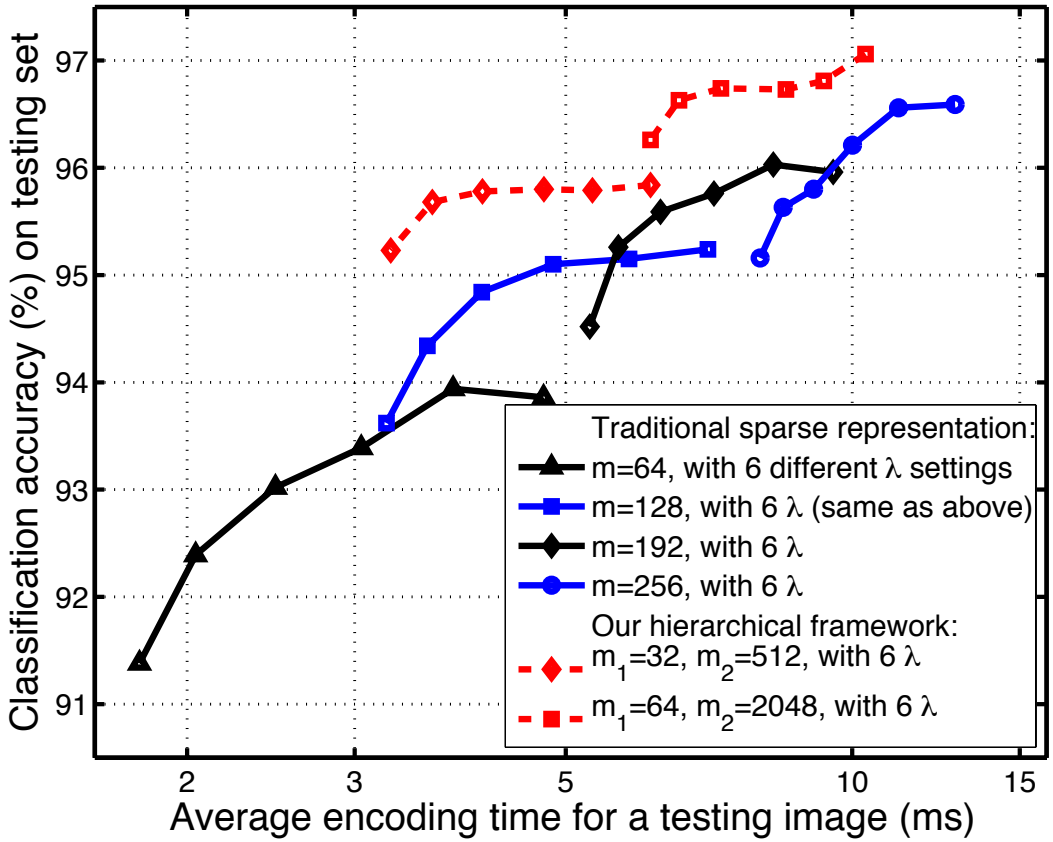
× vector(



)

$$\mathbf{T}_2\mathbf{X} \approx \mathbf{B}_2\mathbf{W}_2$$

Evaluation: Time and Quality



- Hand written digit images in MNIST data set, dimension = 784 (28x28), *m* is the dictionary size;
- Use the sparse coding solvers in (Lee et al., 2006 NIPS) (MatLab) to perform basic optimizations;
- Use liblinear (Fan et al., 2008 JMLR) (C/C++) classifier on the sparse representation weights;
- Ran on an Intel Xeon X5570 2.93GHz processor.

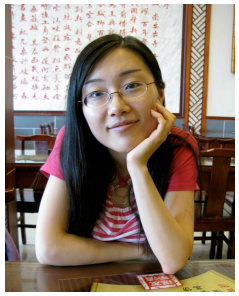
Conclusion

- Learning sparse representation is computationally challenging, but there is hope.
- Lasso screening test significantly speeds up lasso at virtually no additional cost.
- Hierarchical dictionaries and random projections make learning sparse representation more efficient.

Supplemental material and Matlab toolbox available on my website:

qr.net/sparse

Acknowledgements



Hao Xu
Princeton University

Peter J. Ramadge
Princeton University

Kai Yu
NEC Labs America Inc.

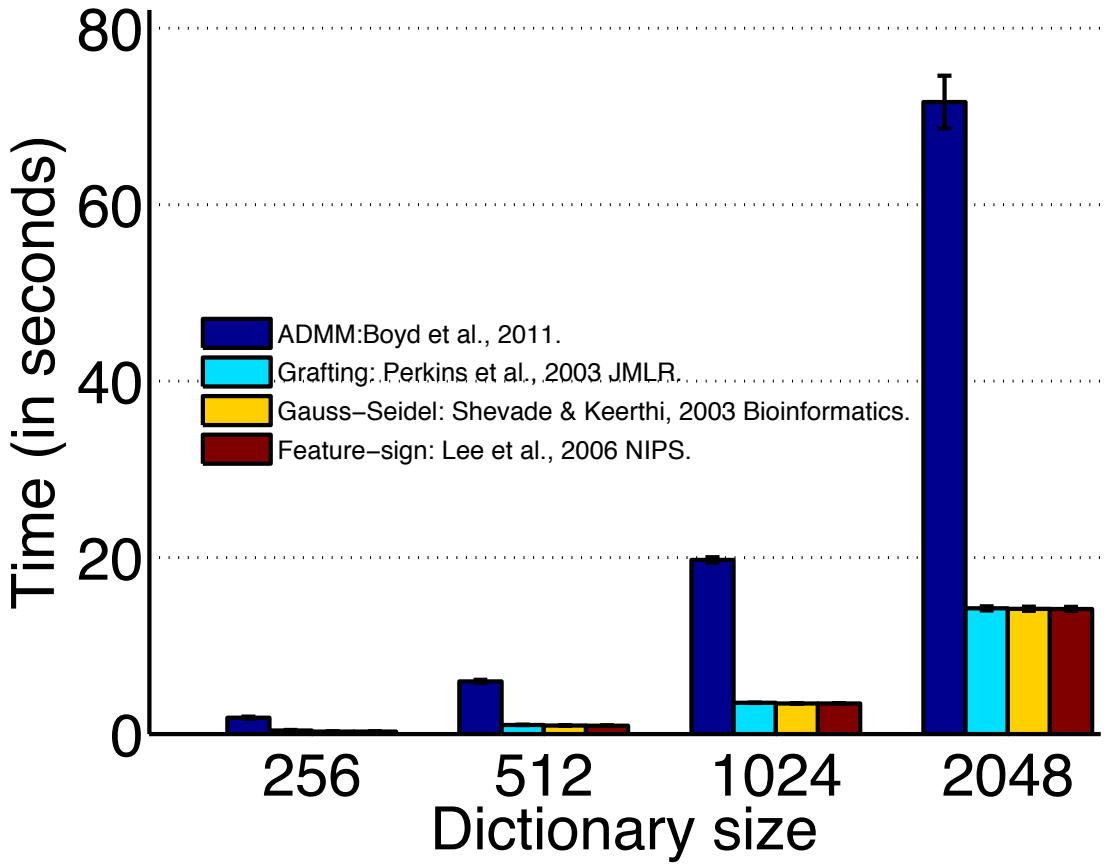
Tong Zhang
Rutgers University

Charlotte Elizabeth Procter Honorific Fellowship
Princeton University

Grant CCF-1116208
National Science Foundation

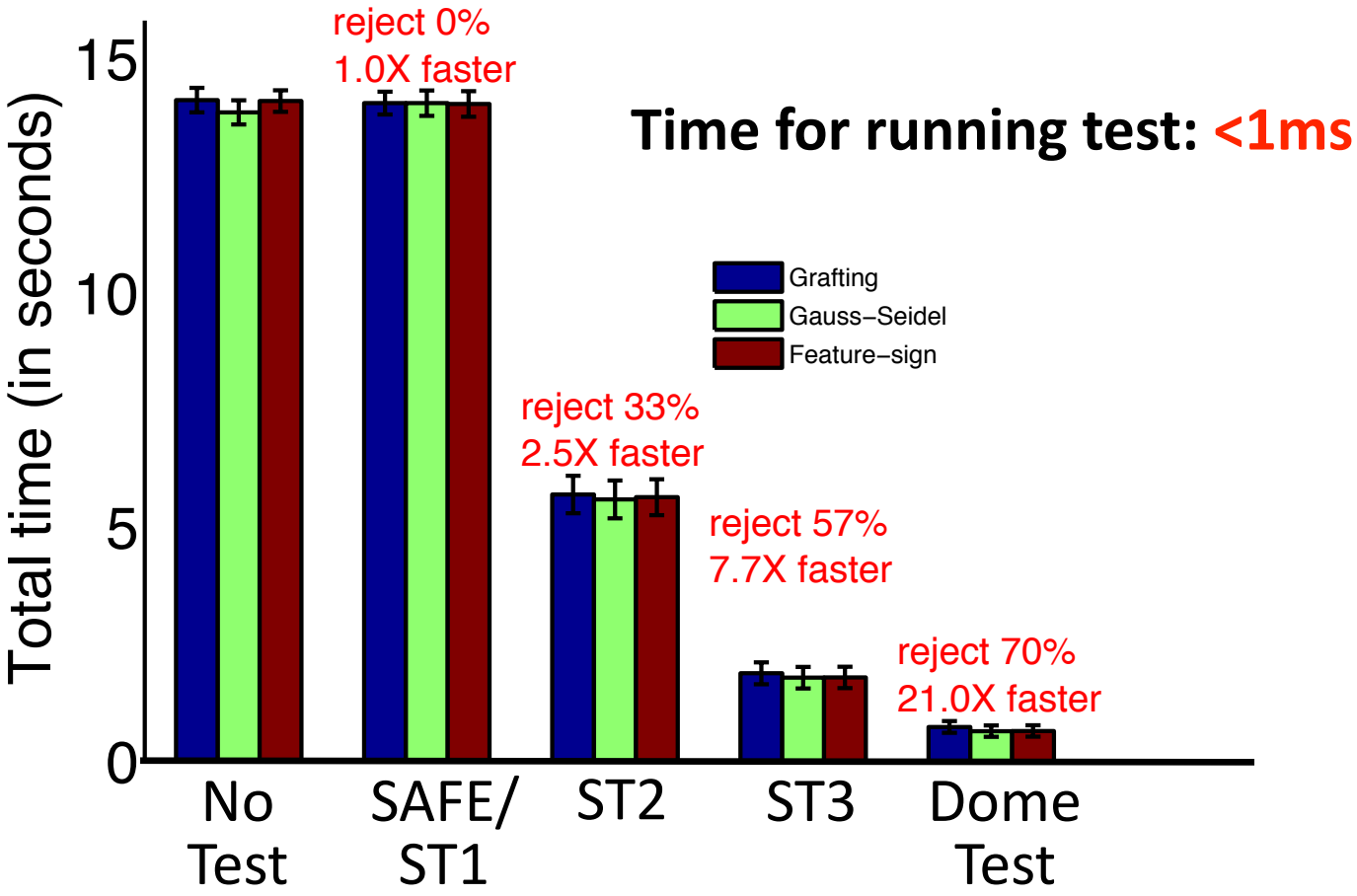
See you at the poster session! Poster T036

Solving One Lasso Problem



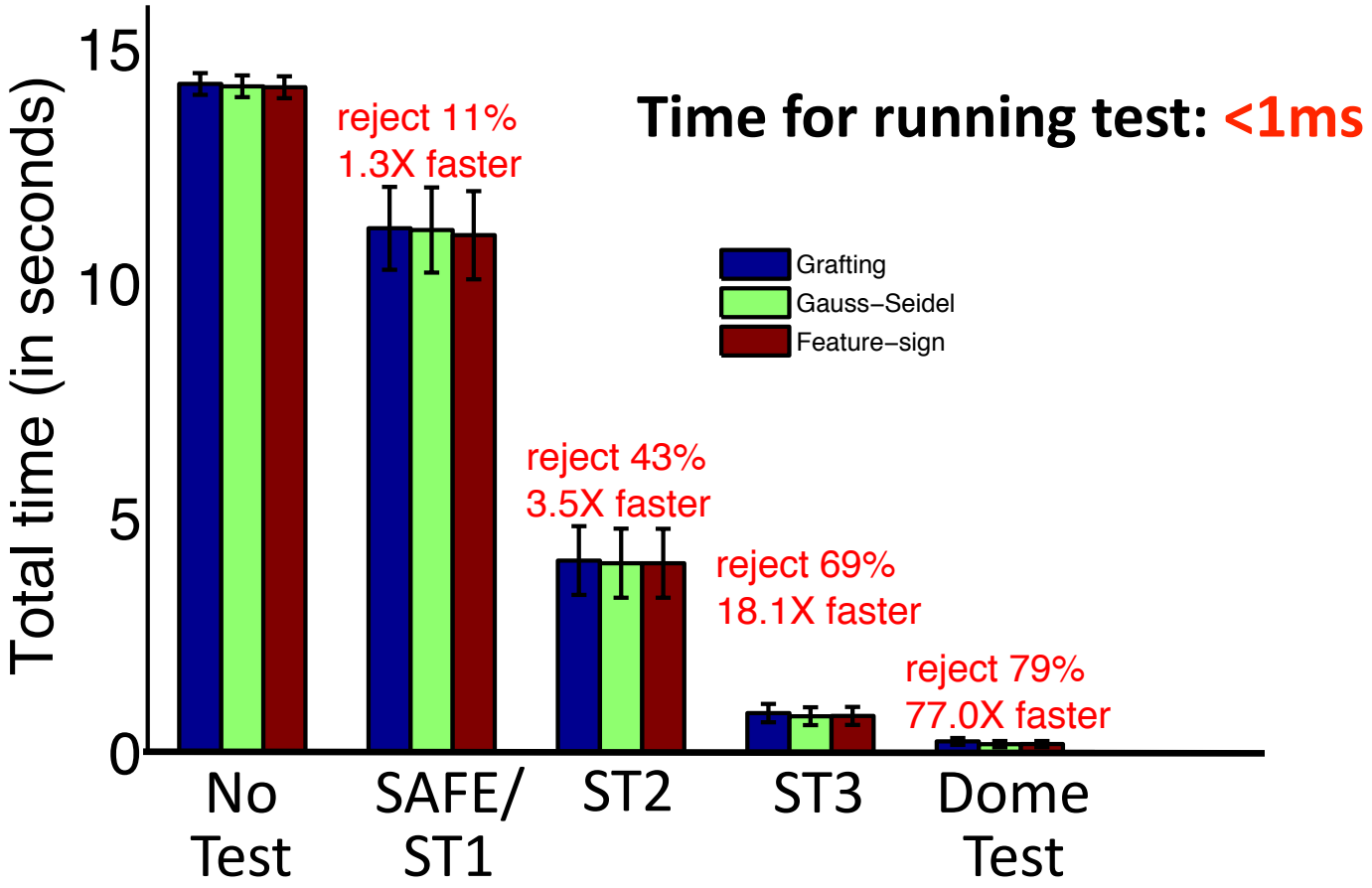
- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.6;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Speed Up Lasso



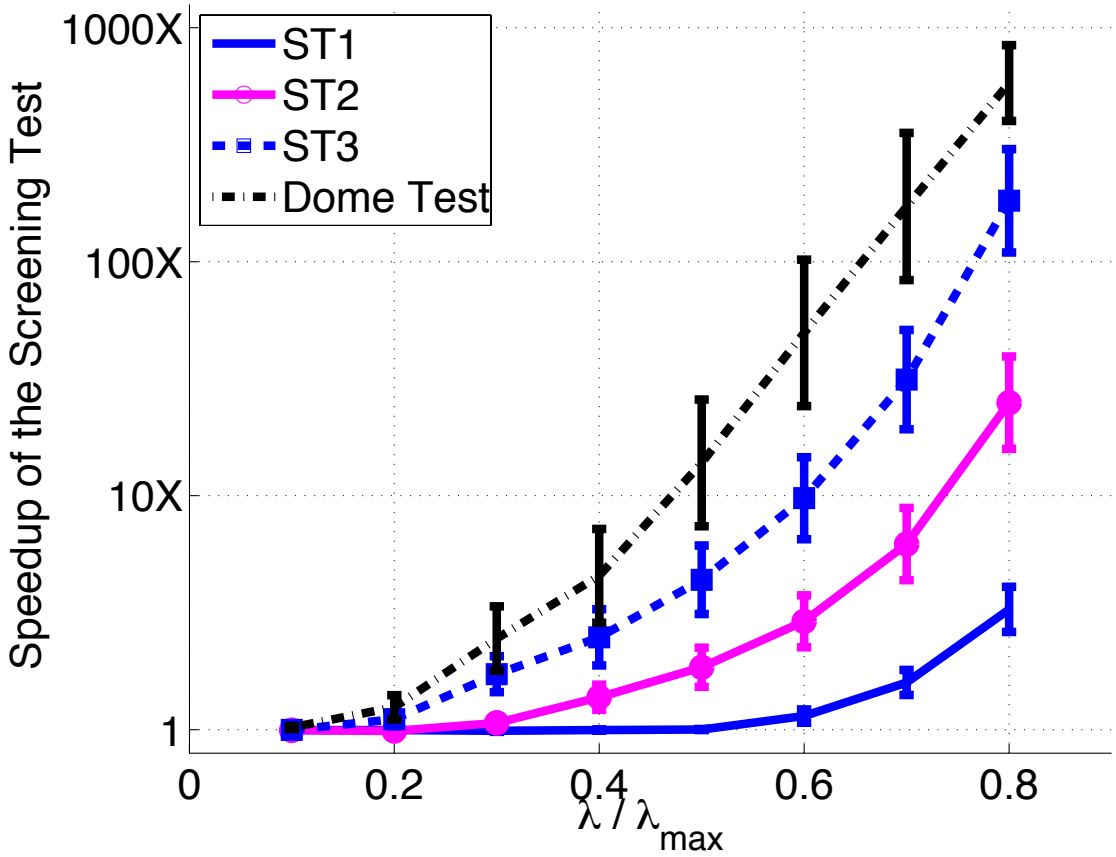
- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.5; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Speed Up Lasso



- Frontal faces in Yale B Extended data set, dimension = 32256, normalized, λ=0.6; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Speed Up Lasso

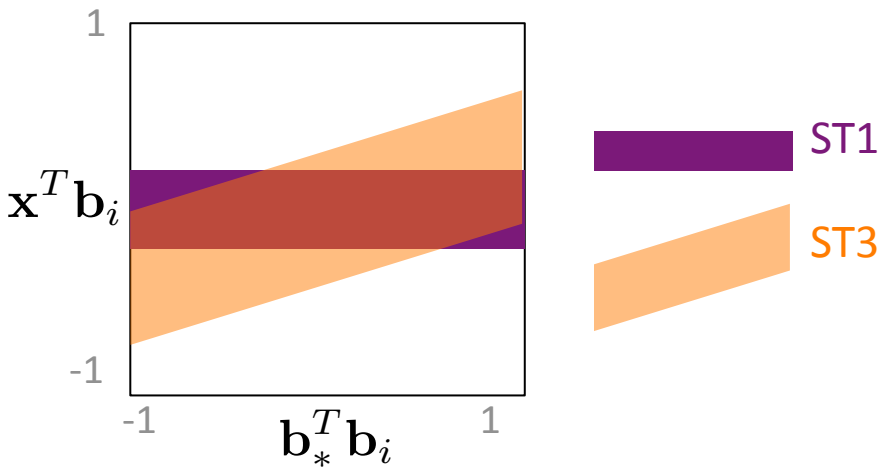


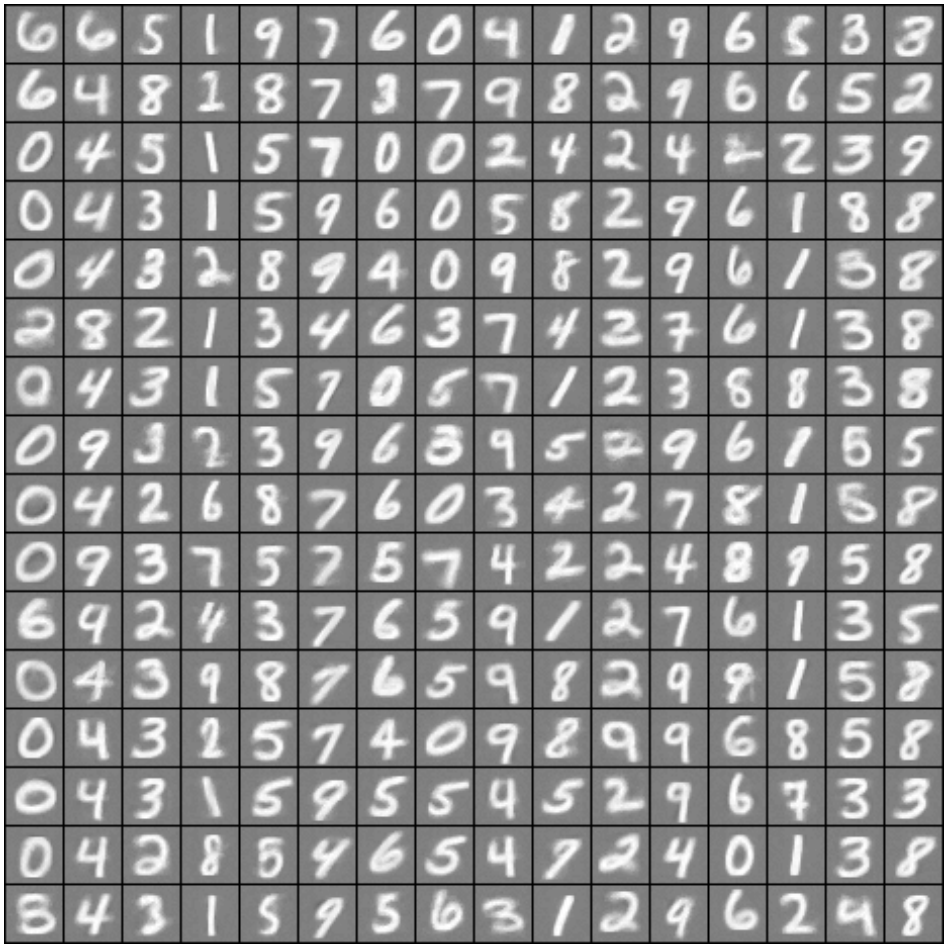
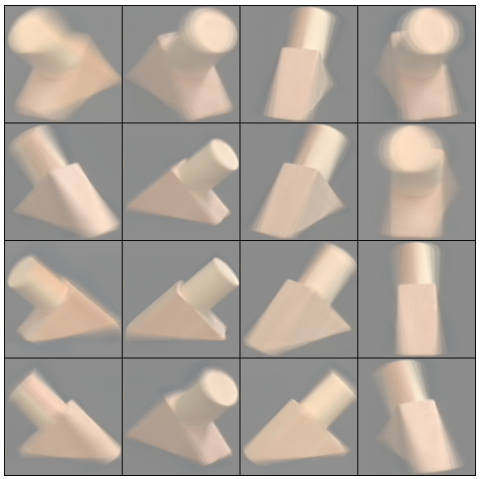
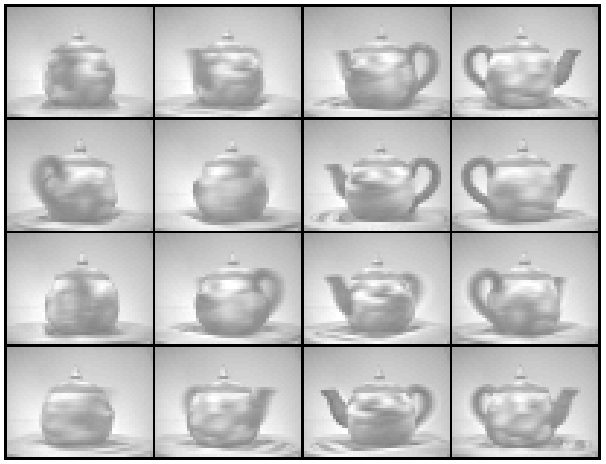
- Frontal faces in Yale B Extended data set, dimension = 32256, normalized; dictionary size: 2048;
- Reporting average time and standard error over 10 randomly chosen lasso problems;
- Averaged over three solvers: Grating, Gauss-Seidel and Feature-sign
- MatLab implementation, ran on an Intel Xeon X5570 2.93GHz processor.

Fast and Online

- SAFE/ST1 (Ghaoui et al.): $|\mathbf{x}^T\mathbf{b}_i| < \lambda - 1 + \lambda/\lambda_{\max}$.
- ST2: $|\mathbf{x}^T\mathbf{b}_i| < \lambda_{\max}(1 - 2r_3)$.
- ST3: $|\mathbf{x}^T\mathbf{b}_i - (\lambda_{\max} - \lambda)\mathbf{b}_*^T\mathbf{b}_i| < \lambda(1 - r_3)$.

$$\lambda_{\max} = \max_i |\mathbf{x}^T\mathbf{b}_i|, \ \mathbf{b}_* = \arg\max_{\mathbf{b}\in\{\pm\mathbf{b}_i\}} \mathbf{x}^T\mathbf{b}, \quad r_3 = \sqrt{1/\lambda_{\max}^2 - 1}(\lambda_{\max}/\lambda - 1).$$



ST3 can replace ST2,
ST2 can replace ST1
when $\lambda_{\max} > \sqrt{3}/2 \approx 0.866$

More Organized Dictionaries