An Efficient Algorithm for a Class of Fused Lasso Problems

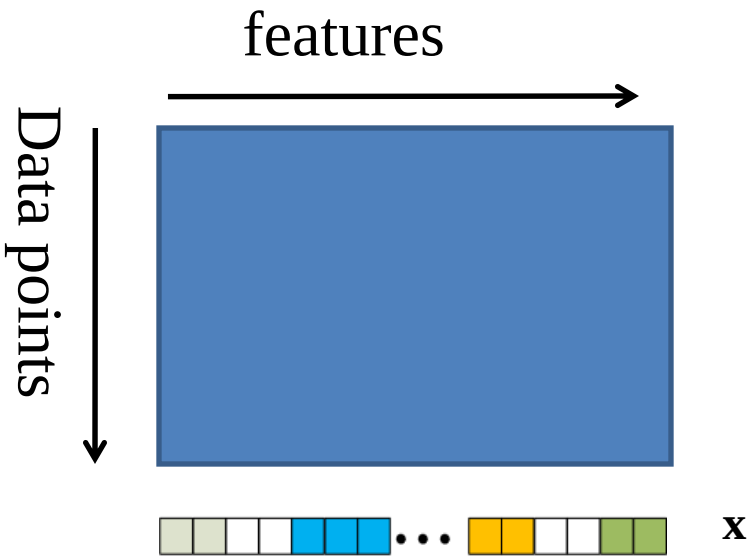
Jun Liu, Lei Yuan, and Jieping Ye

Computer Science and Engineering

The Biodesign Institute

Arizona State University

Sparse Learning for Feature Selection



$$\min \; \text{loss}(\mathbf{x}) + \text{penalty}(\mathbf{x})$$

Convex function defined on a set of training samples

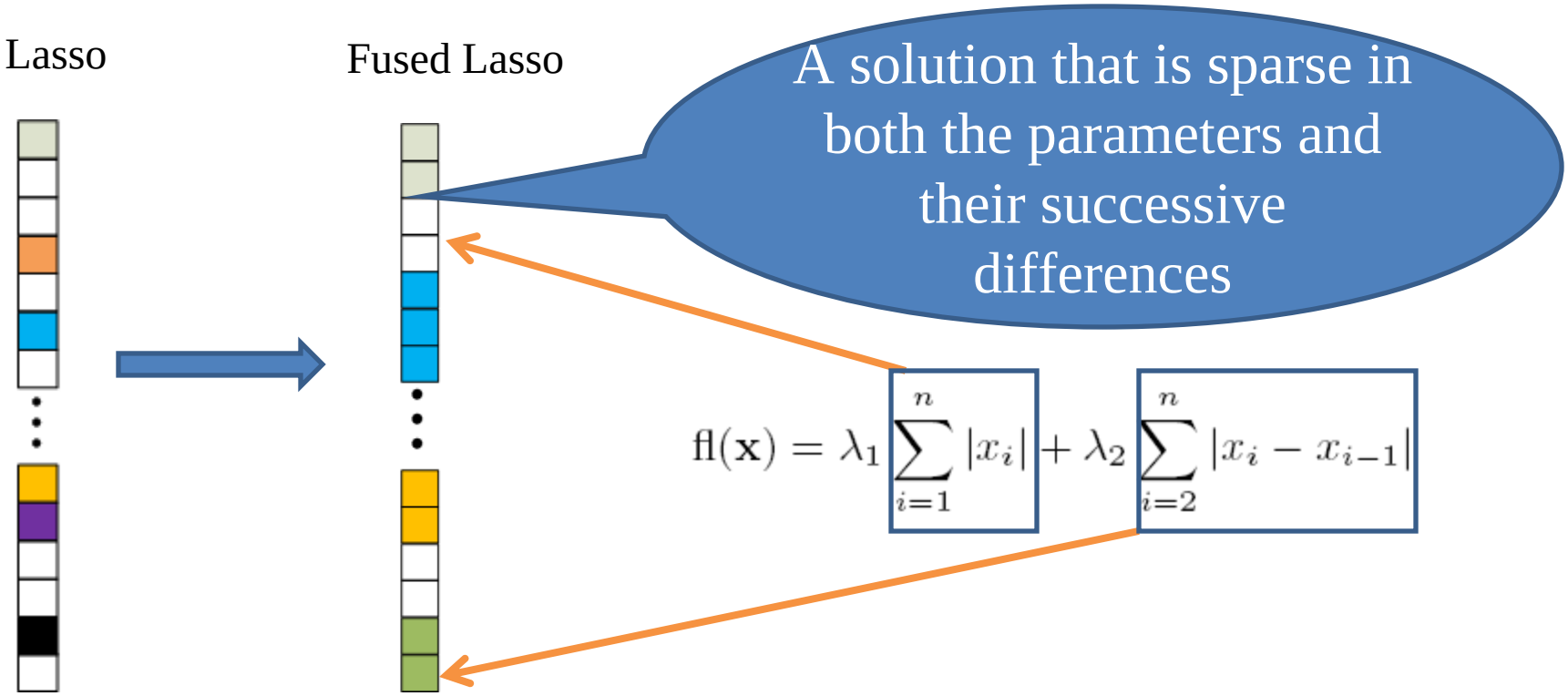
Our prior assumption on the parameter **x**

Outline

- **Fused Lasso and Applications**
- Proposed Algorithm
- Experiments
- Conclusion

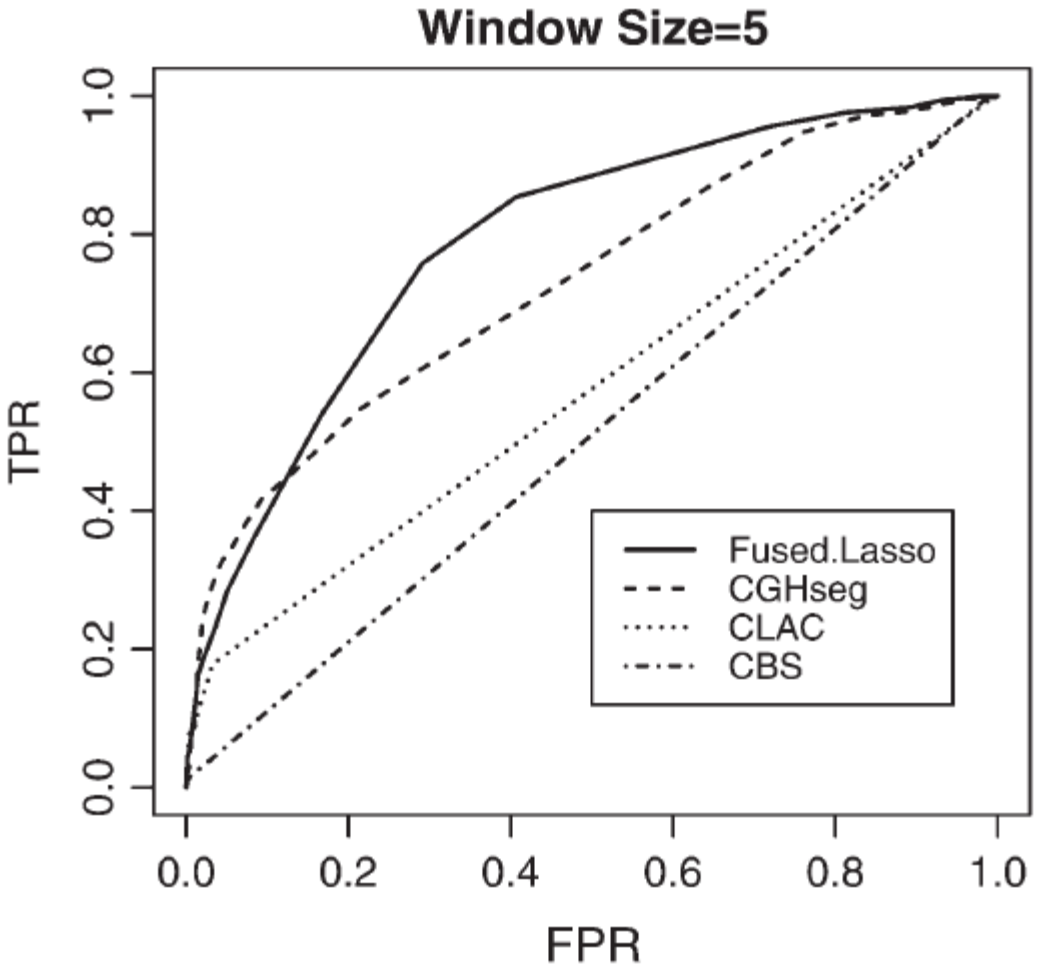
The Fused Lasso Penalty

(Tibshirani et al., 2005; Tibshirani and Wang, 2008; Friedman et al., 2007)



ArrayCGH Data

arrayCGH: array-based Comparative Genomic Hybridization
(Tibshirani and Wang, 2007)



ratio=

DNA copies of the gene in the tumor cells

DNA copies in the reference cells

piecewise constant shape of copy number changes

copy number alterations:
1. large chromosome segmentation gain/loss
2. abrupt local amplification/deletion

Unordered Data

leukaemia data (Tibshirani et al., 2005)

7129 genes 38 samples:
27 in class 1 mphocytic leukaemia)
11 in

Hierarchical clustering can be used to estimate an ordering of the genes, putting correlated genes near one another in the list.

(Tibshirani et al., 2005)

Outline

- Fused Lasso and Applications
- **Proposed Algorithm**
- Experiments
- Conclusion

The Fused Lasso Penalized Problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} h(\mathbf{x}) = \text{loss}(\mathbf{x}) + \text{fl}(\mathbf{x}),$$

Smooth convex loss functions:

- Least squares loss (Tibshirani, 2005)
- Logistic loss (Ahmed and Xing, 2009)

$$\text{fl}(\mathbf{x}) = \lambda_1 \sum_{i=1}^{n} |x_i| + \lambda_2 \sum_{i=2}^{n} |x_i - x_{i-1}|$$

non-smooth and non-separable

Smooth reformulation
(auxiliary variables and constraints)
+
general solver
(e.g., CVX)

"One difficulty in using the fused lasso is computational speed, …, speed could become a practical limitation. This is especially true if five or tenfold cross-validation is carried out."

(Tibshirani et al., 2005)

Efficient Fused Lasso Algorithm (EFLA)

$$\min_{\mathbf{x}\in\mathbb{R}^n} h(\mathbf{x}) = \underline{\text{loss}(\mathbf{x})} + \underline{\text{fl}(\mathbf{x})},$$

Smooth NonSmooth

Accelerated gradient descent (Nesterov, 2007; Beck and Teboulle, 2009), which has convergence rate of $O(1/k^2)$ for k iterations

$$\mathcal{M}(\mathbf{x}_i, \gamma_i) = [\text{loss}(\mathbf{x}_i) + \langle \text{loss}'(x_i), \mathbf{x} - \mathbf{x}_i \rangle] + \frac{1}{2\gamma_i}\|\mathbf{x} - \mathbf{x}_i\|_2^2 + \text{fl}(\mathbf{x})$$

FLSA is called in each iteration of EFLA

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ \vdots & & & \vdots \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}_{(n-1)\times n}$$

Fused Lasso Signal Approximator (FLSA, Friedman et al., 2007)

Subgradient Finding Algorithm (1)

Fused Lasso Signal Approximator (FLSA)

$$\pi_{\lambda_2}^{\lambda_1}(\mathbf{v}) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \lambda_1\|\mathbf{x}\|_1 + \lambda_2\|R\mathbf{x}\|_1$$

THEOREM 1. *For any* $\lambda_1, \lambda_2 \geq 0$, *we have*

$$\pi_{\lambda_2}^{\lambda_1}(\mathbf{v}) = \mathrm{sgn}(\pi_{\lambda_2}^{0}(\mathbf{v})) \odot \max(|\pi_{\lambda_2}^{0}(\mathbf{v})| - \lambda_1, 0).$$

$$\min_{\mathbf{x}\in\mathbb{R}^n} f_{\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \lambda_2\|R\mathbf{x}\|_1$$

simplified problem

Subgradient Finding Algorithm (2)

(P) $\min_{\mathbf{x}\in\mathbb{R}^n} f_{\lambda_2}(\mathbf{x}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \lambda_2\|R\mathbf{x}\|_1$

primal

$$\min_{\mathbf{x}\in\mathbb{R}^n} \max_{\|\mathbf{z}\|_\infty\le\lambda_2} \phi(\mathbf{x},\mathbf{z}) \equiv \frac{1}{2}\|\mathbf{x}-\mathbf{v}\|^2 + \langle R\mathbf{x},\mathbf{z}\rangle.$$

$$\mathbf{x} = \mathbf{v} - R^{\mathrm{T}}\mathbf{z}.$$

relationship

(D) $\min_{\|\mathbf{z}\|_\infty\le\lambda_2} \psi(\mathbf{z}) \equiv -\phi(\mathbf{v}-R^{\mathrm{T}}\mathbf{z},\mathbf{z}) = \frac{1}{2}\|R^{\mathrm{T}}\mathbf{z}\|^2 - \langle R^{\mathrm{T}}\mathbf{z},\mathbf{v}\rangle.$

dual

THEOREM 3. Let $\tilde{\mathbf{z}}$ be an appropriate solution. Let $\tilde{\mathbf{x}} = \mathbf{v} - R^{\mathrm{T}}\tilde{\mathbf{z}}$

$$\mathrm{gap}(\tilde{\mathbf{x}},\tilde{\mathbf{z}}) = \max_{\mathbf{z}:\|\mathbf{z}\|_\infty\le\lambda_2} \phi(\tilde{\mathbf{x}},\mathbf{z}) - \min_{\mathbf{x}} \phi(\mathbf{x},\tilde{\mathbf{z}}).$$

$$\psi(\tilde{\mathbf{z}}) - \psi(\mathbf{z}^*) \le \mathrm{gap}(\tilde{\mathbf{x}},\tilde{\mathbf{z}}),$$

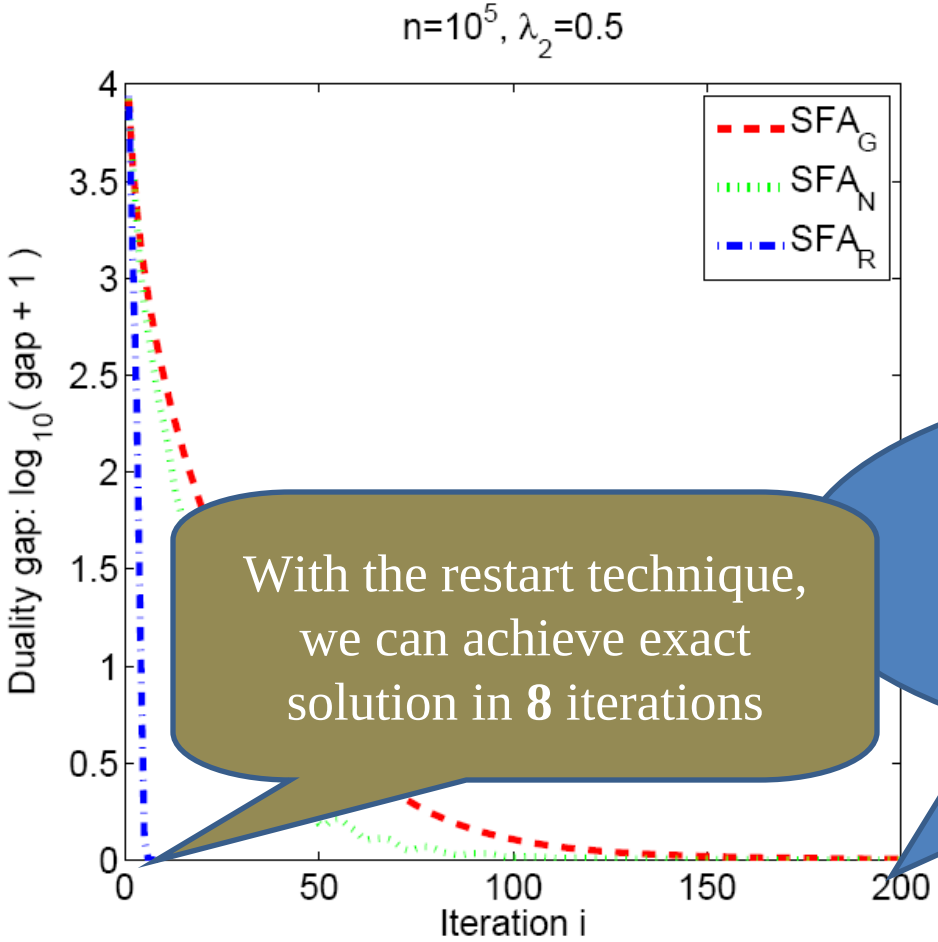
$$f_{\lambda_2}(\tilde{\mathbf{x}}) - f_{\lambda_2}(\mathbf{x}^*) \le \mathrm{gap}(\tilde{\mathbf{x}},\tilde{\mathbf{z}}).$$

duality gap

Subgradient Finding Algorithm (3)

(D) $$\min_{\|\mathbf{z}\|_\infty \le \lambda_2} \psi(\mathbf{z}) \equiv -\phi(\mathbf{v} - R^{\mathrm{T}}\mathbf{z}, \mathbf{z}) = \frac{1}{2}\|R^{\mathrm{T}}\mathbf{z}\|^2 - \langle R^{\mathrm{T}}\mathbf{z}, \mathbf{v}\rangle.$$ dual

v is of dimensionality $n=10^5$. Its entries are from the standard normal distribution.



With the restart technique, we can achieve exact solution in **8** iterations

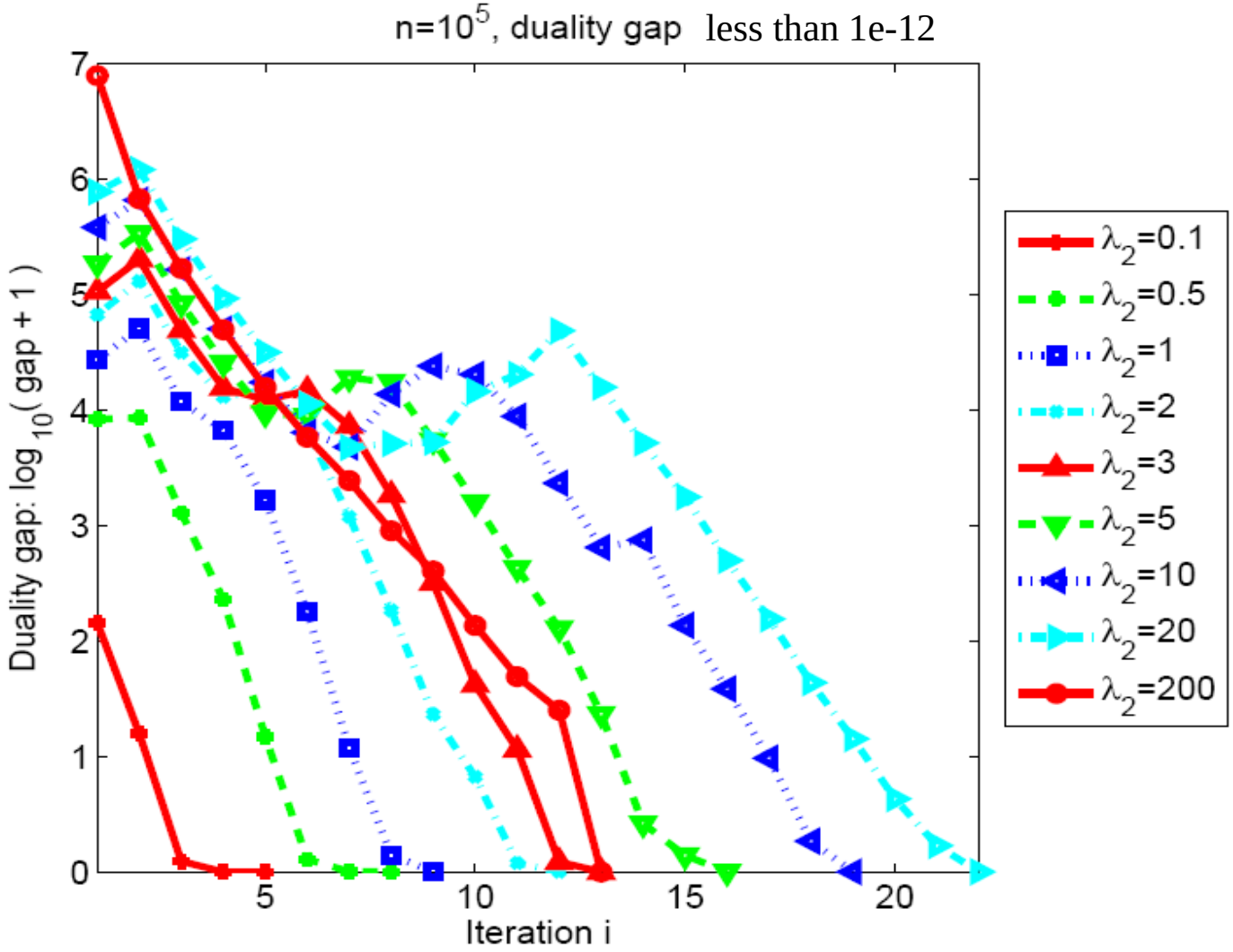
SFA_G and SFA_N achieve duality gaps of 1e-2 and 1e-3 in 200 iterations, respectively.

SFA_G: SFA via Gradient Descent
SFA_N: SFA via Nesterov's method
SFA_R: SFA via the restart technique

Outline

- Fused Lasso and Applications
- Proposed Algorithm
- **Experiments**
- Conclusion

Illustration of SFA



Efficiency of SFA

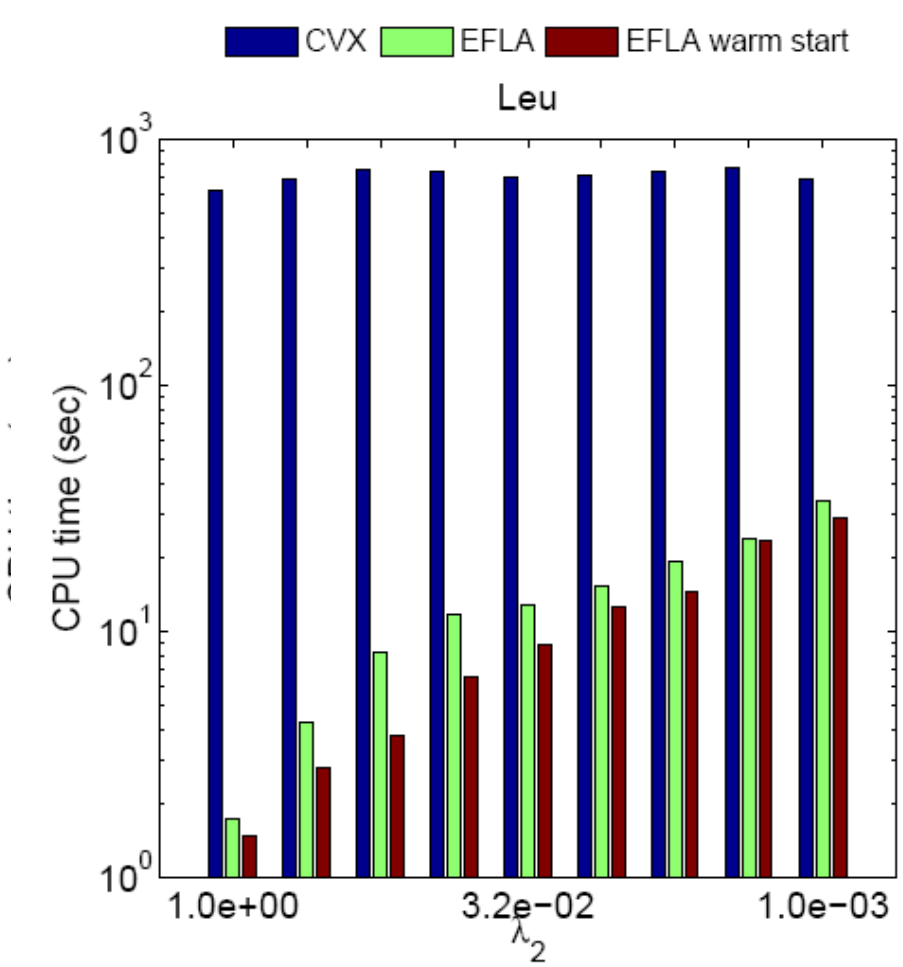
$\lambda_2 = r \times \lambda_2^{\max}$

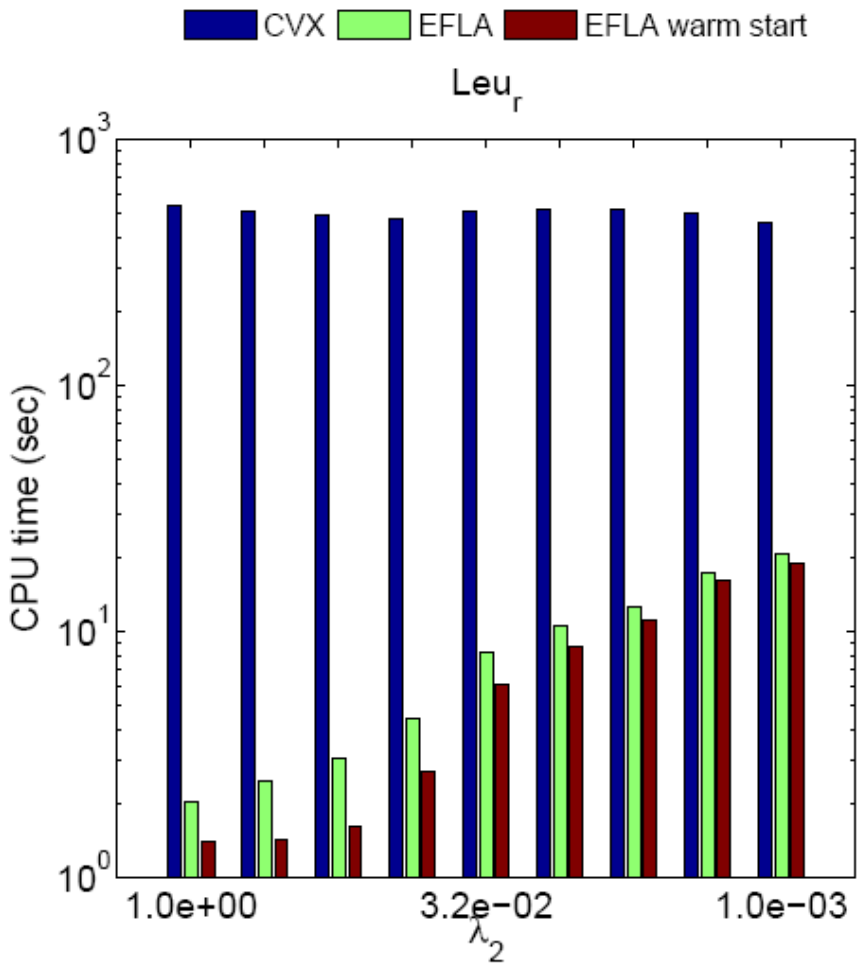
	n	10^2	10^3	10^4	10^5	10^6	10^7
10^{-3}	SFA_R Iteration ($\|S\|$)	1 (98)	2 (969)	3 (9025)	7 (70058)	10 (324617)	15 (782902)
	SFA_R Time	2.9×10^{-5}	1.6×10^{-4}	2.1×10^{-3}	4.8×10^{-2}	0.72	9.2
	pathFLSA Time	5.9×10^{-4}	6.1×10^{-3}	6.2×10^{-2}	0.76	9.6	-
10^{-2}	SFA_R Iteration ($\|S\|$)	2 (90)	4 (714)	9 (3363)	14 (7394)	20 (10789)	29 (13092)
	SFA_R Time	3.0×10^{-5}	2.6×10^{-4}	4.2×10^{-3}	5.9×10^{-2}	1.1	14
	pathFLSA Time	6.1×10^{-4}	6.1×10^{-3}	6.1×10^{-2}	0.76	9.6	-
10^{-1}	SFA_R Iteration ($\|S\|$)	5 (34)	9 (85)	16 (113)	24 (119)	32 (127)	42 (139)
	SFA_R Time	4.1×10^{-5}	3.5×10^{-4}	5.1×10^{-3}	8.7×10^{-2}	1.5	20
	pathFLSA Time	6.4×10^{-4}	6.1×10^{-3}	6.1×10^{-2}	0.76	9.7	-
1	SFA_R Iteration ($\|S\|$)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
	SFA_R Time	1.7×10^{-5}	4.8×10^{-5}	3.5×10^{-4}	5.0×10^{-3}	5.4×10^{-2}	0.53
	pathFLSA Time	4.8×10^{-4}	6.1×10^{-3}	6.2×10^{-2}	0.77	9.7	-

Much more efficient than pathFLSA (over 10 times for most cases)

Efficiency of EFLA

(Comparison with the CVX solver)





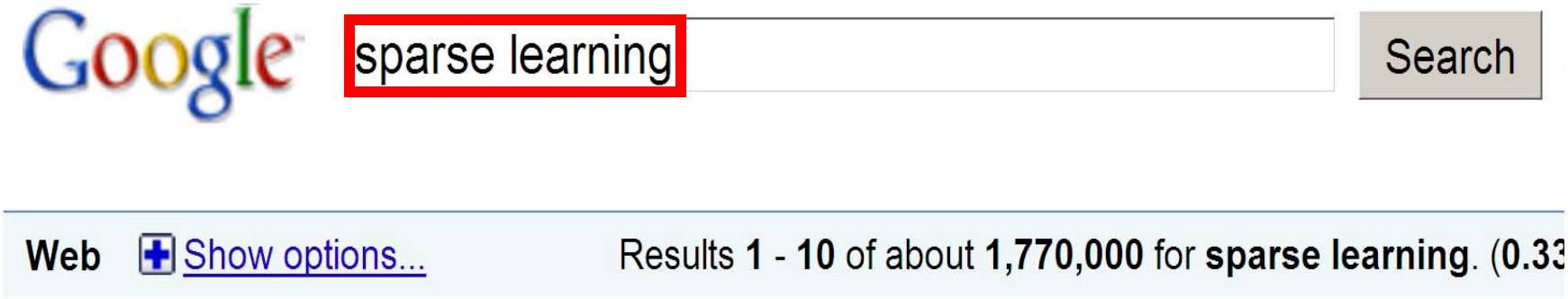
Classification Performance

	Fused Lasso	Lasso
Array CGH	88%	82%
Prostate Cancer	98%	98%
Leukemias	96%	94%
Leukemias Reordered	97%	94%

- Significant performance improvement on Array CGH
- Comparable results on the prostate cancer data set
- Hierarchical clustering can help improve the performance

SLEP: Sparse Learning with Efficient Projections

Liu, Ji, and Ye (2009) SLEP: A Sparse Learning Package
http://www.public.asu.edu/~jye02/Software/SLEP/



Conclusion and Future work

Contributions:

- An efficient algorithm for the class of fused Lasso problem
- Subgradient finding algorithm with a novel restart technique

Future work:

- Extend the algorithm to the multi-dimensional fused Lasso
- Apply the proposed algorithm for learning time-varying network

Thank you!