FAST AND ADAPTIVE ADMM

Tom Goldstein

COMPUTER SCIENCE
UNIVERSITY OF MARYLAND

CONSTRAINED PROBLEMS

$$\begin{array}{ll} \text{minimize} & H(u) + G(v) \\ \text{subject to} & Au + Bv = b \end{array}$$

Big idea: Lagrange multipliers

$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \langle \lambda, b - Au - Bv \rangle + \frac{\tau}{2} \|b - Au - Bv\|^2$$

- Optimality for λ: $b - Au - Bv = 0$
- Reduced energy: $H(u) + G(v)$
- Saddle-point = Solution to constrained problem

ADMM

$$\begin{array}{ll} \text{minimize} & H(u) + G(v) \\ \text{subject to} & Au + Bv = b \end{array}$$

Big Idea: Lagrange multipliers

$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \langle \lambda, b - Au - Bv \rangle + \frac{\tau}{2} \|b - Au - Bv\|^2$$

Alternating Direction Method of Multipliers

$$u_{k+1} = \arg\min_{u} H(u) + \langle \lambda_k, -Au \rangle + \frac{\tau}{2} \|b - Au - Bv_k\|^2$$

$$v_{k+1} = \arg\min_{v} G(v) + \langle \lambda_k, -Bv \rangle + \frac{\tau}{2} \|b - Au_{k+1} - Bv\|^2$$

$$\lambda_{k+1} = \lambda_k + \tau(b - Au_{k+1} - Bv_{k+1})$$

DISTRIBUTED PROBLEMS

machine learning = model fitting

- data is stored across many servers
- datasets are big (memory is an issue)
- communication is expensive

Google compute engine

DoD Supercomputing Resource Center

…use ADMM

DISTRIBUTED PROBLEMS

$$\text{minimize} \quad g(x) + \sum_i f_i(x)$$

example: sparse least squares

$$\text{minimize} \quad \mu|x| + \frac{1}{2}\|Ax - b\|^2$$

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{pmatrix}$$

$$\text{minimize} \quad \mu|x| + \sum_i \frac{1}{2}\|A_i x - b_i\|^2$$

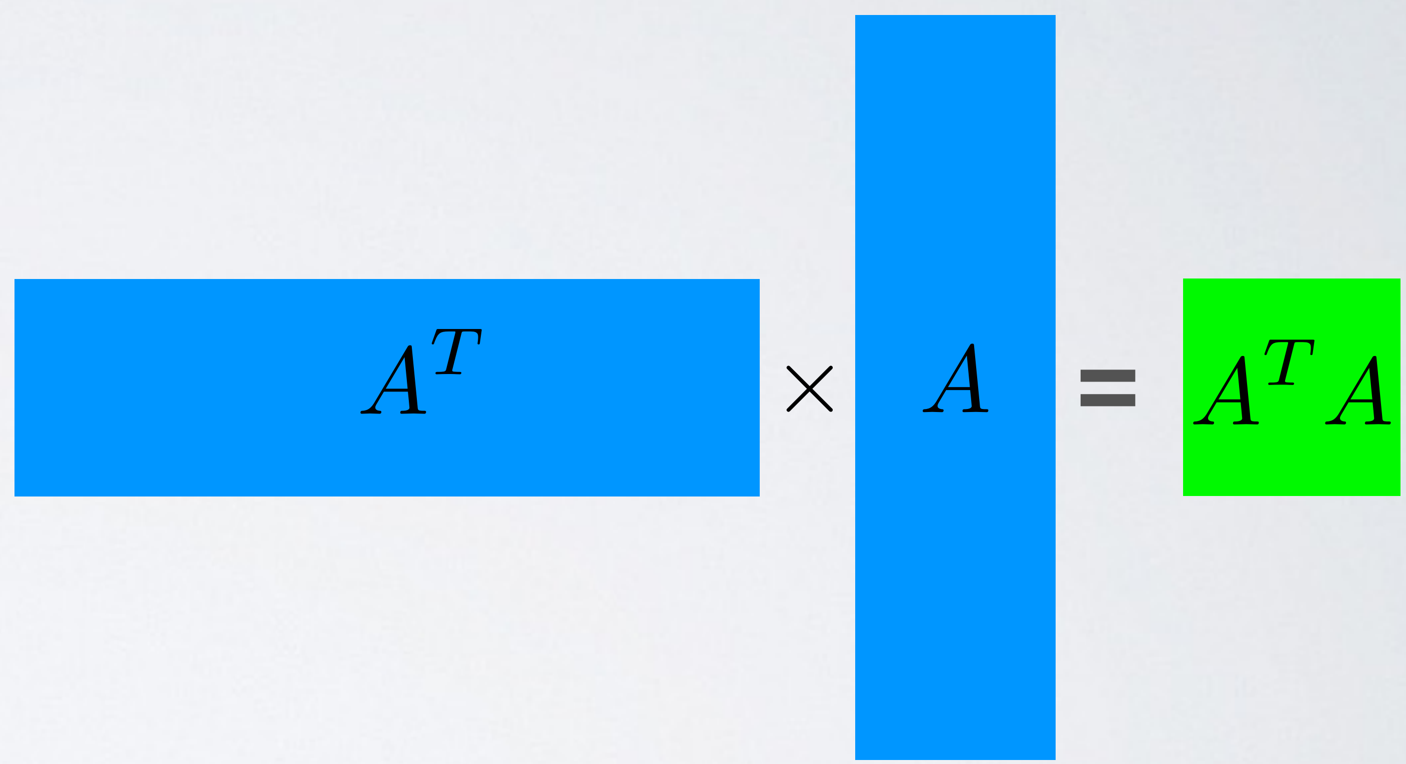
data stored on different servers

TRANSPOSE REDUCTION

minimize $\quad \frac{1}{2}\|Ax - b\|^2$

$$\underline{(A^T A)}^{-1} A^T b$$

normal equations



TRANSPOSE REDUCTION

$$\text{minimize} \quad \frac{1}{2}\|Ax - b\|^2$$

$$\underline{(A^T A)}^{-1} A^T b$$

normal equations

$$A = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_N \end{pmatrix}$$

distributed computation

$$A^T b = \sum A_i b_i$$

$$A^T A = \sum A_i^T A_i$$

Big idea

Solve complex problems with ADMM,
solve least-squares sub-problems with TR

UNWRAPPED ADMM

$$\text{minimize} \quad g(x) + f(Ax) = g(x) + \sum_i f_i(A_i x)$$

Example: SVM

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + h(Ax)$$

$$A = \text{data}, \quad h = \text{hinge loss}$$

UNWRAPPED ADMM

$$\text{minimize} \quad g(x) + f(Ax) = g(x) + \sum_i f_i(A_i x)$$

Example: SVM

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + h(Ax)$$

"unwrapped" form

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|x\|^2 + h(z) \\ \text{subject to} \quad & z = Ax \end{aligned}$$

TRANSPOSE REDUCTION ADMM

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + h(Ax)$$

"unwrapped" form

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|x\|^2 + h(z) \\ \text{subject to} \quad & z = Ax \end{aligned}$$

scaled augmented Lagrangian

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + h(z) + \frac{\tau}{2}\|z - Ax + \lambda\|^2$$

TRANSPOSE REDUCTION ADMM

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + h(Ax)$$

scaled augmented Lagrangian

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + h(z) + \frac{\tau}{2}\|z - Ax + \lambda\|^2$$

ADMM

$$x^{k+1} = \min_x \frac{1}{2}\|x\|^2 + \frac{\tau}{2}\|z^k - Ax + \lambda^k\|^2$$

$$z^{k+1} = \min_z h(z) + \frac{\tau}{2}\|z - Ax^{k+1} + \lambda^k\|^2$$

$$\lambda^{k+1} = \lambda^k + z^{k+1} - Ax^{k+1}$$

Least squares: use TR here

GLOBAL UPDATE

scaled augmented Lagrangian

$$\text{minimize} \quad \frac{1}{2}\|x\|^2 + \sum_i h(z_i) + \frac{\tau}{2}\|z_i - A_i x + \lambda_i\|^2$$

central servers: x-update

$$x^{k+1} = \min_x \frac{1}{2}\|x\|^2 + \frac{\tau}{2}\|z^k - Ax + \lambda^k\|^2$$

solution

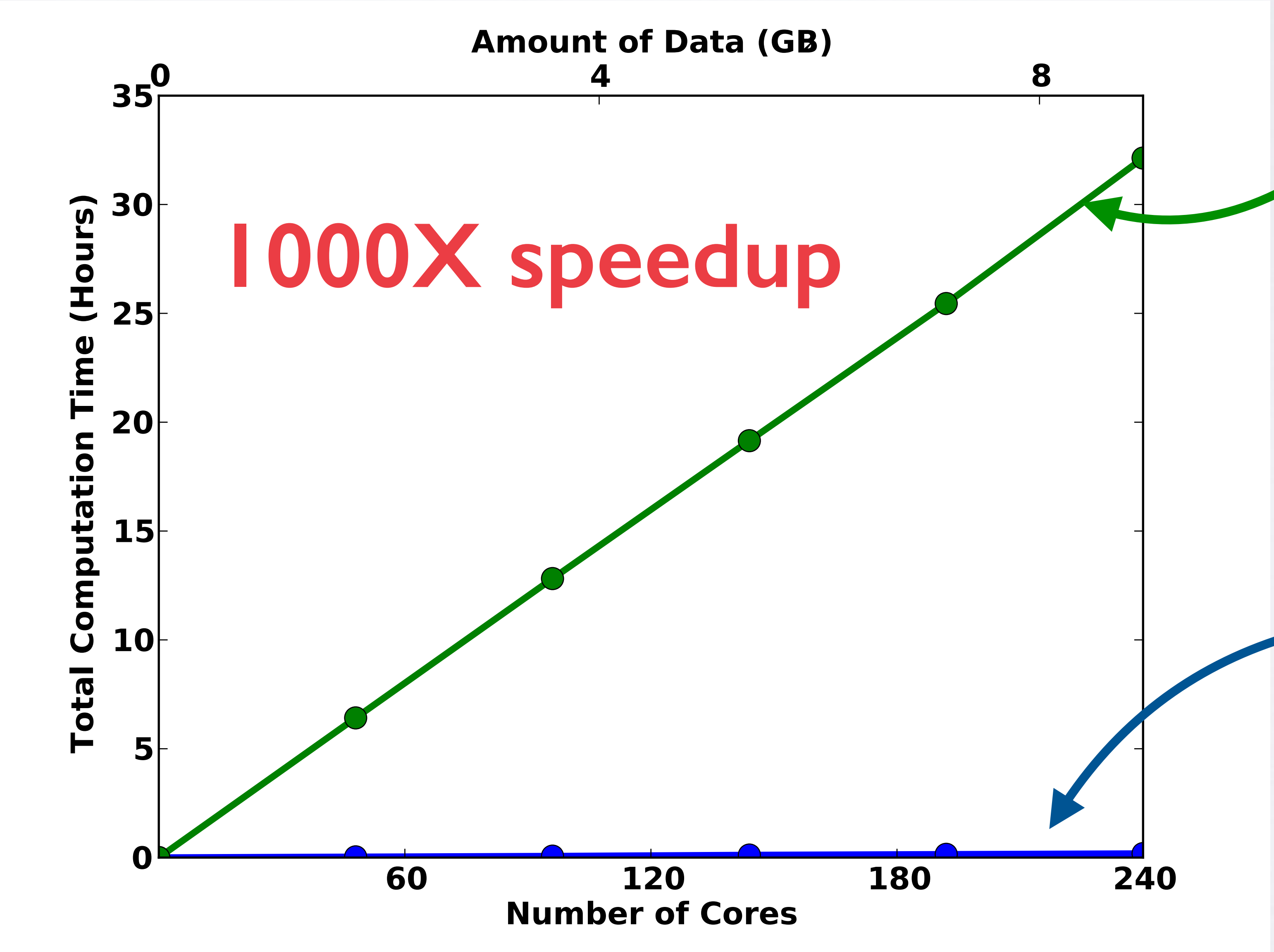
$$x^{k+1} = (A^T A)^{-1} \sum_i \underline{A_i^T (z_i^{k+1} + \lambda_i^k)}$$

pre-computed once

computed in the cloud on each iteration

EXPERIMENT: SVM

2K features, 50K data points/core



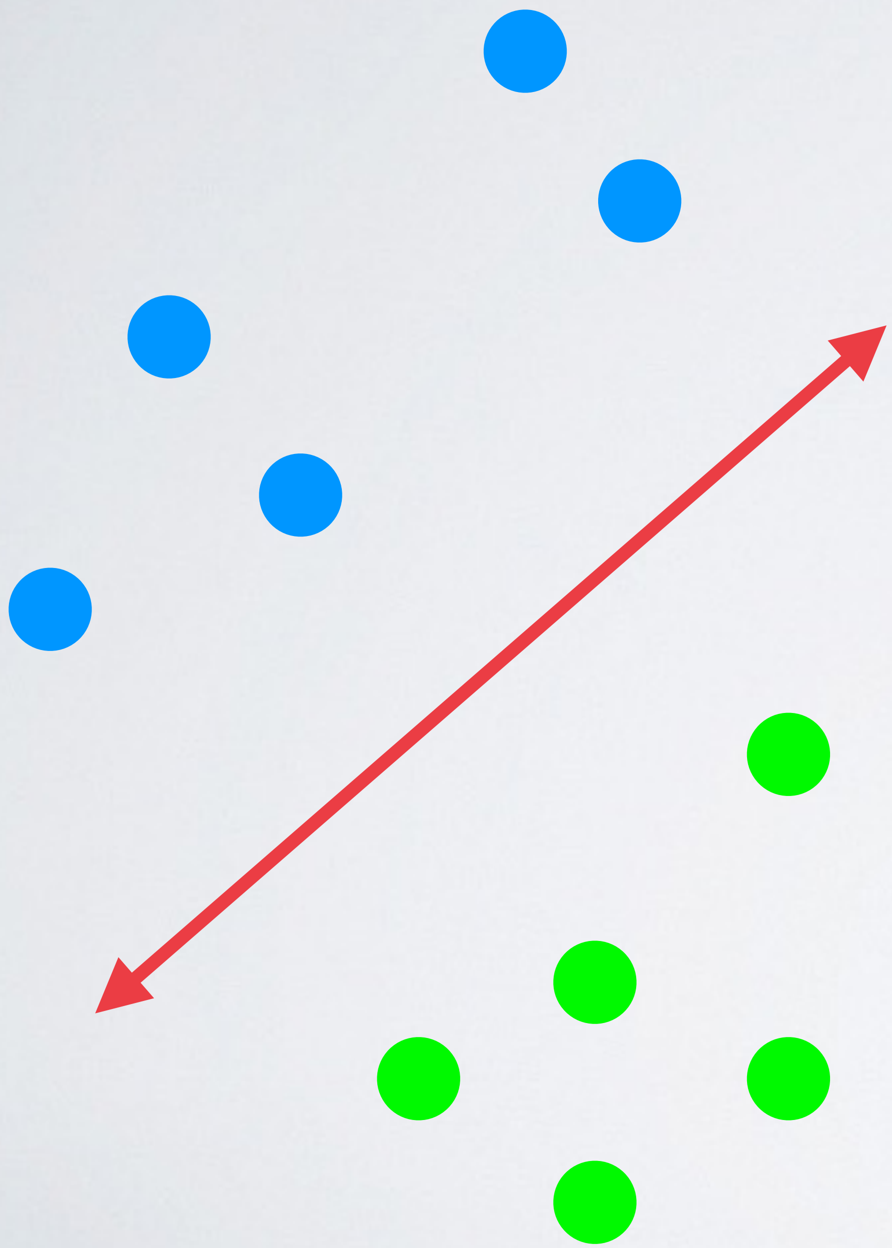
consensus

transpose reduction

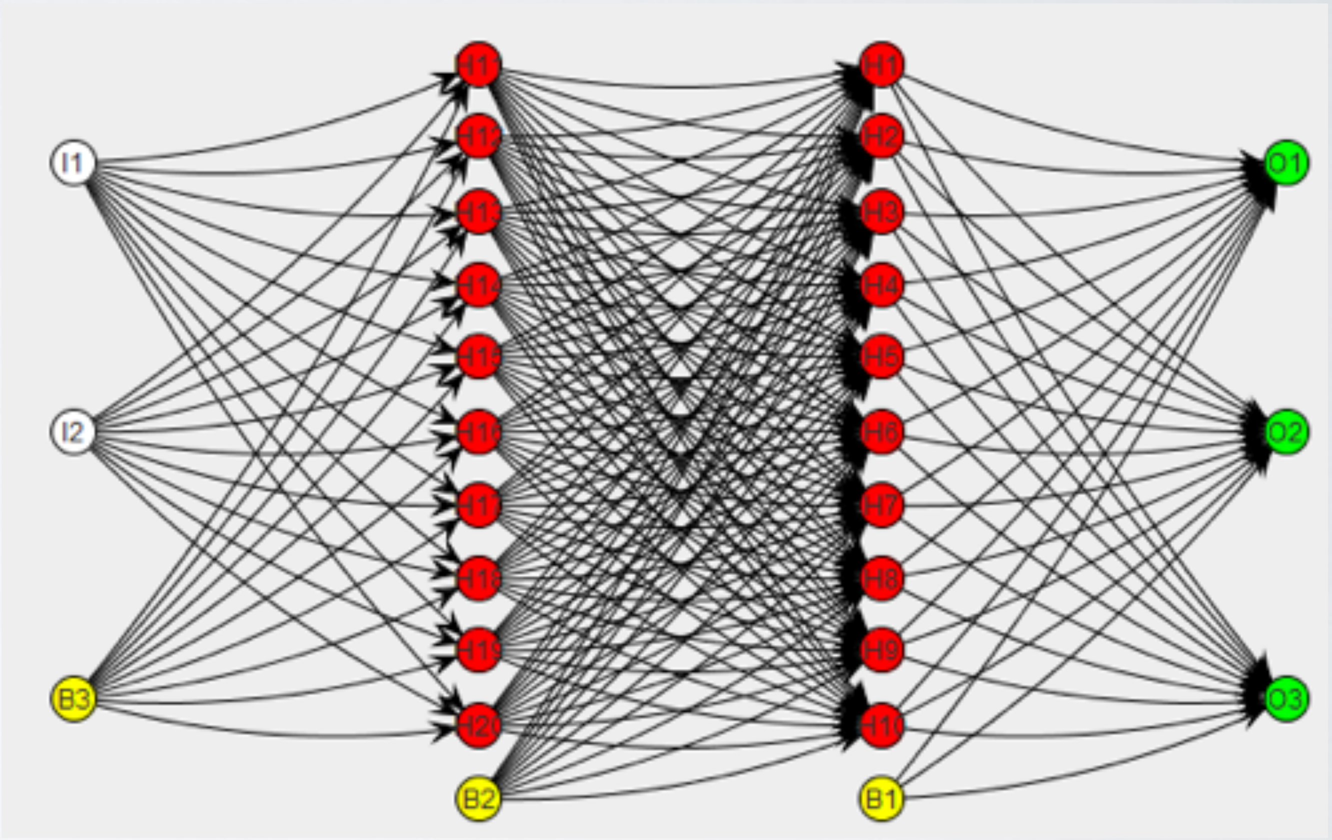
Goldstein & Taylor: "Unwrapping ADMM..." AISTATS 16

MORE COMPLEX PROBLEMS

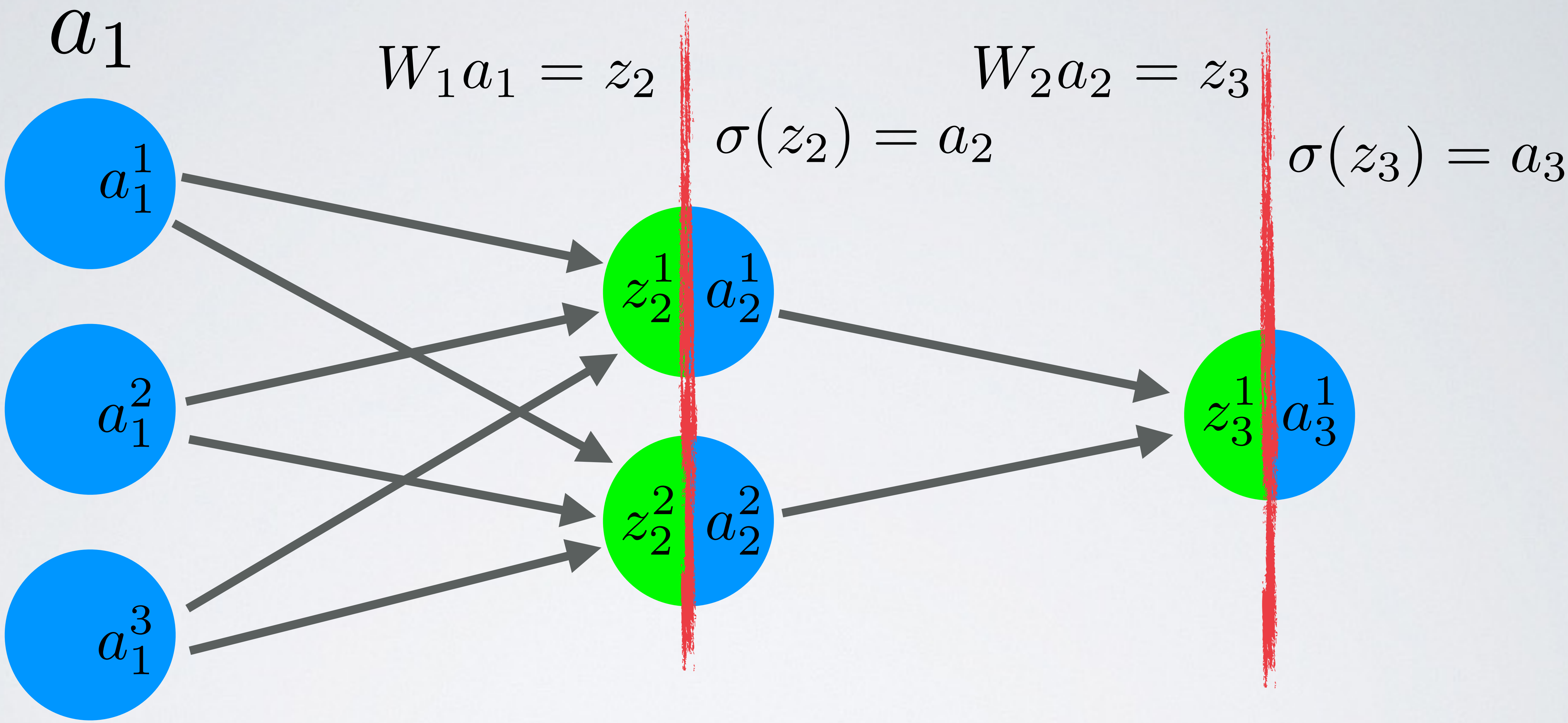
GLMs & Linear classifiers



Neural nets



ADMM FOR NEURAL NETS



$$\text{minimize} \quad \ell(a_3)$$
$$+\frac{1}{2}\|z_2 - W_1 a_1\|^2 + \frac{1}{2}\|a_2 - \sigma(z_2)\|^2$$
$$+\frac{1}{2}\|z_3 - W_2 a_2\|^2 + \frac{1}{2}\|a_3 - \sigma(z_3)\|^2$$

MINIMIZATION STEPS

$$
\begin{aligned}
\text{minimize} \quad & \ell(a_3) \\
+\frac{1}{2}\|z_2 - W_1 a_1\|^2 &+ \frac{1}{2}\|a_2 - \sigma(z_2)\|^2 \\
+\frac{1}{2}\|z_3 - W_2 a_2\|^2 &+ \frac{1}{2}\|a_3 - \sigma(z_3)\|^2
\end{aligned}
$$

Solve for weights: least squares convex

Solve for activations: least squares + ridge penalty convex

Solve for inputs: coordinate-minimization non-convex but **global**

Use TR to solve least squares: scales across nodes

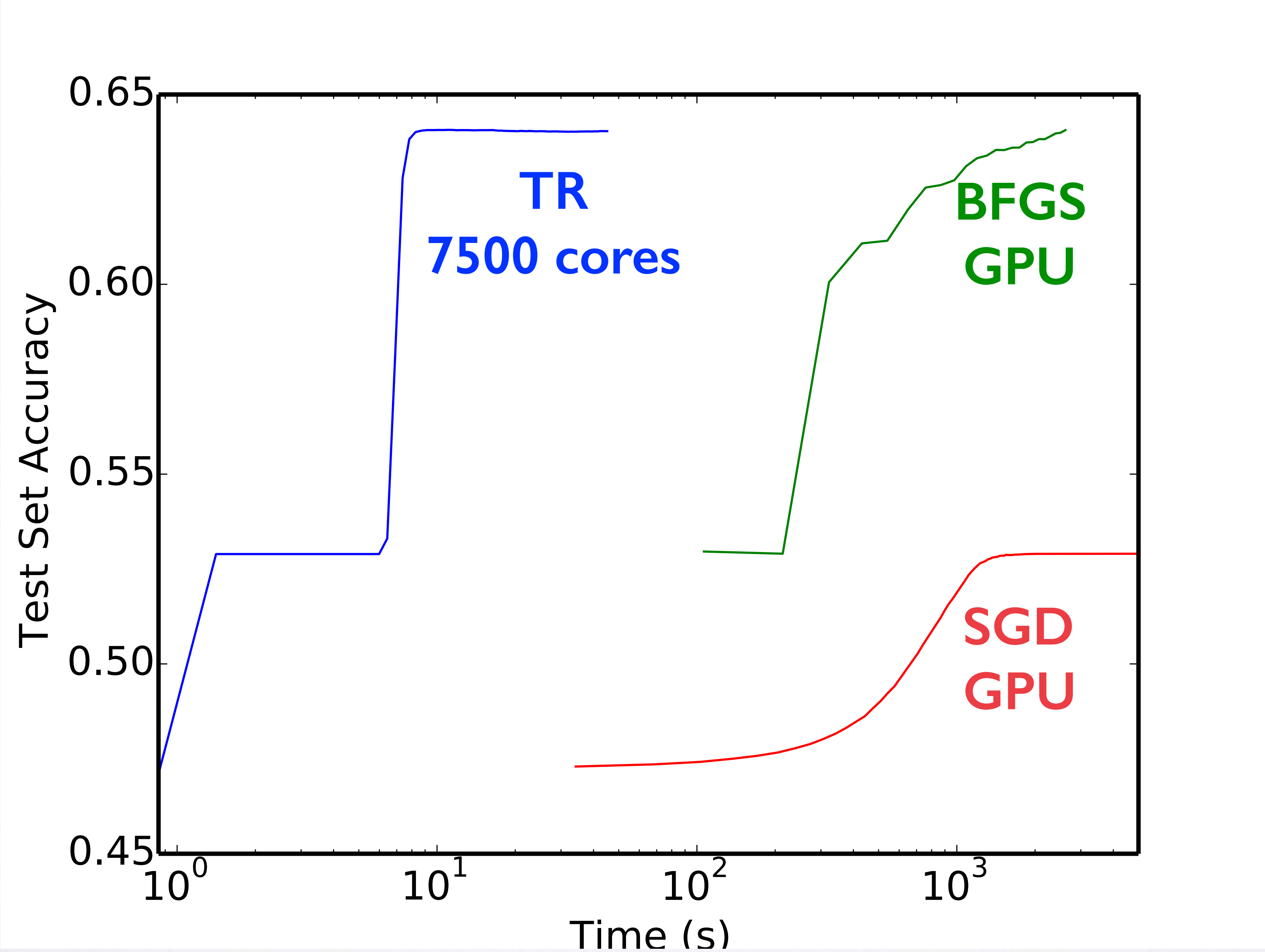
NEURAL NETS

3 hidden layers

~150K weights

"Higgs" particle classification

7500 core TR vs GPU



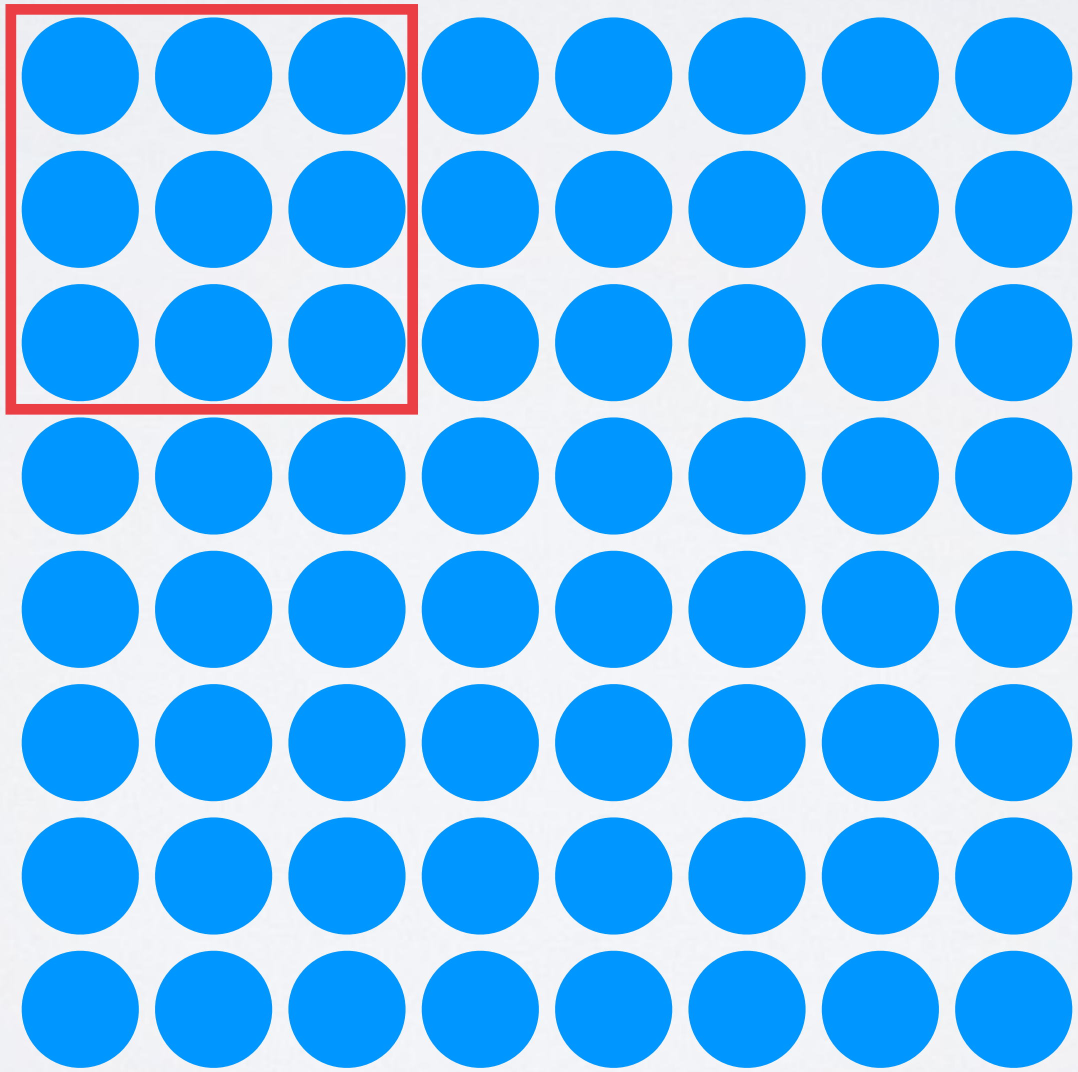
"Training Neural Networks Without Gradients: A Scalable ADMM Approach."
Taylor, Burmeister, Xu, Singh, Patel, Goldstein. ICML 2016

Imaging Application

Regularizing Smooth Support

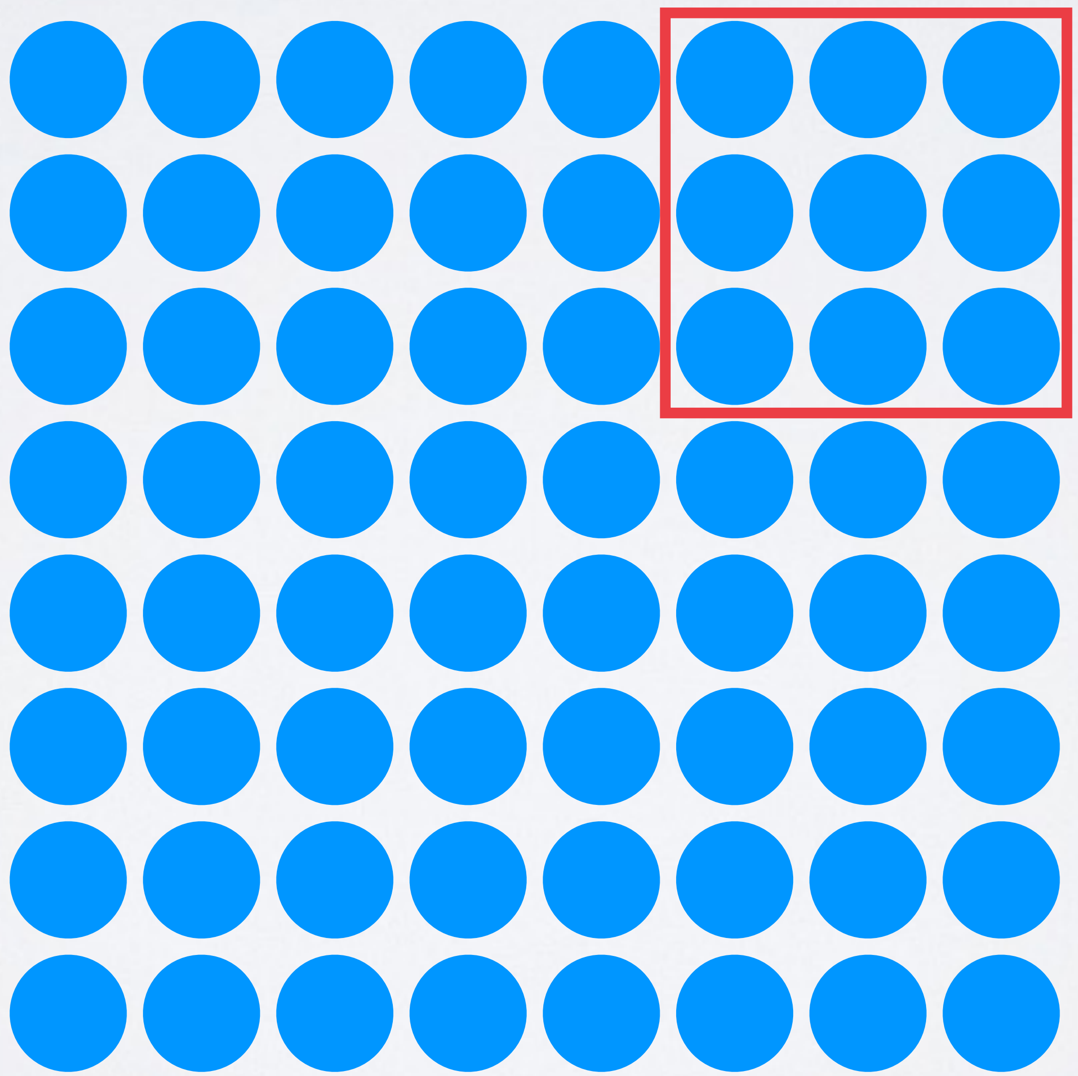
BLOCK-SPARSE REGULARIZATION

$$J(x) = \sum_{c \in \mathcal{C}} \|x_c\|_2,$$



BLOCK-SPARSE REGULARIZATION

$$J(x) = \sum_{c \in \mathcal{C}} \|x_c\|_2,$$



NUMERICS

simple block sparse problem

$$\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{x}-\mathbf{v}\|_2^2 + \lambda \sum_{c\in\mathcal{C}_i} \|\mathbf{x}_c\|_2.$$

constrained formulation

$$\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{x}-\mathbf{v}\|_2^2 + \lambda \sum_{c\in\mathcal{C}_i} \|\mathbf{y}_c\|_2.$$

$$\text{subject to} \;\; \mathbf{y}_c = \mathbf{x}_c, \;\forall c$$

augmented Lagrangian

$$\underset{\mathbf{x}\in\mathbb{R}^N}{\text{minimize}} \quad \|\mathbf{x}-\mathbf{v}\|_2^2 + \sum_{c\in\mathcal{C}_i} \lambda\|\mathbf{y}_c\|_2 + \frac{\tau}{2}\|\mathbf{y}_c - \mathbf{x}_c + \lambda_c\|^2$$

Yuan and Yin. "Model Selection and Regression…" 2006

LOW-RANK + SPARSE

L1 + low rank

Block Sparse + low rank

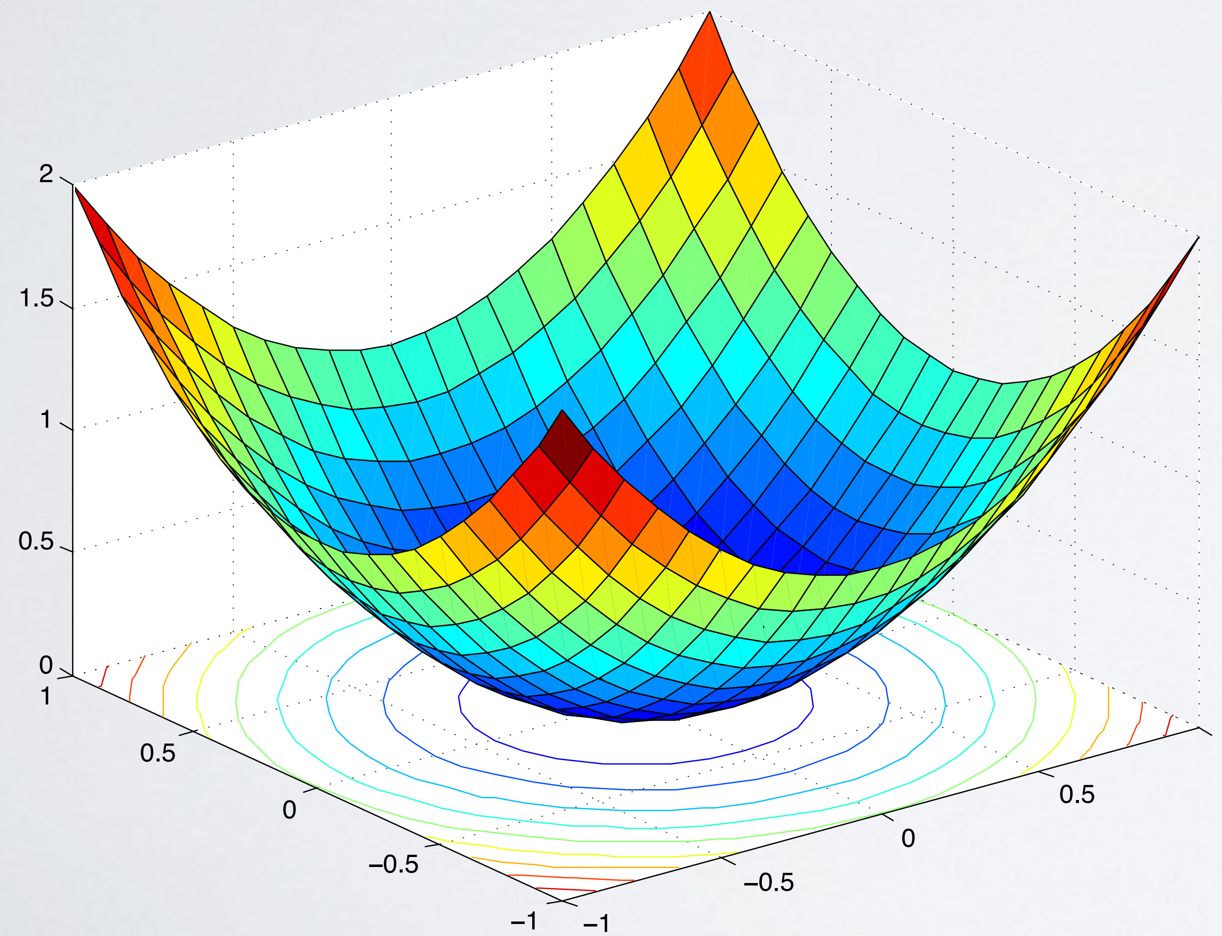
Shah, Studer, Goldstein. "Convex Regularization of Sparse Signals with Smooth Support"

ACCELERATED SPLITTING METHODS

HOW TO MEASURE CONVERGENCE?

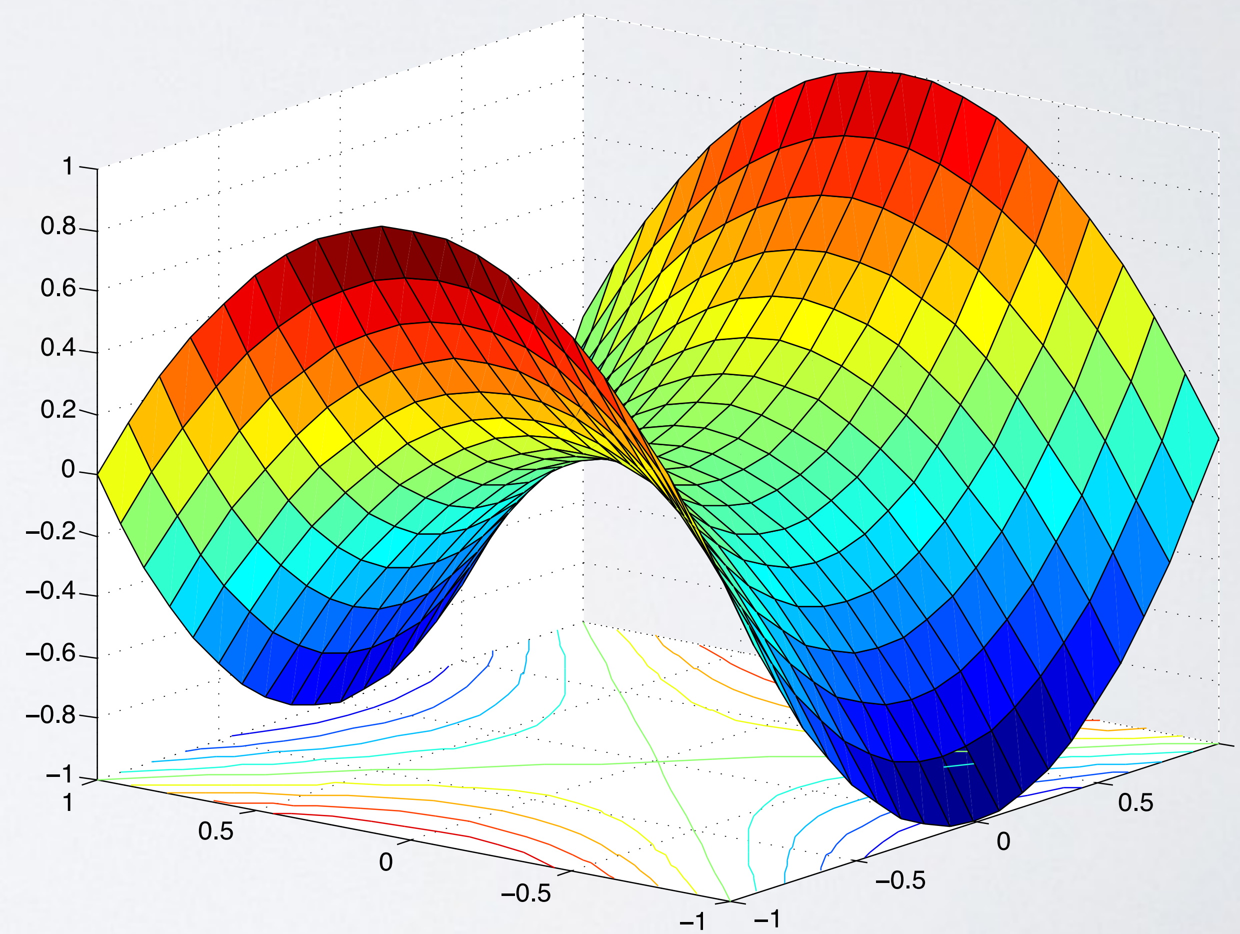
No "Objective" to minimize

Unconstrained



Convex

Constrained



Saddle

RESIDUALS

$$
\begin{array}{ll}
\text{minimize} & H(u) + G(v) \\
\text{subject to} & Au + Bv = b
\end{array}
$$

- Lagrangian

$$\min_{u,v} \max_{\lambda} \quad H(u) + G(v) + \langle \lambda, b - Au - Bv \rangle$$

- Derivative for λ $\qquad b - Au - Bv = 0$

- Derivative for u $\qquad \partial H(u) - A^T \lambda = 0$

- We have converge when derivatives are 'small'

$$r_k = b - Au_k - Bv_k$$

$$d_k = \partial H(u_k) - A^T \lambda_k$$

$$d_k = A^T B(v_k - v_{k-1})$$

EXPLICIT RESIDUALS

- Explicit formulas for residuals

$$r_k = b - Au_k - Bv_k$$
$$d_k = \tau A^T B(v_k - v_{k-1})$$

- Combined residual

$$c_k = \|r_k\|^2 + \frac{1}{\tau}\|d_k\|^2$$

- ADMM/AMA converge at rate

$$c_k \leq O(1/k)$$

Goal: $O\left(\frac{1}{k^2}\right)$

Goldstein, O'Donoghue, Setzer, Baraniuk. 2012 & Yuan and He. 2012

FAST ADMM

Require: $v_{-1} = \hat{v}_0 \in R^{N_v}, \lambda_{-1} = \hat{\lambda}_0 \in R^{N_b}, \tau > 0$

for $k = 1, 2, 3 \ldots$ **do**

$$u_k = \operatorname{argmin} H(u) + \langle \hat{\lambda}_k, -Au \rangle + \frac{\tau}{2}\|b - Au - B\hat{v}_k\|^2$$

$$v_k = \operatorname{argmin} G(v) + \langle \hat{\lambda}_k, -Bv \rangle + \frac{\tau}{2}\|b - Au_k - Bv\|^2$$

$$\lambda_k = \hat{\lambda}_k + \tau(b - Au_k - Bv_k)$$

$$\alpha_{k+1} = \frac{1+\sqrt{1+4\alpha_k^2}}{2}$$

$$\hat{v}_{k+1} = v_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(v_k - v_{k-1})$$

$$\hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}}(\lambda_k - \lambda_{k-1})$$

end for

Goldstein, O'Donoghue, Setzer, Baraniuk. "Fast alternating direction methods." 2014

FAST ADMM

Require: $v_{-1} = \hat{v}_0 \in R^{N_v}, \lambda_{-1} = \hat{\lambda}_0 \in R^{N_b}, \tau > 0$

for $k = 1, 2, 3 \ldots$ **do**

$\quad u_k = \operatorname{argmin} H(u) + \langle \hat{\lambda}_k, -Au \rangle + \frac{\tau}{2} \|b - Au - B\hat{v}_k\|^2$

$\quad v_k = \operatorname{argmin} G(v) + \langle \hat{\lambda}_k, -Bv \rangle + \frac{\tau}{2} \|b - Au_k - Bv\|^2$

$\quad \lambda_k = \hat{\lambda}_k + \tau(b - Au_k - Bv_k)$

$\quad \alpha_{k+1} = \frac{1 + \sqrt{1 + 4\alpha_k^2}}{2}$

$\quad \hat{v}_{k+1} = v_k + \frac{\alpha_k - 1}{\alpha_{k+1}} (v_k - v_{k-1})$

$\quad \hat{\lambda}_{k+1} = \lambda_k + \frac{\alpha_k - 1}{\alpha_{k+1}} (\lambda_k - \lambda_{k-1})$

end for

Goldstein, O'Donoghue, Setzer, Baraniuk. “Fast alternating direction methods.” 2014

CONVERGENCE RESULTS

To prove formal convergence bounds, need assumptions:

- Strong convexity of the objective
- Stepsize restriction

Theorem

Suppose H and G are strongly convex and that

$$\tau^3 < \frac{\sigma_H \sigma_G^2}{\rho(A^T A)\rho(B^T B)^2},$$

then fast ADMM converges with

$$c_k \leq \frac{C\tau \|\hat{\lambda}_1 - \lambda^\star\|^2}{(k+2)^2}.$$

Without strong convexity, convergence is still guaranteed using a "restart" method.

Goldstein, O'Donoghue, Setzer, Baraniuk. "Fast alternating direction methods." 2014

RESULTS: ROF

$$\min |\nabla u| + \frac{\mu}{2}\|u - f\|^2$$

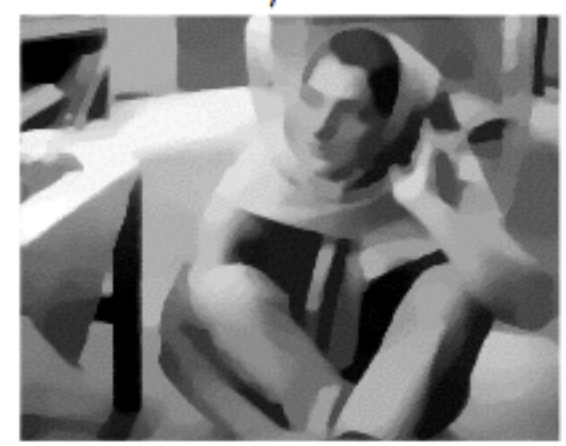
10/17

24/86

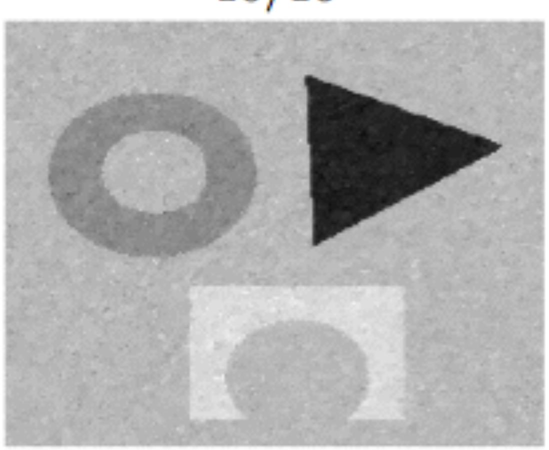
172/3116

10/16

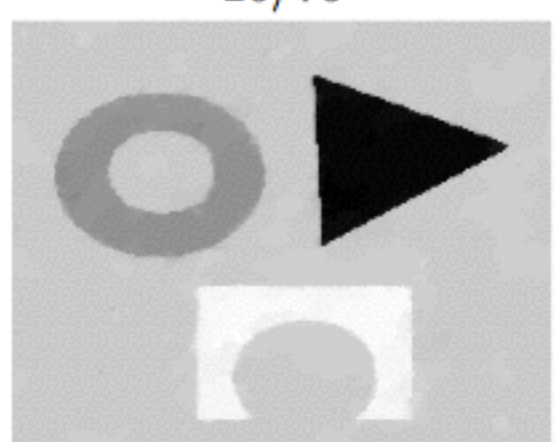
20/75



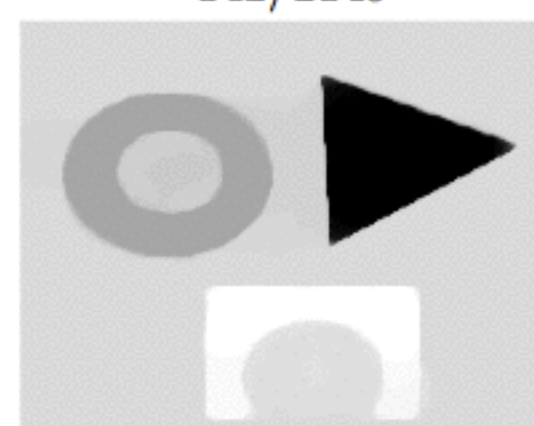
142/2149



10/16



25/97



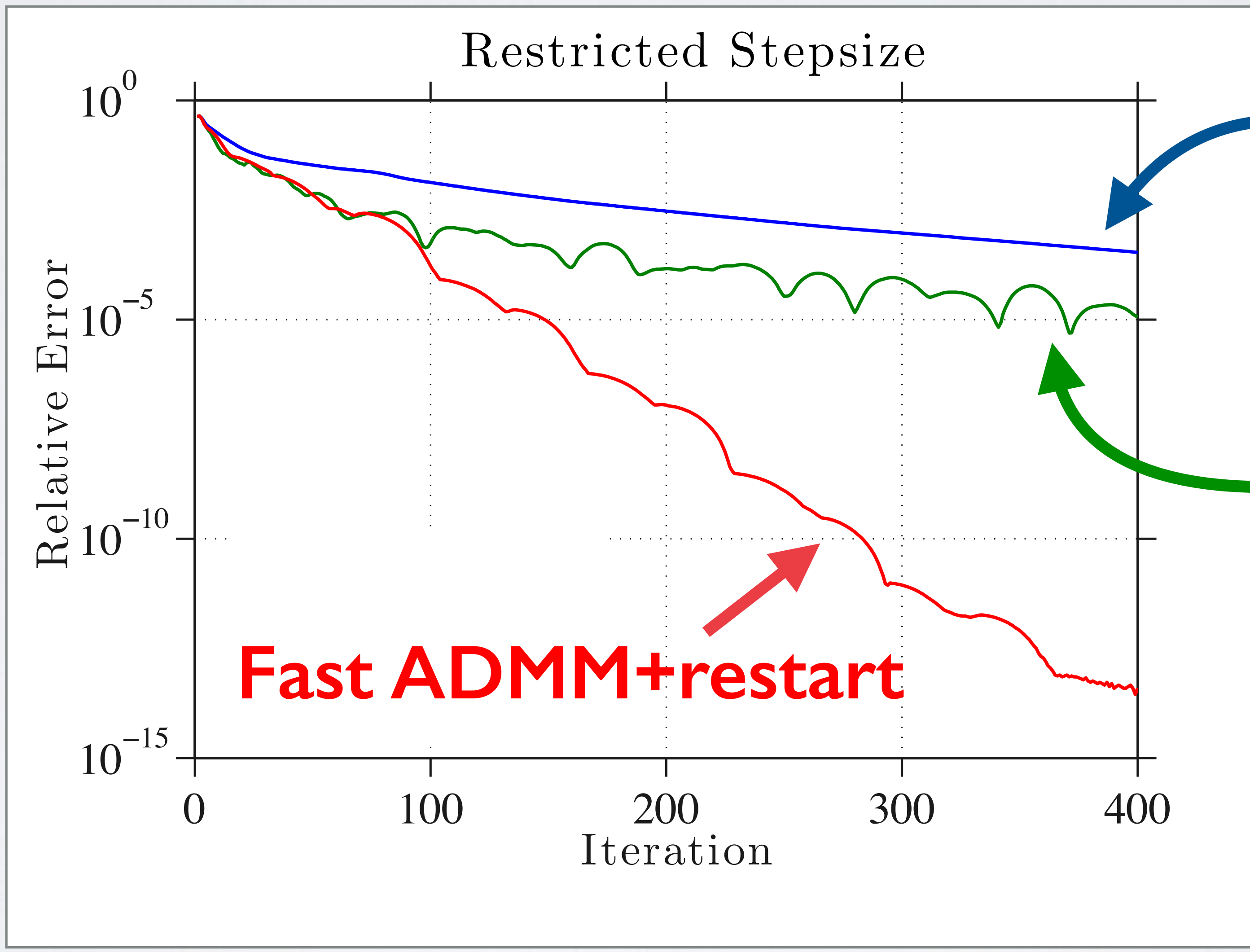
190/4990

ELASTIC NET REGRESSION

$$\min_u \lambda_1 |u| + \frac{\lambda_2}{2}\|u\|^2 + \frac{1}{2}\|Au - f\|^2$$

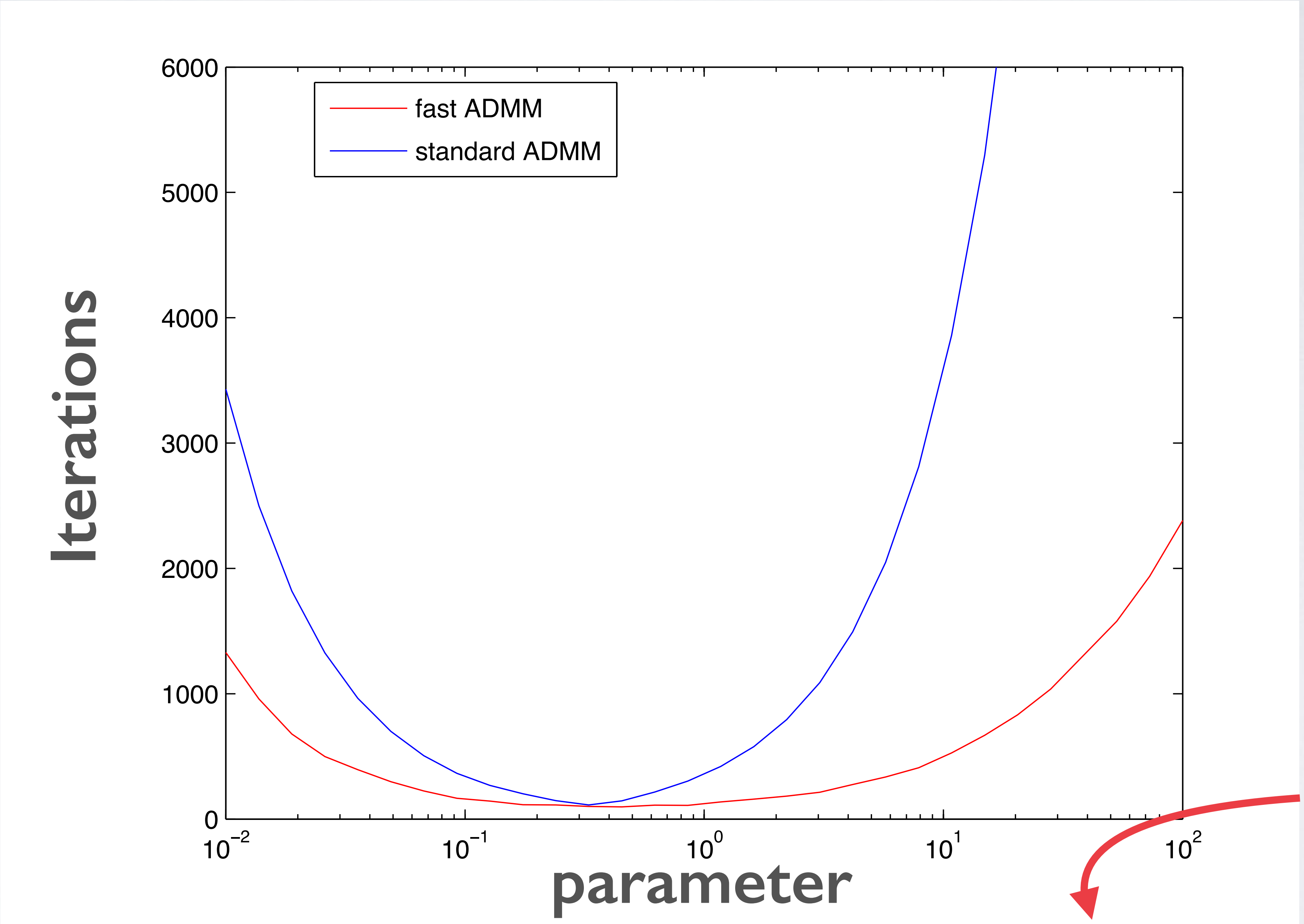
Random A, Sparsity = 15/40

20X Faster



Goldstein, O'Donoghue, Setzer, Baraniuk. “Fast alternating direction methods.” 2014

BIG ADVANTAGE: PARAMETER SENSITIVITY



vary this guy

$$\max_{\lambda}\min_{u,v} H(u)+G(v)+\langle \lambda, b-Au-Bv\rangle+\frac{\tau}{2}\|b-Au-Bv\|^2$$

NEW STUFF:
ADAPTIVE ADMM

$$\begin{array}{ll} \text{minimize} & H(u) + G(v) \\ \text{subject to} & Au + Bv = b \end{array}$$

how to choose?

Augmented Lagrangian

$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \langle \lambda, b - Au - Bv \rangle + \frac{\tau}{2} \|b - Au - Bv\|^2$$

SPECTRAL STEPSIZE RULES

$$\begin{array}{ll} \text{minimize} & H(u)+G(v) \\ \text{subject to} & Au+Bv=b \end{array}$$

Advantages

Fast! Superlinear for some problems

Automated

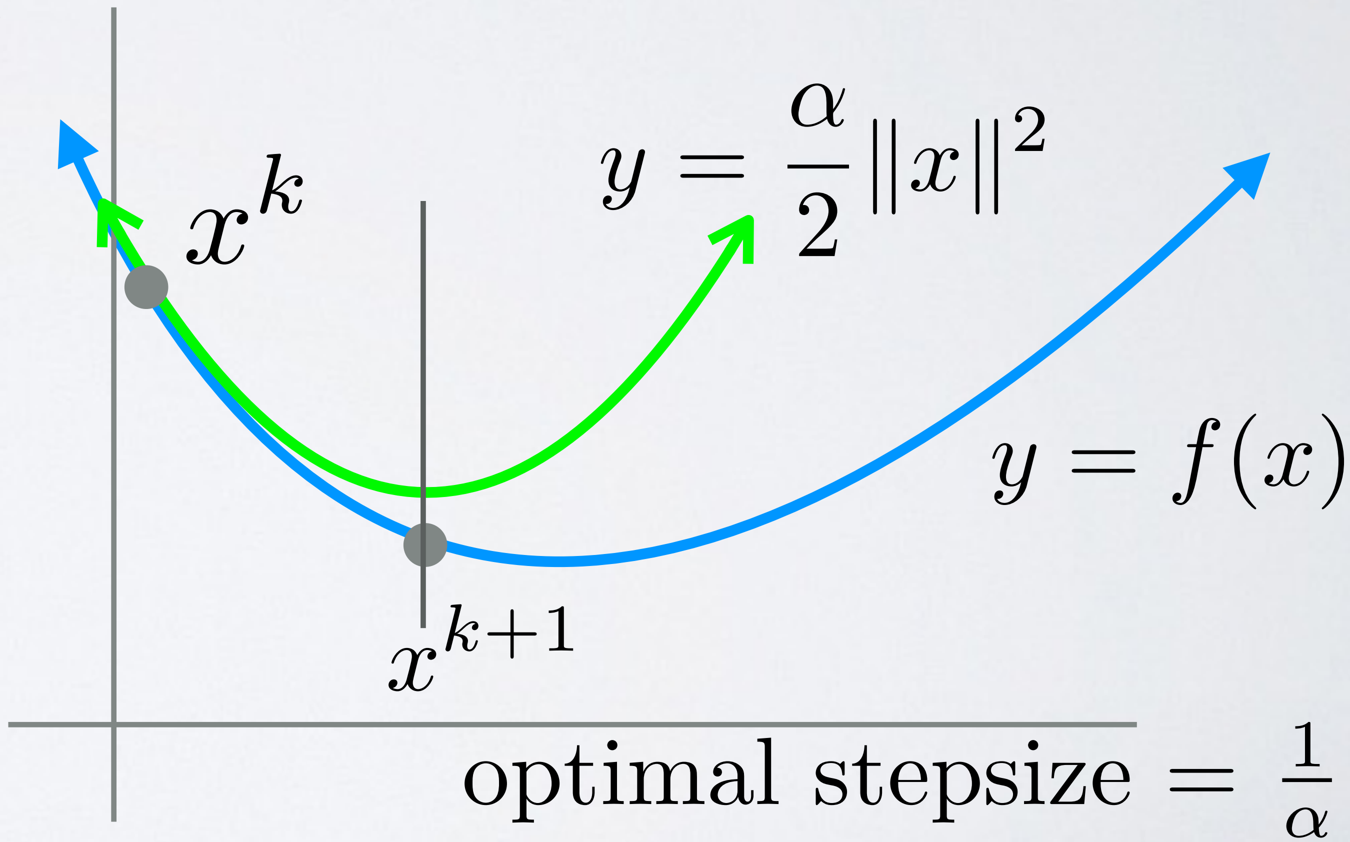
Spectral adaptive methods

Gradient descent: Barzilai-Borwein

Forward-backward: SpaRSA

Constrained problems: ???

"Spectral" approximation



ADAPTIVE ADMM

$$\begin{array}{ll} \text{minimize} & H(u) + G(v) \\ \text{subject to} & Au + Bv = b \end{array}$$

$$\Downarrow$$

dual problem: no constraints

$$\min_{\lambda} \underbrace{H^*(A^T\lambda)}_{\frac{\alpha}{2}\|\lambda\|^2} - \langle \lambda, b \rangle + \underbrace{G^*(B^T\lambda)}_{\frac{\beta}{2}\|\lambda\|^2}$$

$$\text{optimal stepsize} = \frac{1}{\sqrt{\alpha\beta}}$$

ADAPTIVE ADMM

$$\begin{array}{ll} \text{minimize} & H(u) + G(v) \\ \text{subject to} & Au + Bv = b \end{array}$$

curvatures are "free" given ADMM iterates

$$\min_\lambda \underbrace{H^*(A^T\lambda)}_{\frac{\alpha}{2}\|\lambda\|^2} - \langle \lambda, b\rangle + \underbrace{G^*(B^T\lambda)}_{\frac{\beta}{2}\|\lambda\|^2}$$

$$\text{optimal stepsize} = \frac{1}{\sqrt{\alpha\beta}}$$

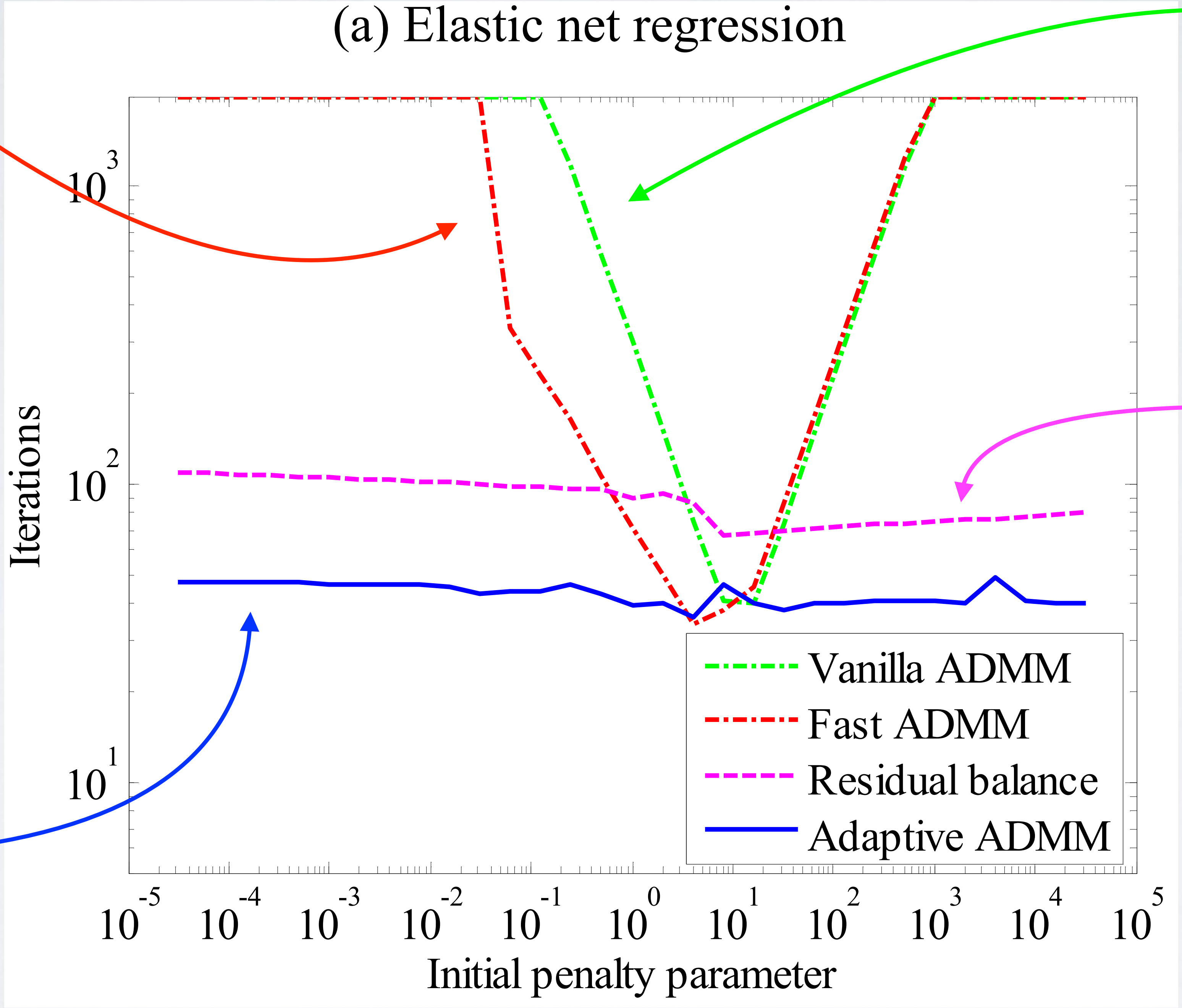
$$\alpha = \frac{(\hat\lambda_k - \hat\lambda_0)^T A(u_k - u_{k_0})}{\|\hat\lambda_k - \hat\lambda_0\|^2}$$

$$\beta = \frac{(\lambda_k - \lambda_0)^T B(v_k - v_{k_0})}{\|\lambda_k - \lambda_0\|^2}$$

DEPENDENCE ON STEPSIZE GUESS

Fast ADMM

Vanilla ADMM

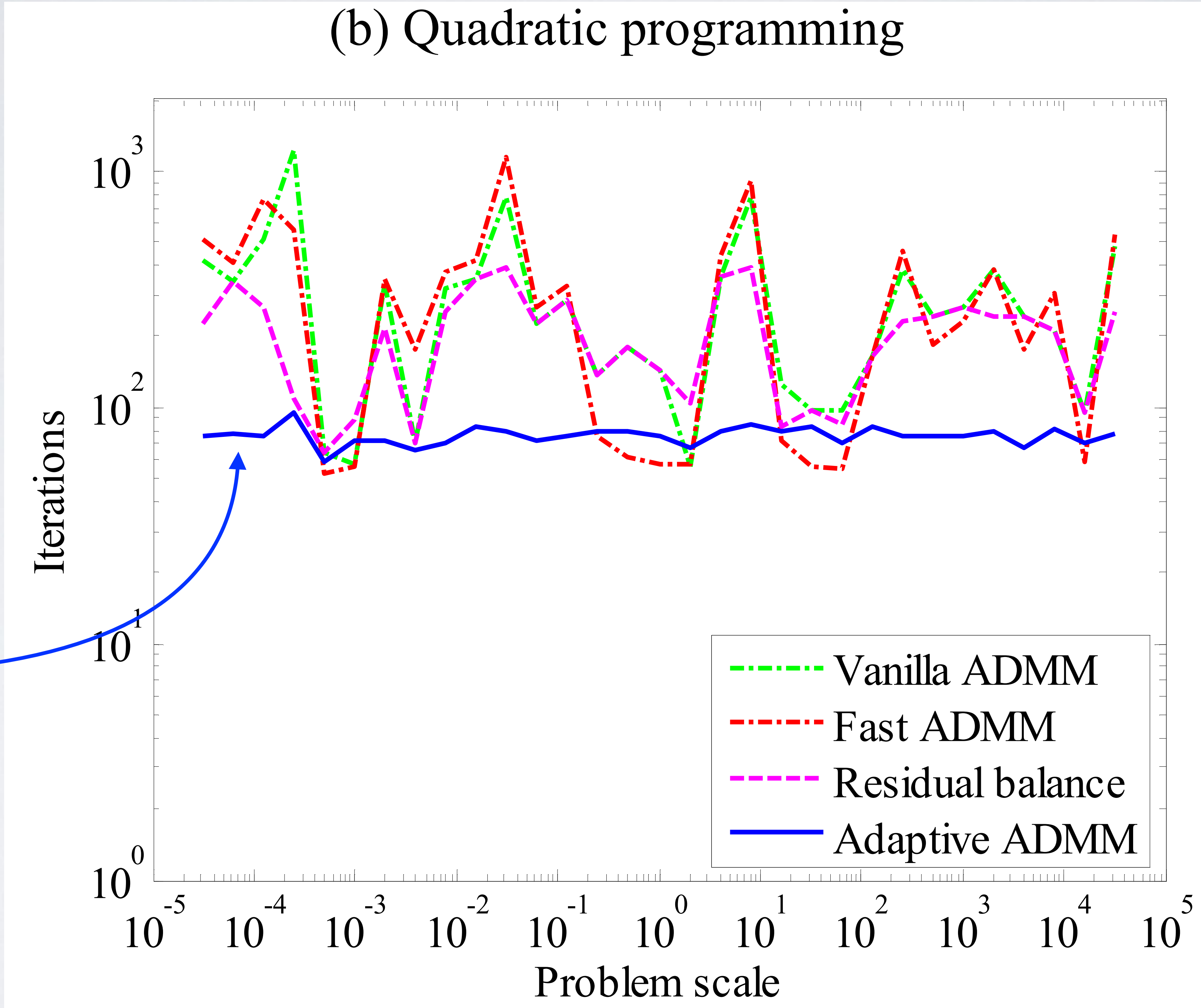


Residual Balancing

Adaptive ADMM

Zheng Xu, Mario Figueiredo, Tom Goldstein.
"Adaptive ADMM with spectral penalty parameter selection." 2014

PROBLEM SCALING



Adaptive ADMM

Zheng Xu, Mario Figueiredo, Tom Goldstein. "Adaptive ADMM with spectral penalty parameter selection." 2014

ACKNOWLEDGEMENTS

Thanks to my collaborators

Fast Alternating Direction Methods

Brendan O'Donoghue (Google Deepmind)
Ernie Esser (Univ. of British Columbia)
Simon Setzer (Saarland University)
Rich Baraniuk (Rice)

"Transpose Reduction" & "Training Neural Nets without Gradients"

Gavin Taylor (US Naval Academy)
Ankit Patel (Rice)

Adaptive ADMM with spectral penalty parameter selection

Zheng Xu (Maryland)
Mario Figueiredo (University of Lisbon)