HIGHLY PARALLEL METHODS FOR MACHINE LEARNING AND SIGNAL RECOVERY

Tom Goldstein

TOPICS

Introduction

ADMM / Fast ADMM

Application: Distributed computing

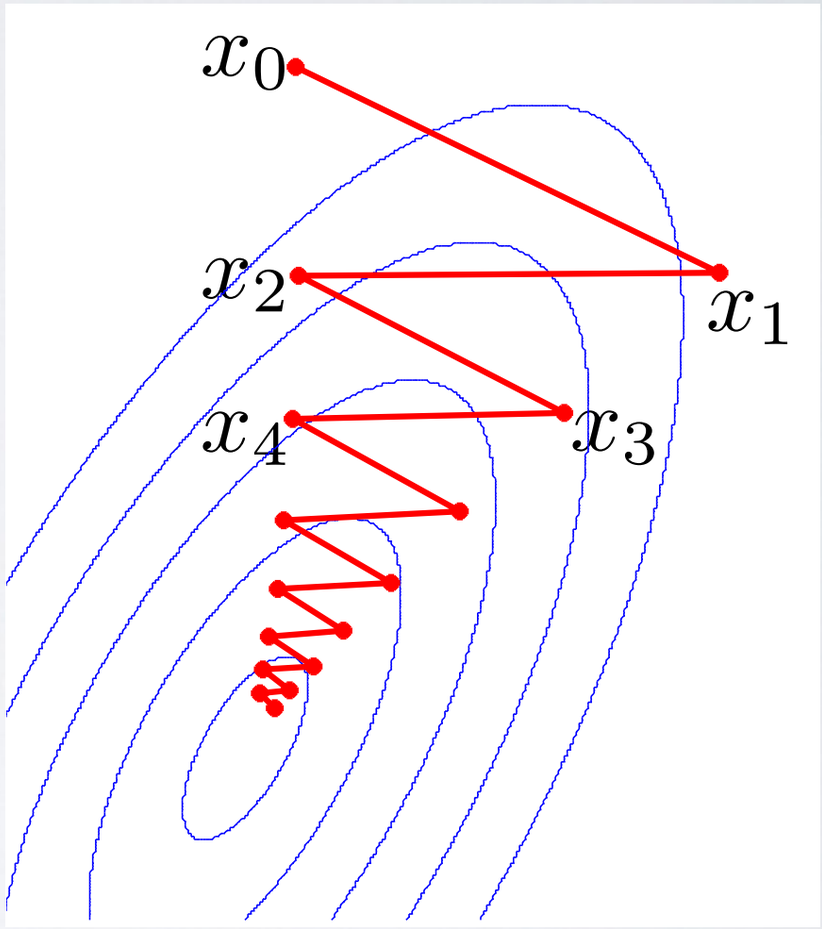
Automation & Adaptivity

FIRST-ORDER METHODS

$$\underset{x}{\text{minimize}}\, F(x)$$

Generalizes Gradient descent

$$x_{k+1} = x_k - \tau \nabla F(x_k)$$

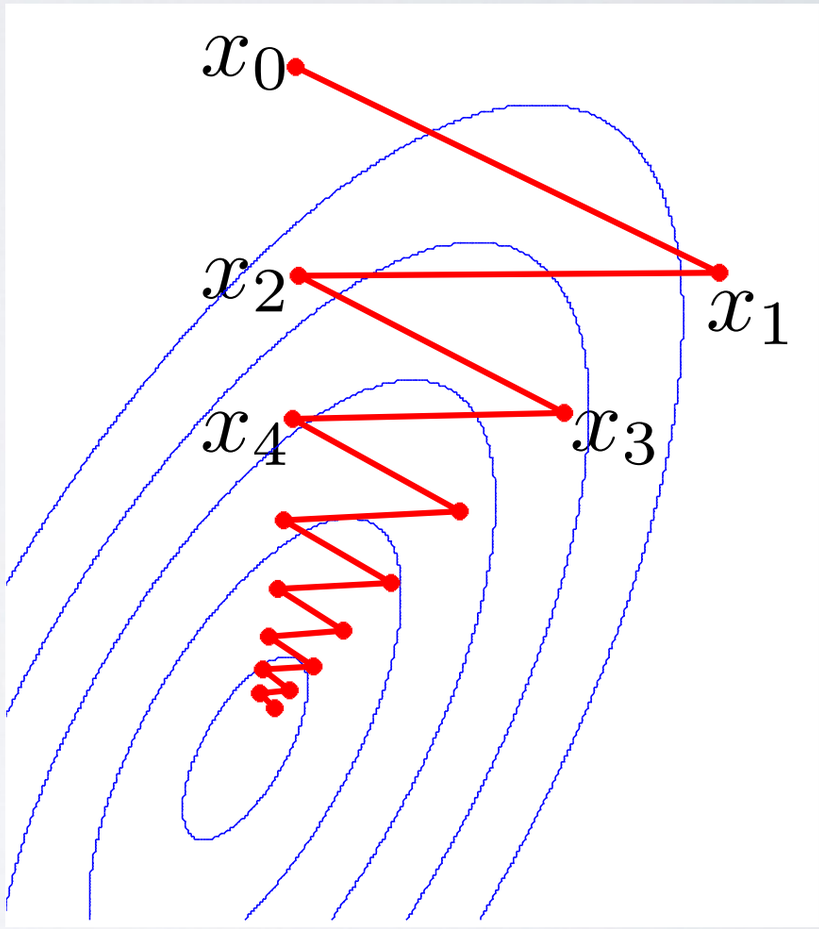


FIRST-ORDER METHODS

$$\underset{x}{\text{minimize}}\, F(x)$$

Generalizes Gradient descent

$$x_{k+1} = x_k - \tau \nabla F(x_k)$$



Pros

- Linear complexity
- Parallelizable
- Low memory requirements

FIRST-ORDER METHODS

$$\underset{x}{\text{minimize}}\, F(x)$$

Generalizes Gradient descent

$$x_{k+1} = x_k - \tau \nabla F(x_k)$$

Pros

- Linear complexity
- Parallelizable
- Low memory requirements

Con

- Poor convergence rates

FIRST-ORDER METHODS

$$\underset{x}{\text{minimize}}\, F(x)$$

Generalizes Gradient descent

$$x_{k+1} = x_k - \tau \nabla F(x_k)$$

Pros

- Linear complexity
- Parallelizable
- Low memory requirements

Con

- Poor convergence rates

Solution:
Adaptivity and Acceleration

CONSTRAINED PROBLEMS

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

Big idea: Lagrange multipliers

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

CONSTRAINED PROBLEMS

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

Big idea: Lagrange multipliers

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

CONSTRAINED PROBLEMS

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

Big idea: Lagrange multipliers

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

CONSTRAINED PROBLEMS

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

Big idea: Lagrange multipliers

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

CONSTRAINED PROBLEMS

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

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

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

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

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

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

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

EXAMPLE PROBLEMS

Non-Smooth Problems

- TV Denoising

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

noisy image

- TV Deblurring

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

- General Problem:

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

Goldstein & Osher, "Split Bregman," 2009

EXAMPLE PROBLEMS

Non-Smooth Problems

- TV Denoising

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

clean image

- TV Deblurring

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

- General Problem:

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

Goldstein & Osher, "Split Bregman," 2009

EXAMPLE PROBLEMS

Non-Smooth Problems

- TV Denoising

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

total variation

- TV Deblurring

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

- General Problem:

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

Goldstein & Osher, "Split Bregman," 2009

EXAMPLE PROBLEMS

Non-Smooth Problems

- TV Denoising

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

- TV Deblurring

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

blurred image

- General Problem:

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

Goldstein & Osher, "Split Bregman," 2009

EXAMPLE PROBLEMS

Non-Smooth Problems

- TV Denoising

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

- TV Deblurring

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

Convolution

- General Problem:

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

Goldstein & Osher, "Split Bregman," 2009

EXAMPLE PROBLEMS

Non-Smooth Problems

- TV Denoising

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

- TV Deblurring

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

- General Problem:

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

General Problem

Goldstein & Osher, "Split Bregman," 2009

WHY IS SPLITTING GOOD?

Non-Smooth Problems

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

Goldstein & Osher, "Split Bregman," 2009

WHY IS SPLITTING GOOD?

Non-Smooth Problems

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

- Make change of Variables: $v \leftarrow \nabla u$

Goldstein & Osher, "Split Bregman," 2009

WHY IS SPLITTING GOOD?

Non-Smooth Problems

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

- Make change of Variables: $v \leftarrow \nabla u$
- 'Split Bregman' form:

$$\begin{aligned} \text{minimize} \quad & |v| + \frac{\mu}{2}\|Au - f\|^2 \\ \text{subject to} \quad & v - \nabla u = 0 \end{aligned}$$

Goldstein & Osher, "Split Bregman," 2009

WHY IS SPLITTING GOOD?

Non-Smooth Problems

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

- Make change of Variables: $v \leftarrow \nabla u$
- 'Split Bregman' form:

$$\begin{aligned} \text{minimize} \quad & |v| + \frac{\mu}{2}\|Au - f\|^2 \\ \text{subject to} \quad & v - \nabla u = 0 \end{aligned}$$

- Augmented Lagrangian

$$|v| + \frac{1}{2}\|Au - f\|^2 + \langle \lambda, v - \nabla u \rangle + \frac{\tau}{2}\|v - \nabla u\|^2$$

Goldstein & Osher, "Split Bregman," 2009

WHY IS SPLITTING GOOD?

Non-Smooth Problems

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

$$|v| + \frac{1}{2}\|Au - f\|^2 + \langle \lambda, v - \nabla u \rangle + \frac{\tau}{2}\|v - \nabla u\|^2$$

ADMM for TV

$$u_{k+1} = \arg\min_u \|Au - f\|^2 + \frac{\tau}{2}\|v_k - \nabla u - \lambda_k\|^2$$

$$v_{k+1} = \arg\min_v |v| + \frac{\tau}{2}\|v - \nabla u_{k+1} - \lambda_k\|^2$$

$$\lambda_{k+1} = \lambda_k + \tau(\nabla u_{k+1} - v)$$

Goldstein, Osher. 2008

WHY IS SPLITTING BAD?

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

17

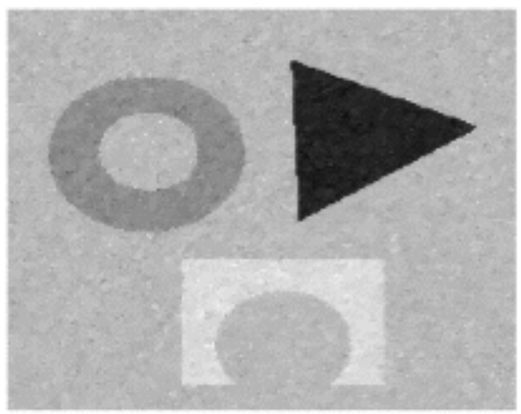
86

3116

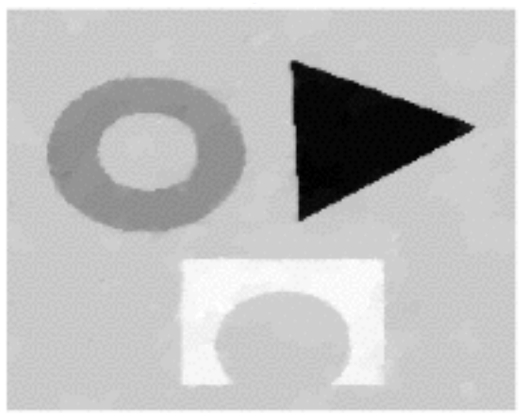
16

75

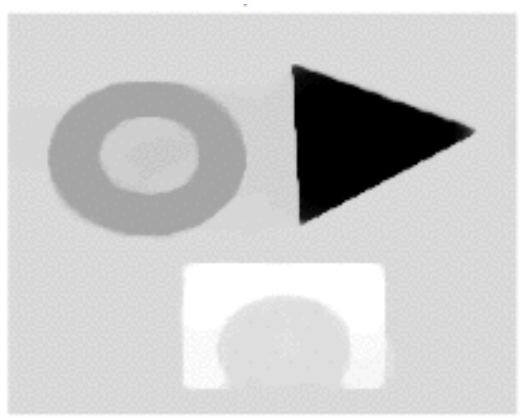
2149



16



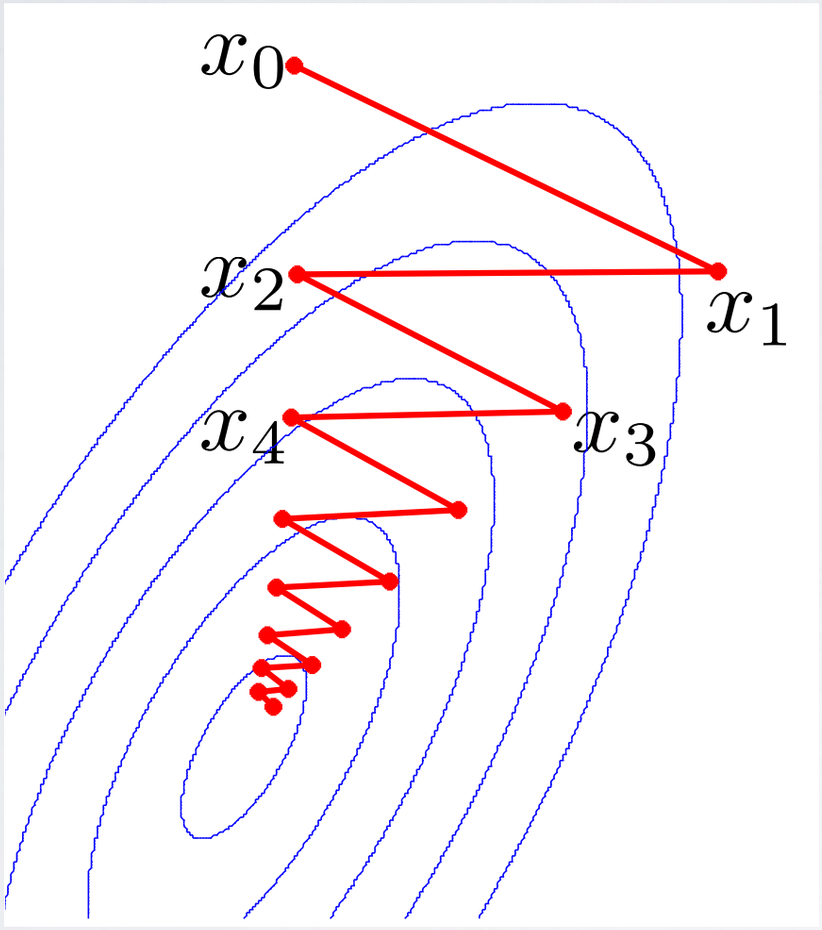
97



4990

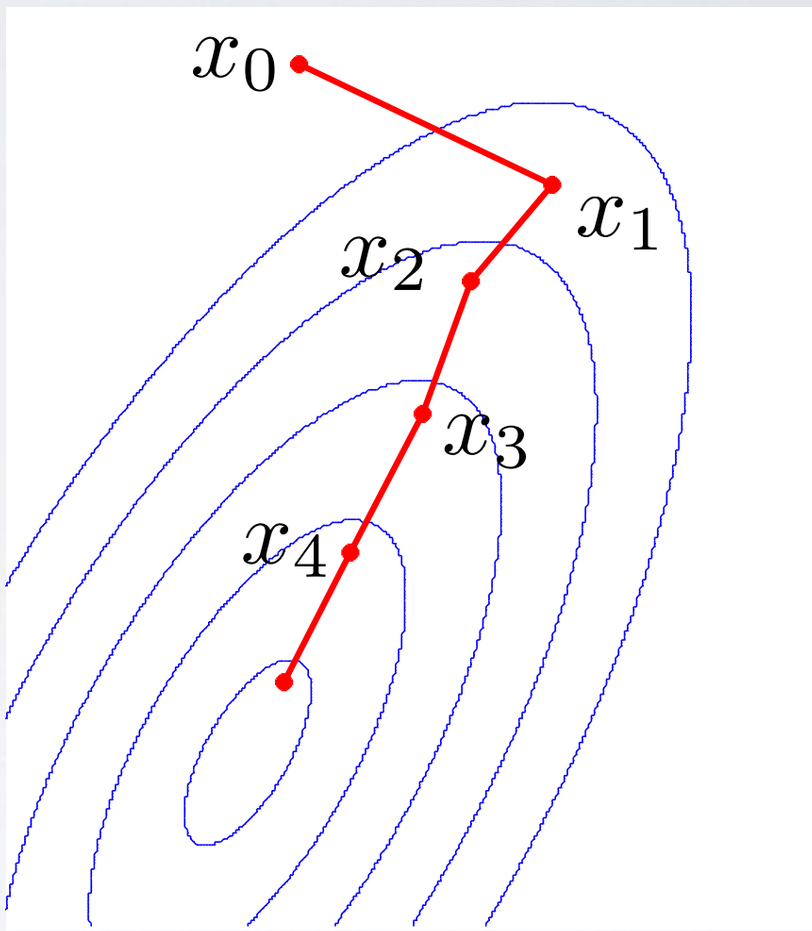
GRADIENT VS. NESTEROV

Gradient



$$O\left(\frac{1}{k}\right)$$

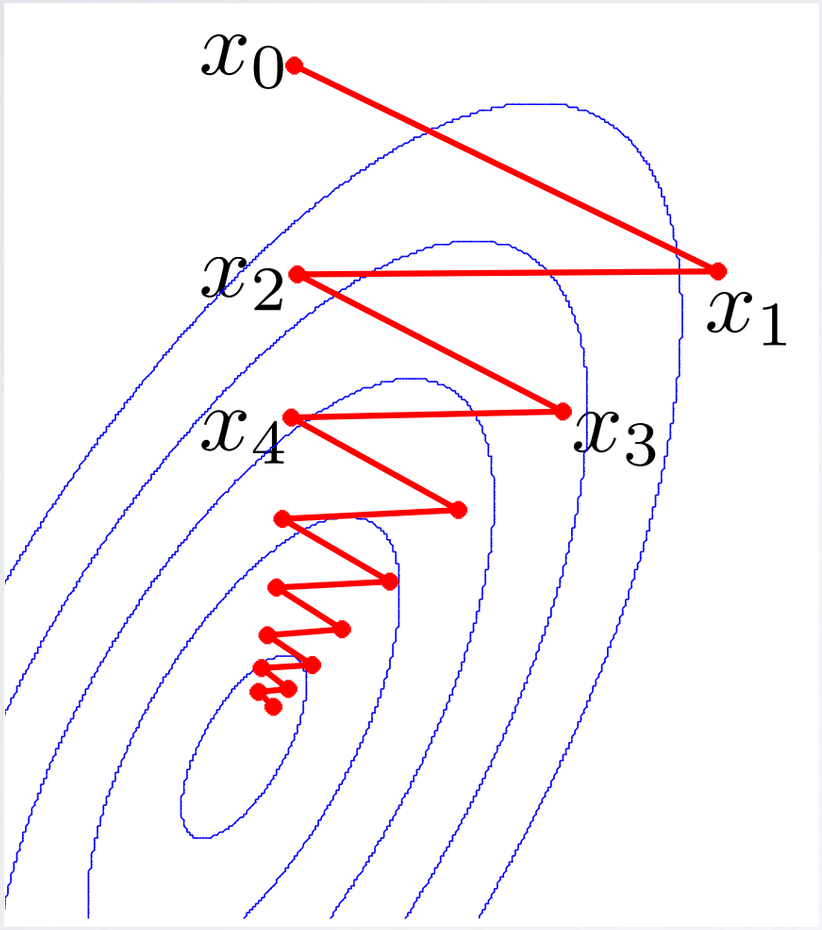
Nesterov



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

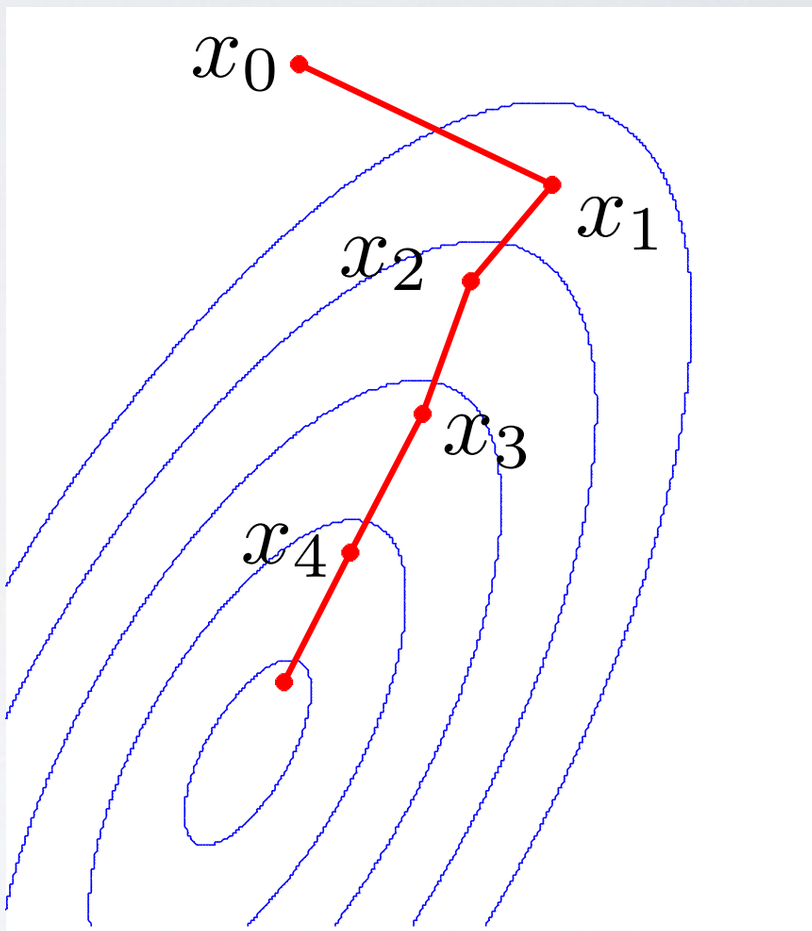
GRADIENT VS. NESTEROV

Gradient



$$O\left(\frac{1}{k}\right)$$

Nesterov



$$O\left(\frac{1}{k^2}\right) \leftarrow \text{Optimal}$$

Nemirovski and Yudin ’83

NESTEROV'S METHOD

$$\underset{x}{\text{minimize}}\, F(x)$$

Gradient Descent $$x_{k+1} = y_k - \tau \nabla F(y_k)$$

Acceleration Factor $$\alpha_{k+1} = \frac{1}{2}\left(1 + \sqrt{4\alpha_k^2 + 1}\right)$$

Prediction $$y_{k+1} = x_{k+1} + \frac{\alpha_k - 1}{\alpha_{k+1}}(x_{k+1} - x_k)$$

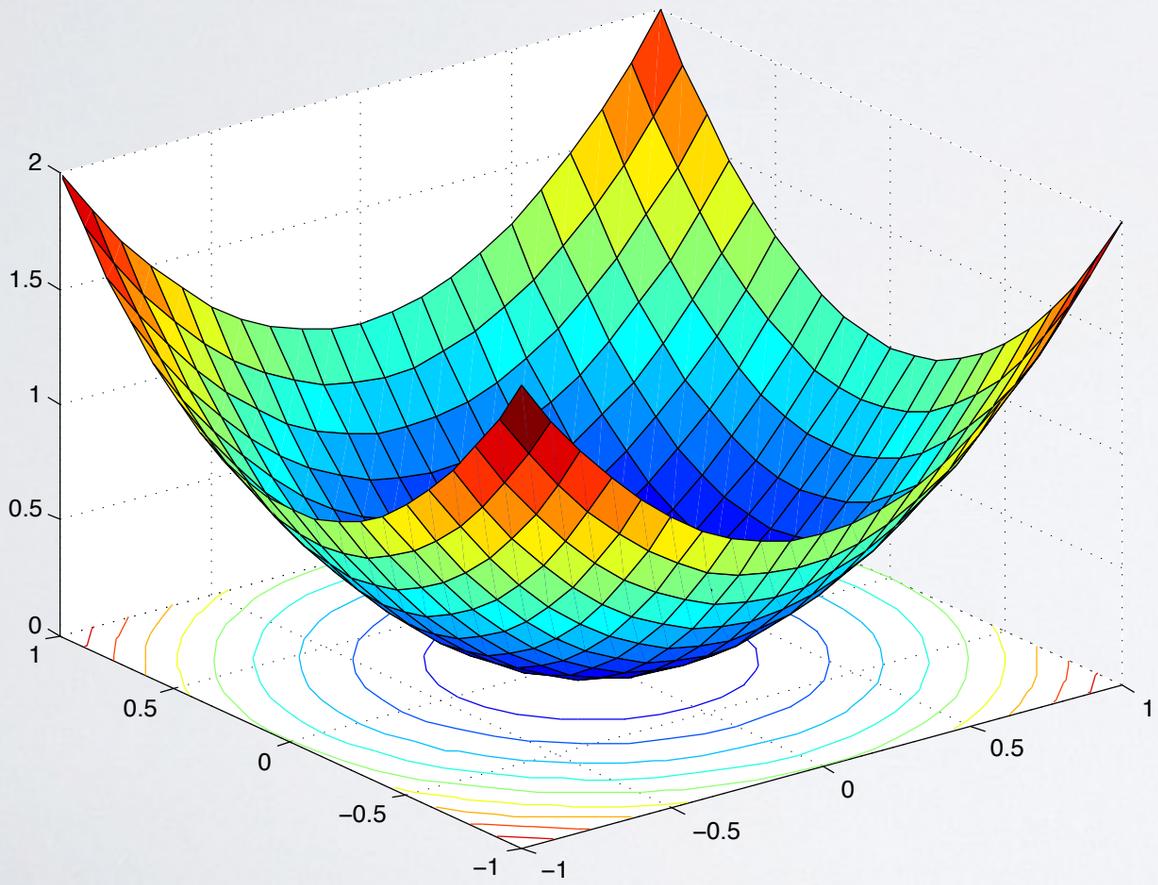
Nesterov '83

ACCELERATED SPLITTING METHODS

HOW TO MEASURE CONVERGENCE?

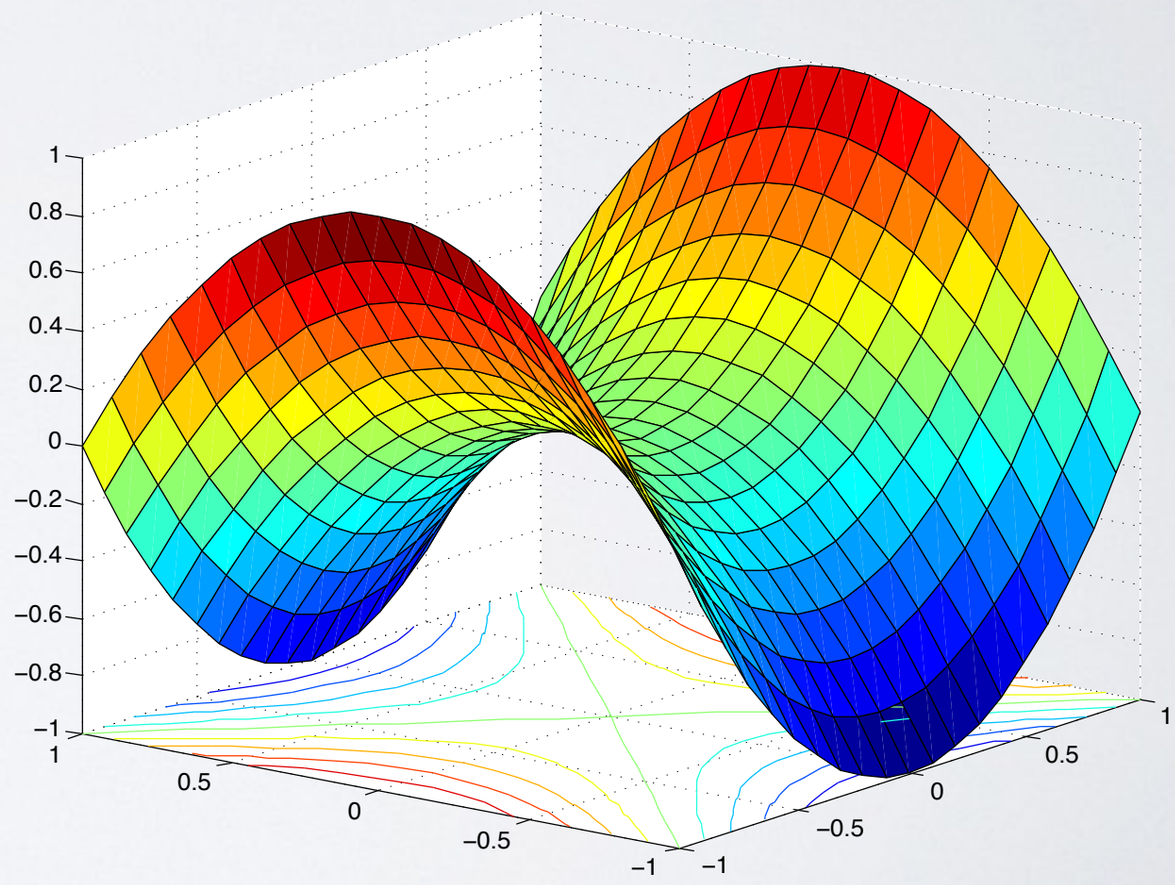
No "Objective" to minimize

Unconstrained



Convex

Constrained



Saddle

RESIDUALS

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

- Lagrangian

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

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

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 convergence when derivatives are 'small'

$$r_k = b - Au_k - Bv_k$$
$$d_k = \partial H(u_k) - A^T \lambda_k$$

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 convergence when derivatives are 'small'

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

EXPLICIT RESIDUALS

- Explicit formulas for residuals

$$r_k = b - Au_k - Bv_k$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

8: **end for**

Goldstein, O'Donoghue, Setzer, Baraniuk. 2012

FAST ADMM

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

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

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

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

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

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

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

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

8: **end for**

Goldstein, O'Donoghue, Setzer, Baraniuk. 2012

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. 2012

RESULTS: ROF

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

10/17

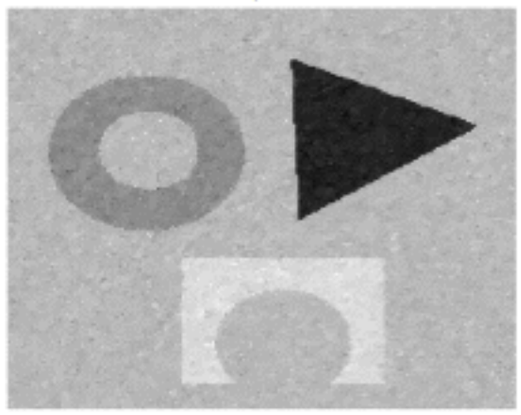
24/86

172/3116

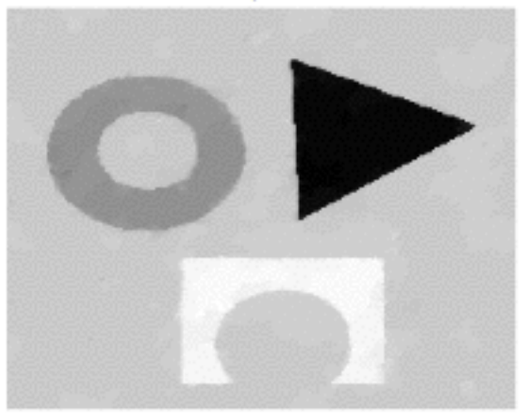
10/16

20/75

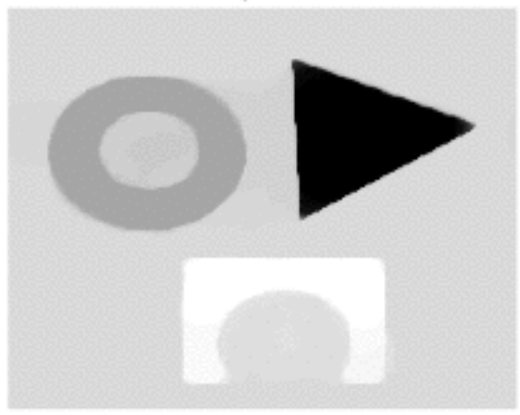
142/2149



10/16



25/97

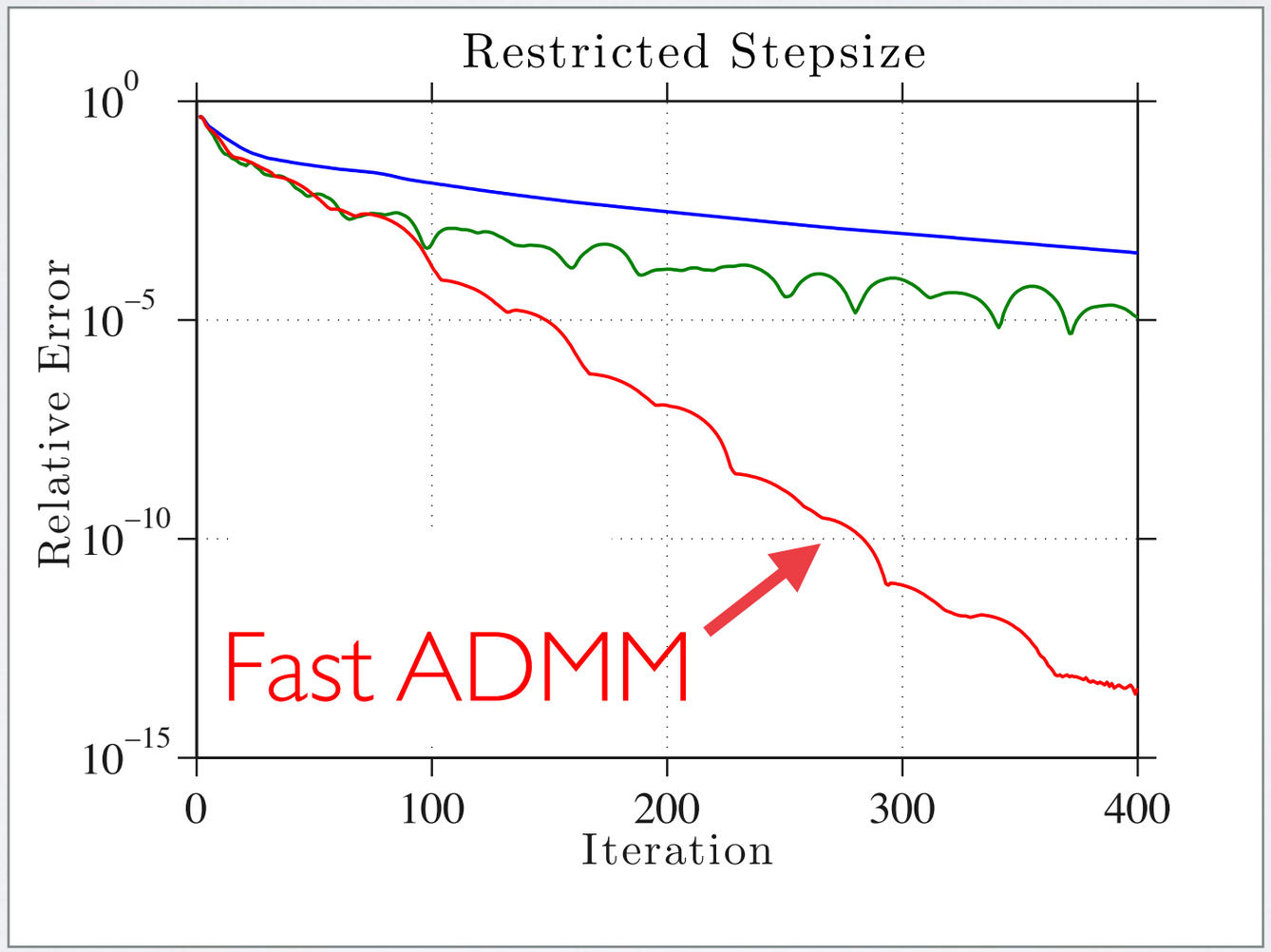


190/4990

MACHINE LEARNING: ELASTIC NET

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

Random A, Sparsity = 15/40



20X Faster

Goldstein, O'Donoghue, Setzer, Baraniuk. 2012

DISTRIBUTED MACHINE LEARNING

Consensus and Transpose Reduction Methods

WHY DISTRIBUTE?

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

CPUs

- Very scalable
- Cheap (cloud platforms)
- Communication cheap

Google compute engine

GPU

- Not so scalable
- Expensive
- Communication expensive

LINEAR CLASSIFIERS

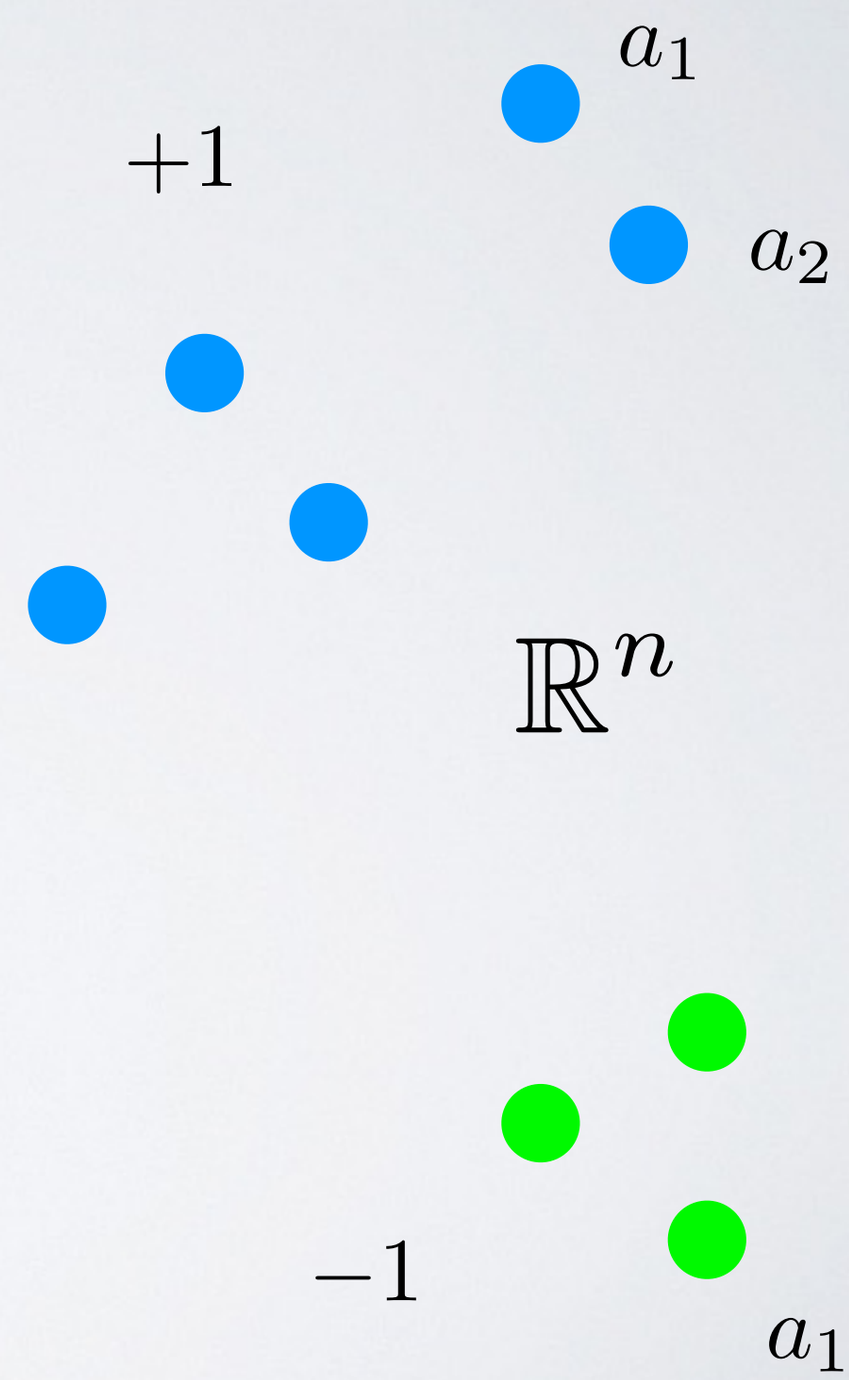
training data

feature vectors

vectors containing descriptions of objects

labels

+1/-1 labels indicating which "class" each vector lies in

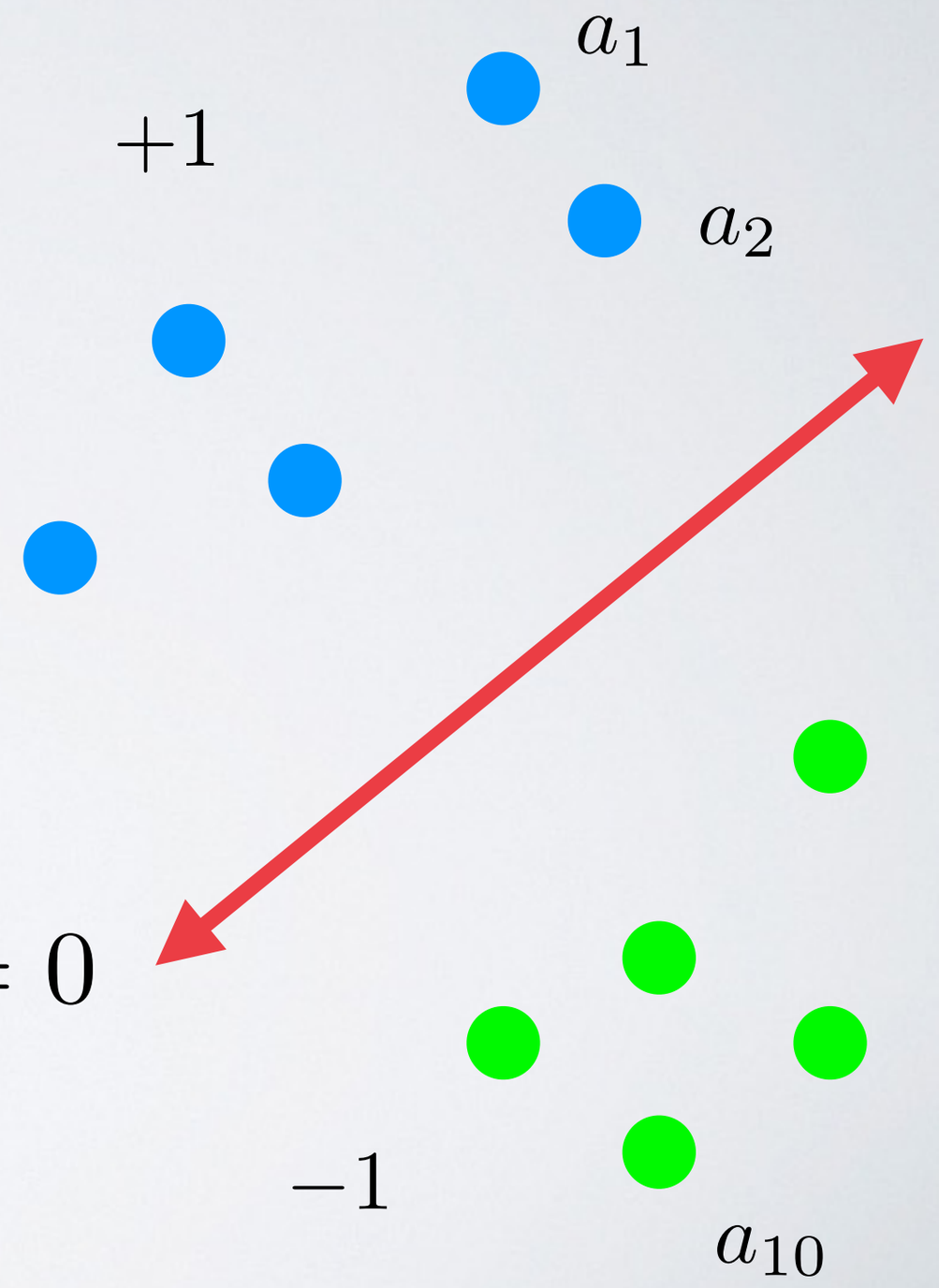


LINEAR CLASSIFIERS

goal

learn a line that separates two object classes

what line is best?

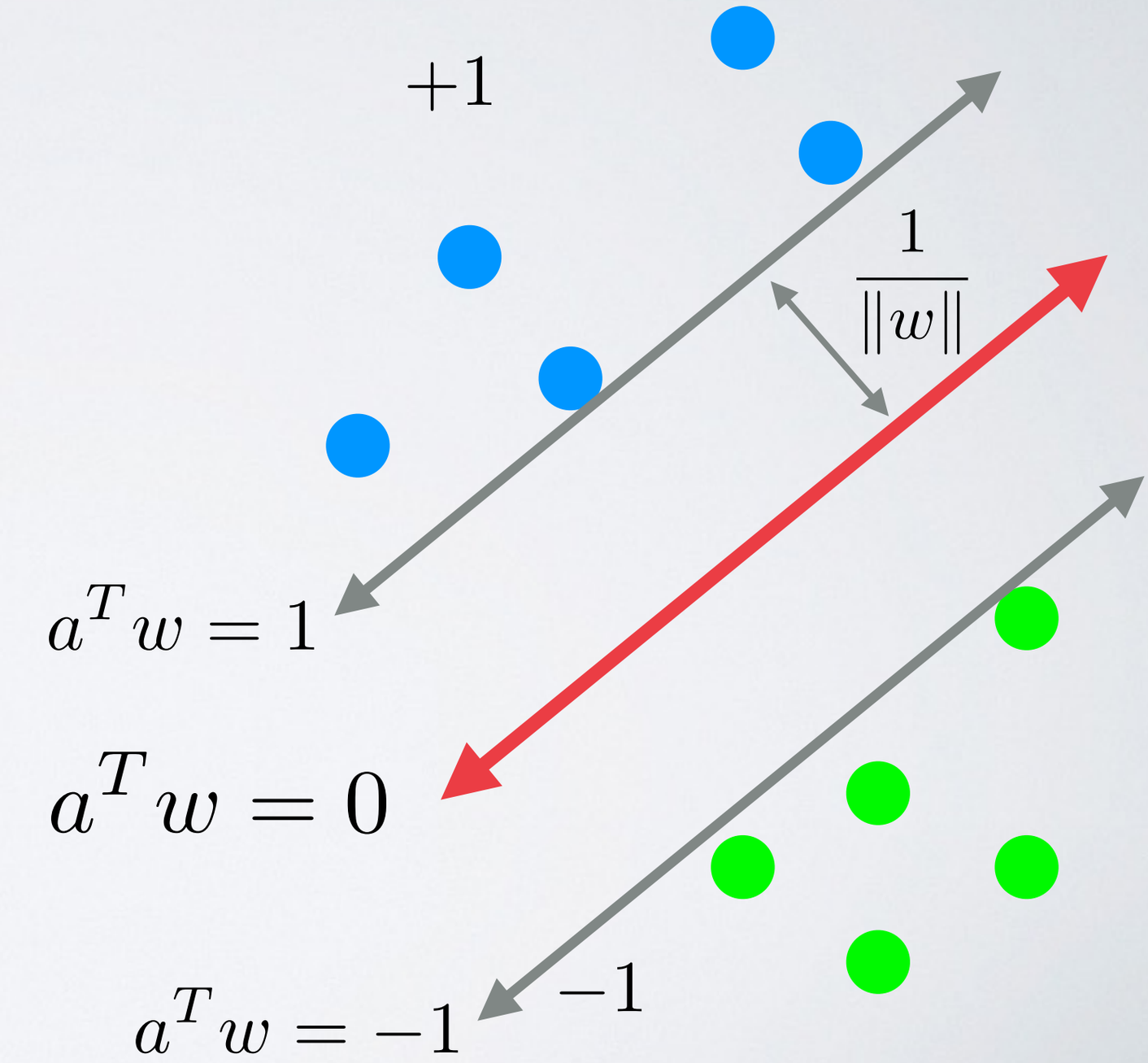


SUPPORT VECTOR MACHINE

SVM

choose line with
maximum "margin"

$$\text{margin width} = \frac{1}{\|w\|}$$



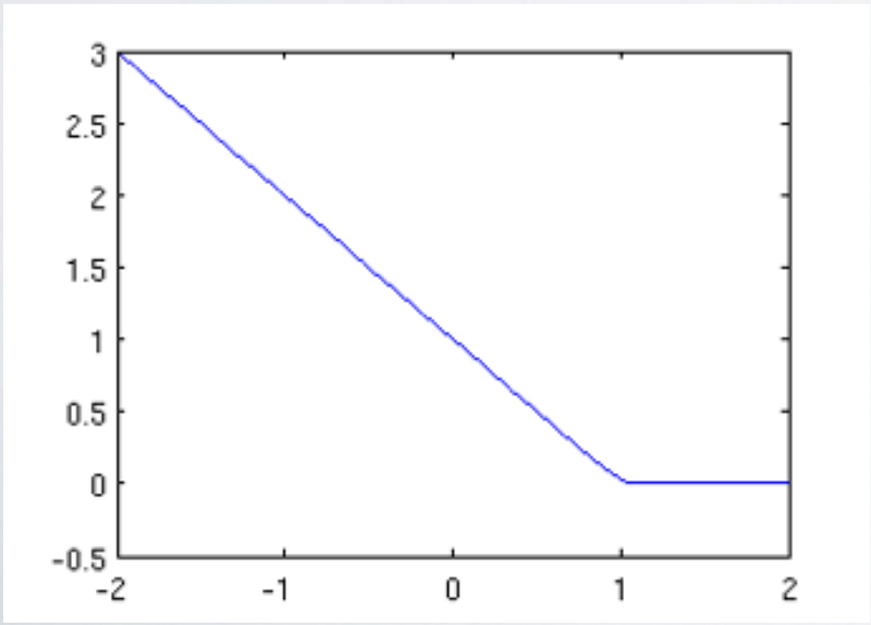
SUPPORT VECTOR MACHINE

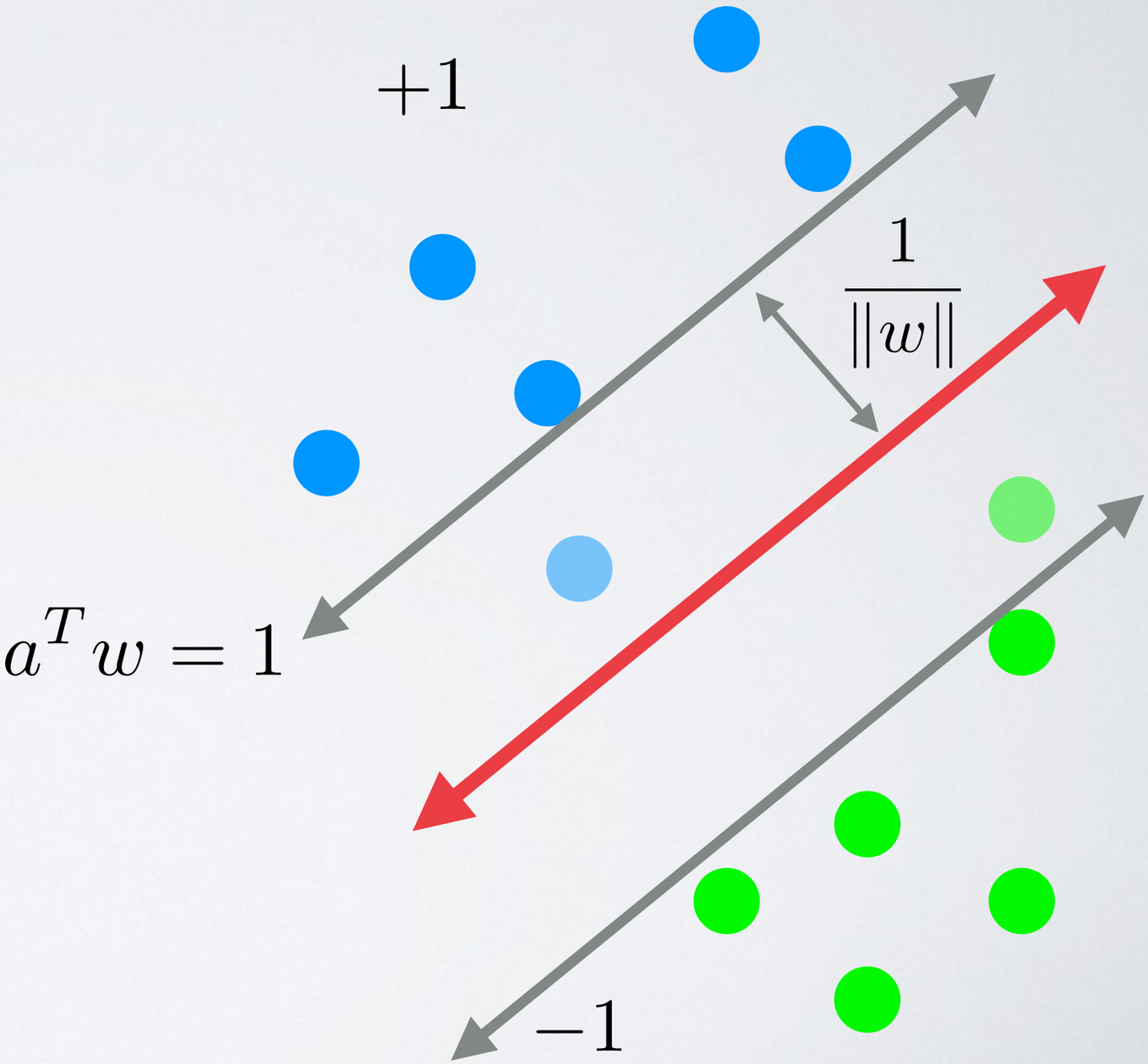
$$\text{margin width} = \frac{1}{\|w\|}$$

hinge loss

penalize points that lie within the margin

$$\sum_i h(a_i^T w)$$





SUPPORT VECTOR MACHINE

$$\text{margin width} = \frac{1}{\|w\|}$$

hinge loss

penalize points that lie within the margin

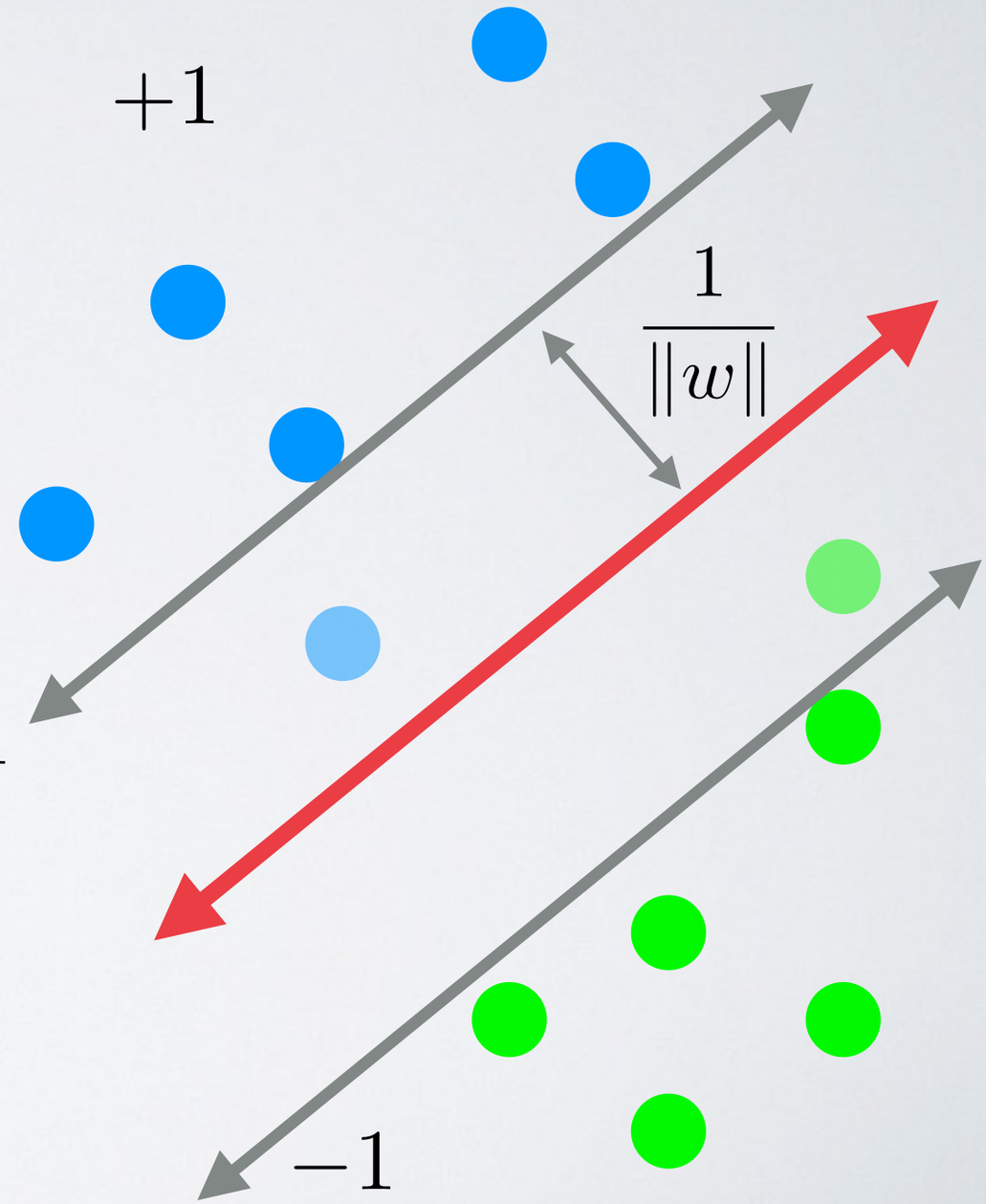
$$\sum_i h(a_i^T w)$$

short-hand

$$h(Aw)$$

combined objective

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



SCALED ADMM

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By + c = 0 \end{aligned}$$

augmented Lagrangian

$$L_\tau(x, y, \lambda) = f(x) + g(y) + \langle \lambda, Ax + By + c \rangle + \frac{\tau}{2} \|Ax + By + c\|^2$$

scaled Lagrangian

$$L_\tau(x, y, \lambda) = f(x) + g(y) + \frac{\tau}{2} \|Ax + By + c + \frac{1}{\tau}\lambda\|^2$$

These differ by a constant

SCALED ADMM

$$\begin{aligned} \text{minimize} \quad & f(x) + g(y) \\ \text{subject to} \quad & Ax + By + c = 0 \end{aligned}$$

scaled Lagrangian

$$L_\tau(x, y, \lambda) = f(x) + g(y) + \frac{\tau}{2}\|Ax + By + c + \frac{1}{\tau}\lambda\|^2$$

$$\lambda \leftarrow \lambda$$

$$L_\tau(x, y, \hat{\lambda}) = f(x) + g(y) + \frac{\tau}{2}\|Ax + By + c + \hat{\lambda}\|^2$$

scaled ADMM

$$x^{k+1} = \arg\min_x f(x) + \frac{\tau}{2}\|Ax + By^k + c + \hat{\lambda}^k\|^2$$

$$y^{k+1} = \arg\min_y g(y) + \frac{\tau}{2}\|Ax^{k+1} + By + c + \hat{\lambda}^k\|^2$$

$$\hat{\lambda}^{k+1} = \hat{\lambda}^k + Ax^{k+1} + By^{k+1} + c$$

DISTRIBUTED PROBLEMS

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

example: sparse least squares

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

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

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

data stored on different servers

CONSENSUS ADMM

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

Central server holds global variables: z

Every client gets local copy of unknowns: x_i

$$\begin{aligned} \text{minimize} \quad & g(z) + \sum_i f_i(x_i) \\ \text{subject to} \quad & x_i = z, \ \forall i \end{aligned}$$

Boyd et al. '10

CONSENSUS ADMM

$$\begin{aligned} \text{minimize} \quad & g(z) + \sum_i f_i(x_i) \\ \text{subject to} \quad & x_i = z, \ \forall i \end{aligned}$$

scaled augmented Lagrangian

$$L = g(z) + \sum_i f_i(x_i) + \sum_i \frac{\tau}{2} \|x_i - z + \lambda_i\|^2$$

consensus ADMM

$$\text{central server:} \ \ z^{k+1} = \arg\min_z g(z) + \sum_i \frac{\tau}{2} \|x_i^k - z + \lambda_i^k\|^2$$

$$\text{remote client:} \ \ x_i^{k+1} = \arg\min_{x_i} f_i(x_i) + \frac{\tau}{2} \|x_i - z^{k+1} + \lambda_i^k\|^2$$

$$\text{remote client:} \ \ \lambda_i^{k+1} = \lambda_i^k + x_i^{k+1} - z^{k+1}$$

EXAMPLE: LASSO

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

scaled augmented Lagrangian

$$L = \mu|z| + \sum_i \frac{1}{2}\|A_i x_i - b_i\|^2 + \sum_i \frac{\tau}{2}\|x_i - z + \lambda_i\|^2$$

consensus LASSO

$$\text{MPI reduce: } \eta^k = \frac{1}{N}\sum_i x_i^k + \lambda_i^k$$

$$\text{central server: } z^{k+1} = \arg\min_z \mu|z| + \frac{N\tau}{2}\|z + \eta\|^2$$

$$\text{remote client: } x_i^{k+1} = \arg\min_{x_i} \frac{1}{2}\|A_i x_i - b_i\|^2 + \frac{\tau}{2}\|x_i - z^{k+1} + \lambda_i^k\|^2$$

$$\text{remote client: } \lambda_i^{k+1} = \lambda_i^k + x_i^{k+1} - z^{k+1}$$

PROBLEMS WITH CONSENSUS

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

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

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

Works well for homogeneous data: everyone agrees

What about heterogenous data?

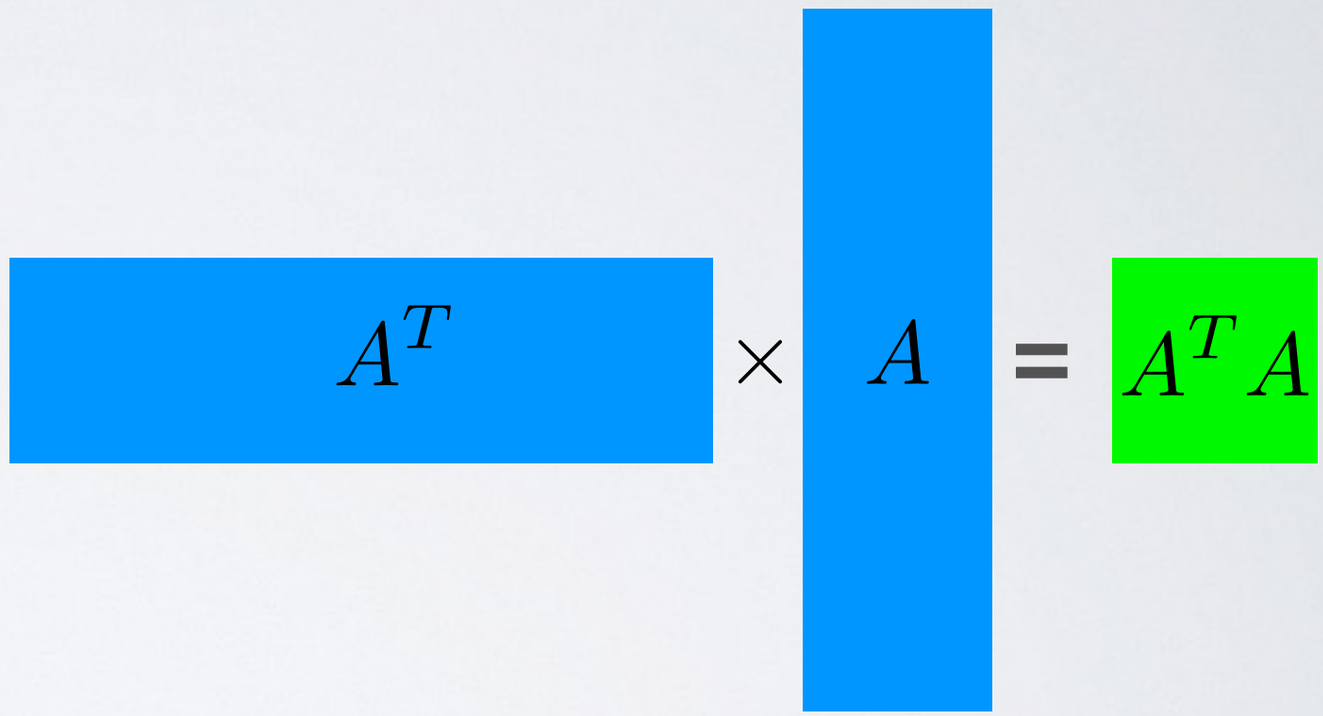
What about many cores?

TRANSPOSE REDUCTION

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

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

normal equations



TRANSPOSE REDUCTION

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

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

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

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

Least squares: use TR here

scaled augmented Lagrangian

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

DISTRIBUTED VERSION

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

"unwrapped" form

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

scaled augmented Lagrangian

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

DISTRIBUTED STEPS

scaled augmented Lagrangian

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

setup phase

Form: $A^T A = \sum_i A_i^T A_i$

remote servers: z-update

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

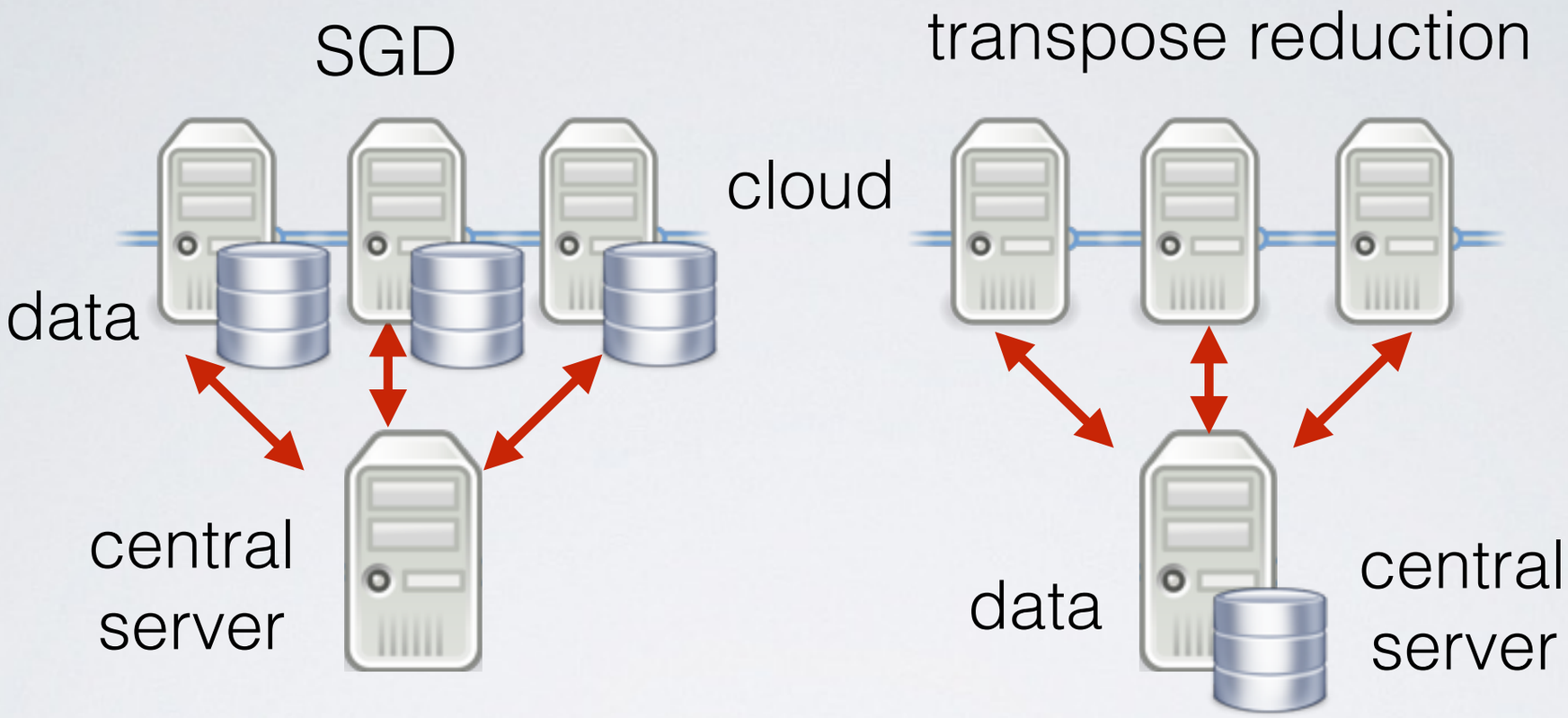
central servers: x-update

$$x^{k+1} = \underset{x}{\text{minimize}} \quad \frac{1}{2}\|x\|^2 + \frac{\tau}{2}\|z_i^{k+1} - A_i x + \lambda_i^k\|^2$$

remote servers: lambda-update

$$\lambda_i^{k+1} = \lambda_i^k + z_i^{k+1} - A_i x^{k+1}$$

TRANSPOSE REDUCTION

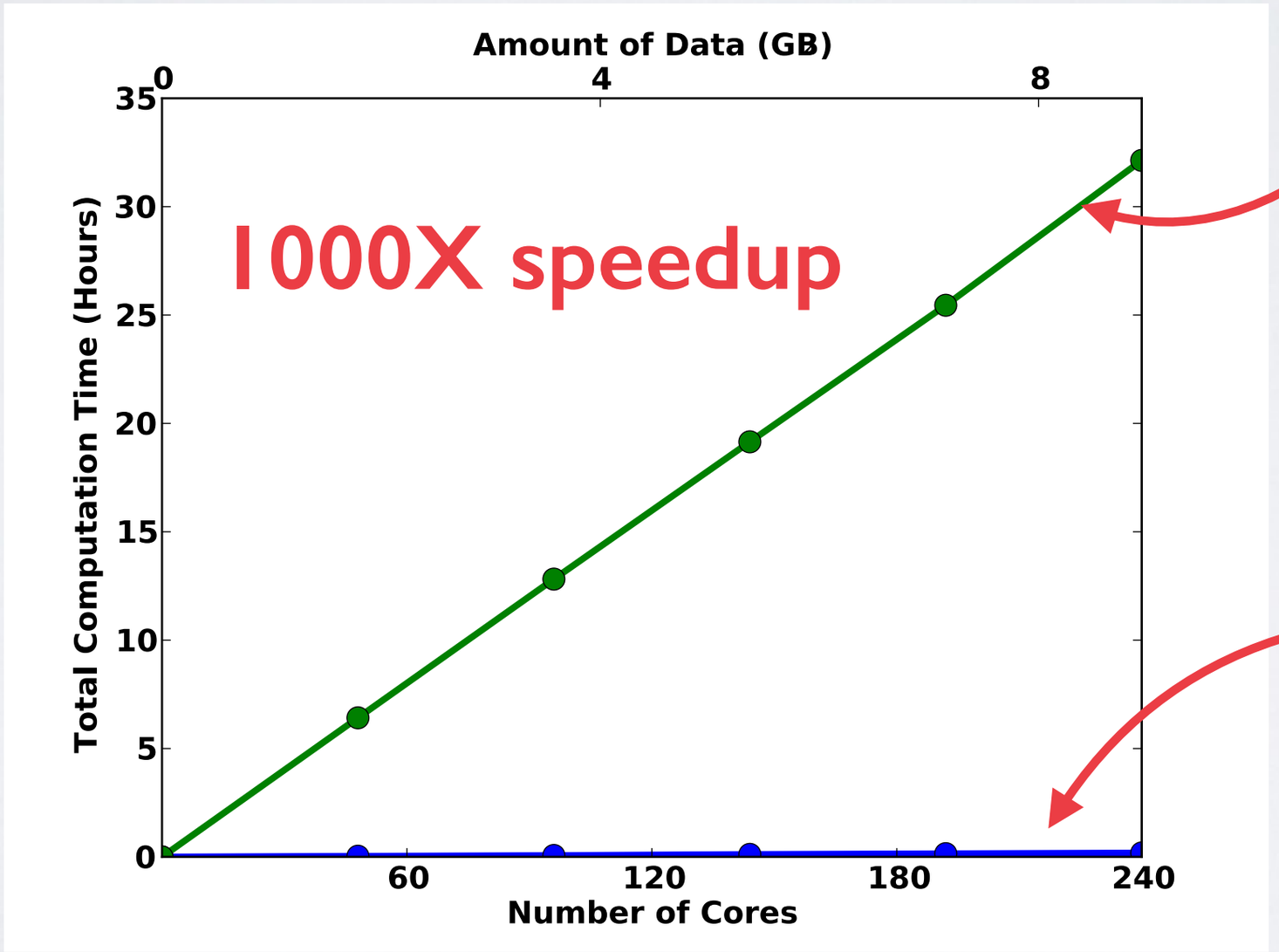


Transpose reduction ADMM

- servers work together to find one solution
- central server makes simple queries
- fits classifier (SVM) on 7TB dataset in 15 seconds using 7500 cores

EXPERIMENT: SVM

2K features, 50K data points/core



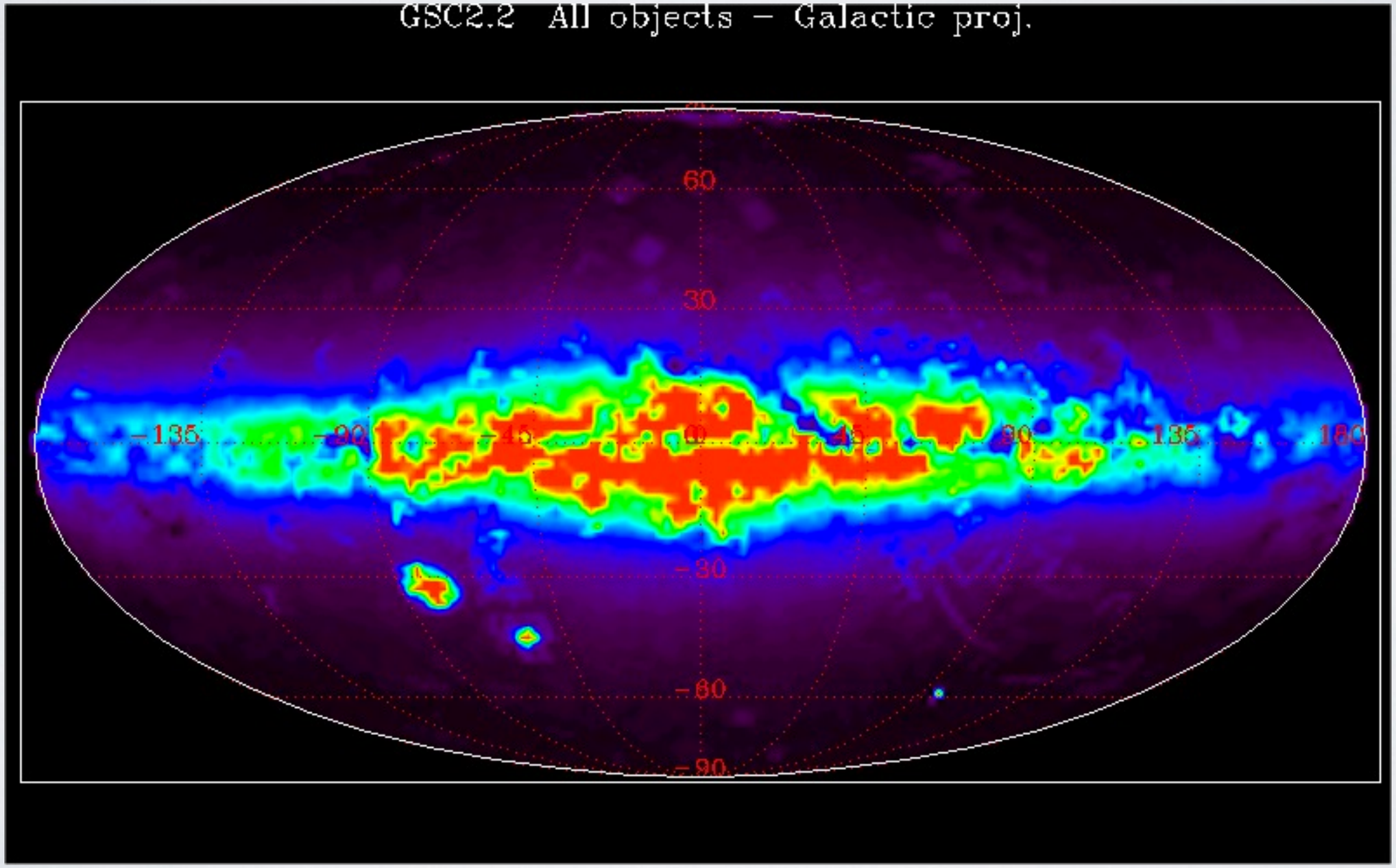
consensus

transpose reduction

Goldstein & Taylor '15

GUIDE STAR CATALOG II

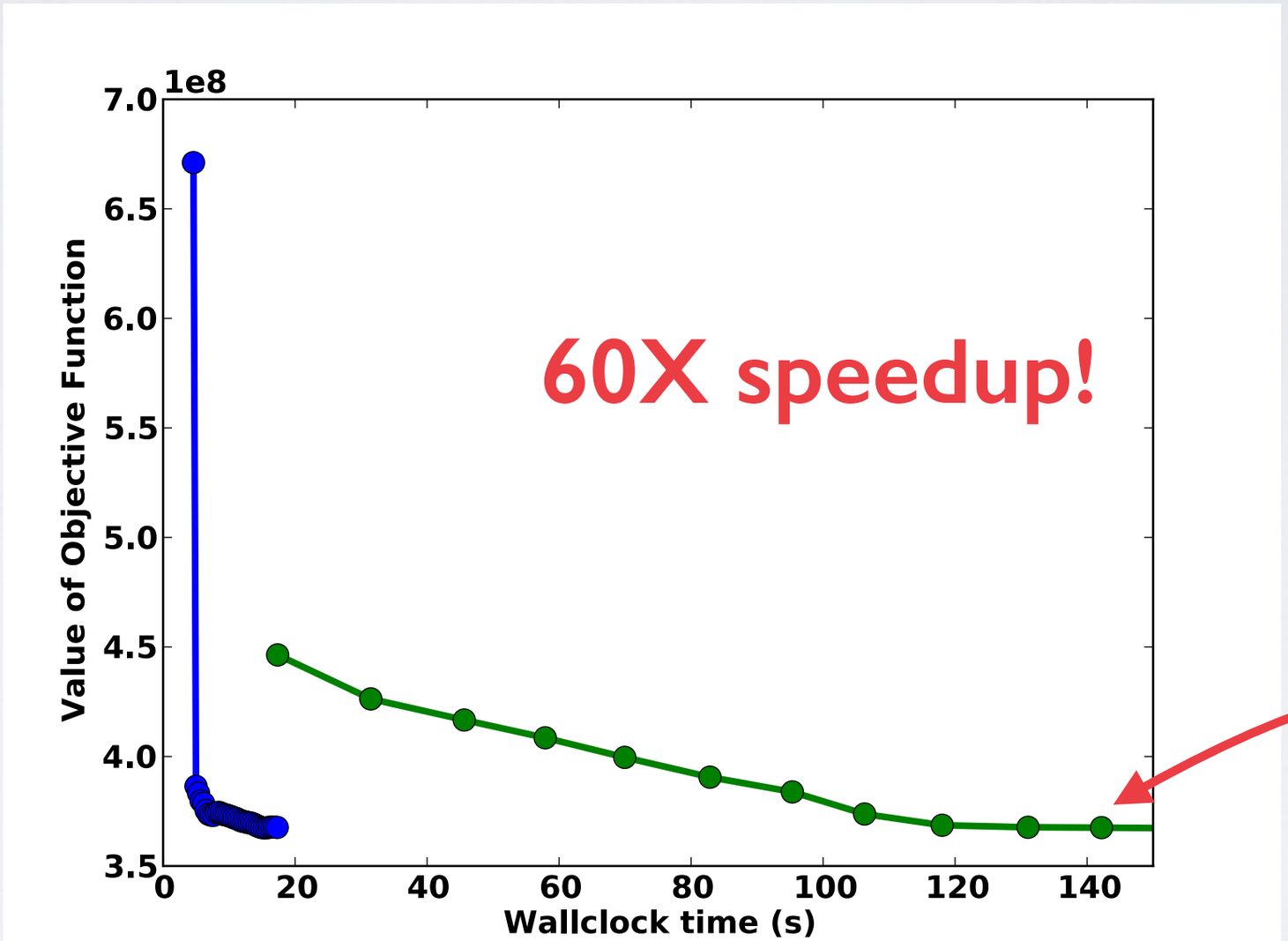
All known objects from Palomar and UK Schmidt surveys
About 2 billion objects



H.Jenkner, B.Lasker, '90

USNO GUIDE STAR DATABASE

960M features vectors (1.8TB), 2500 cores

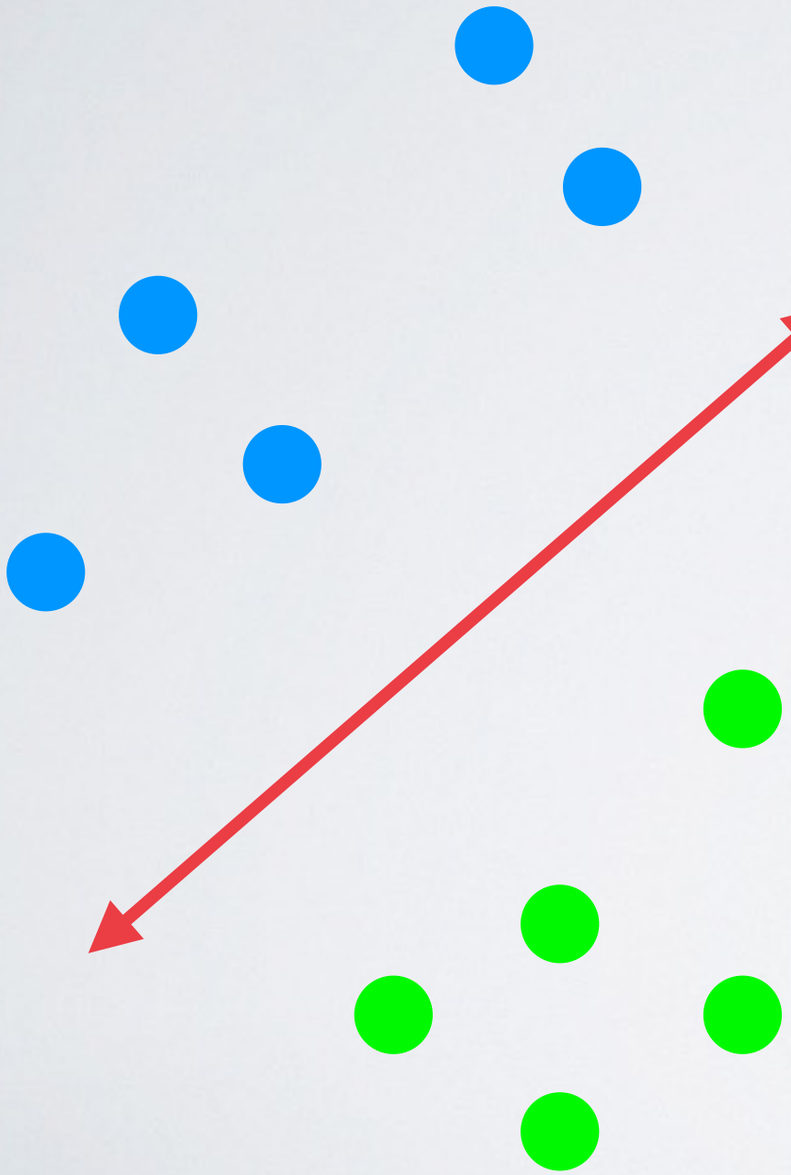


consensus does not converge until t=1180 sec

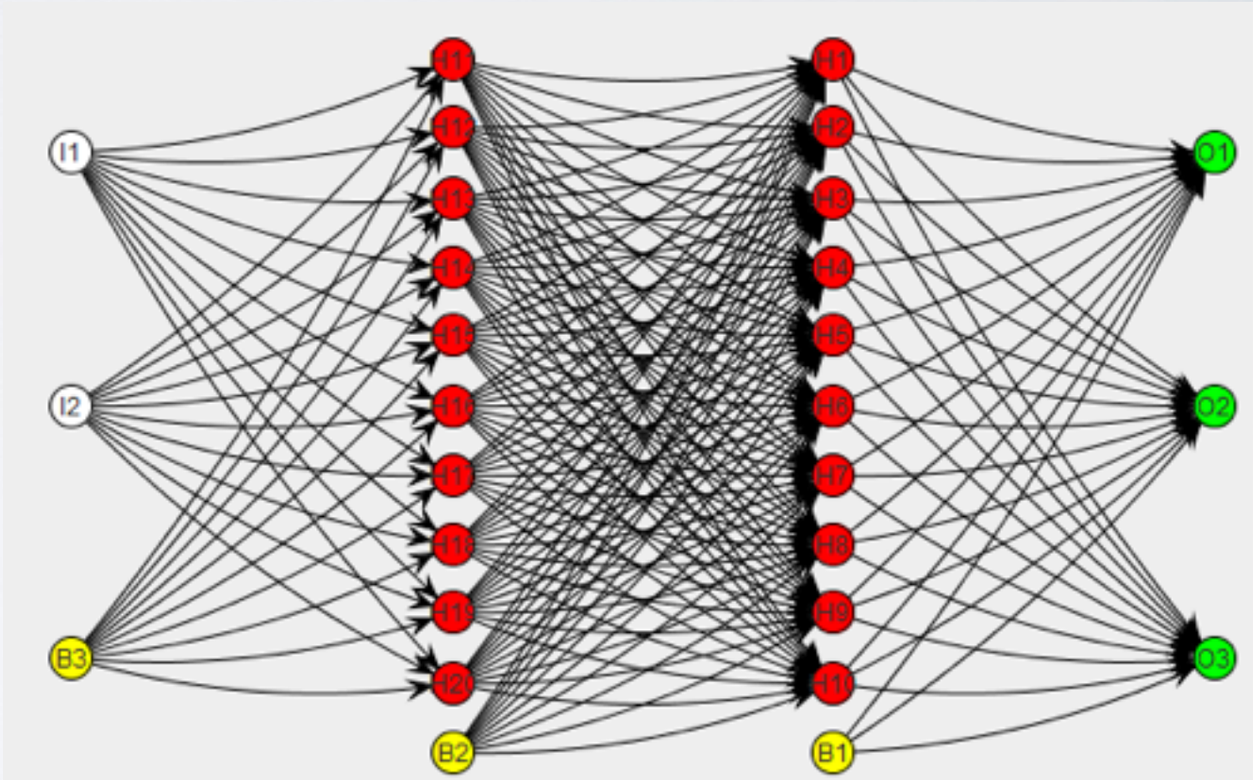
Goldstein & Taylor '15

MORE COMPLEX PROBLEMS

GLMs & Linear classifiers



Neural nets



ADMM FOR NEURAL NETS

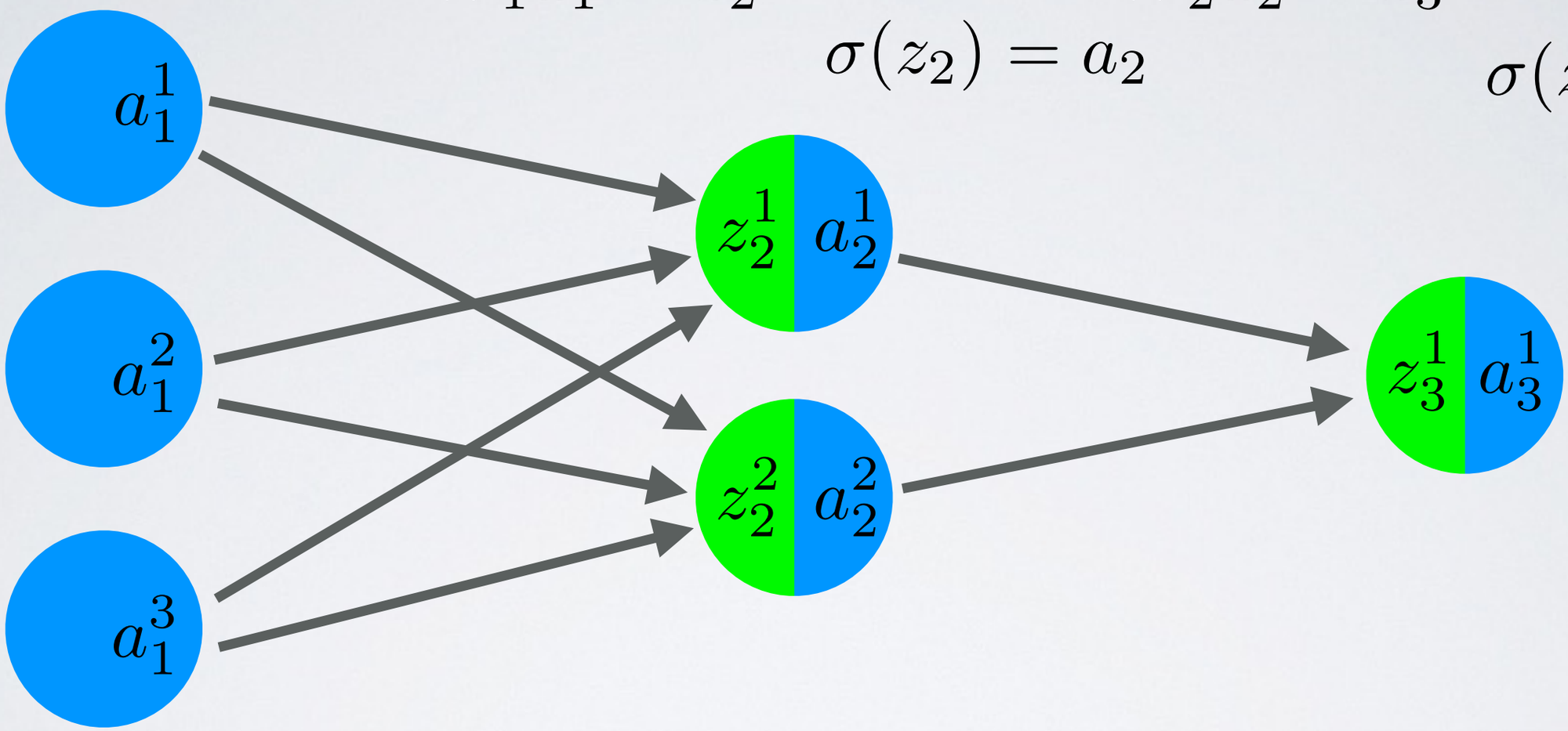
a_1

$W_1 a_1 = z_2$

$\sigma(z_2) = a_2$

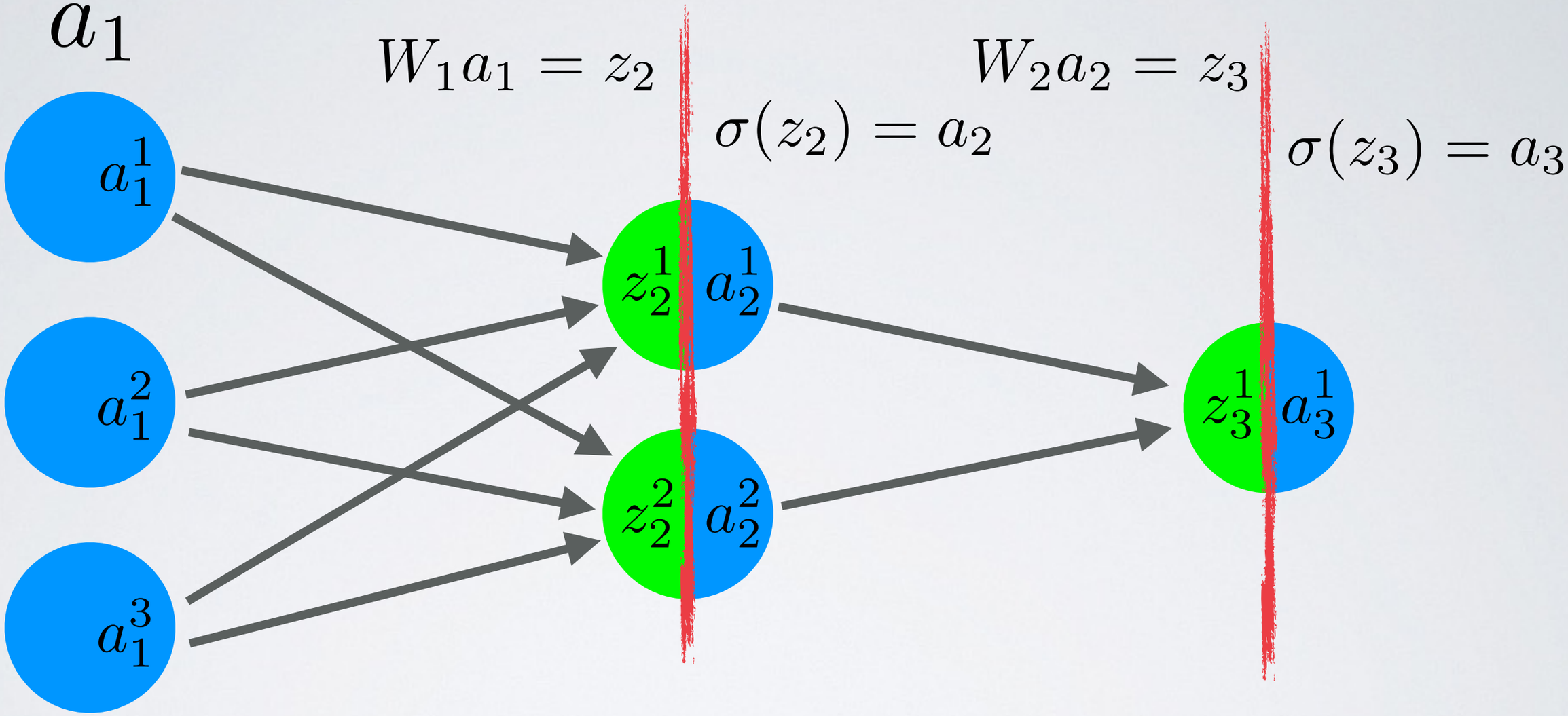
$W_2 a_2 = z_3$

$\sigma(z_3) = a_3$



minimize $\ell(a_3)$

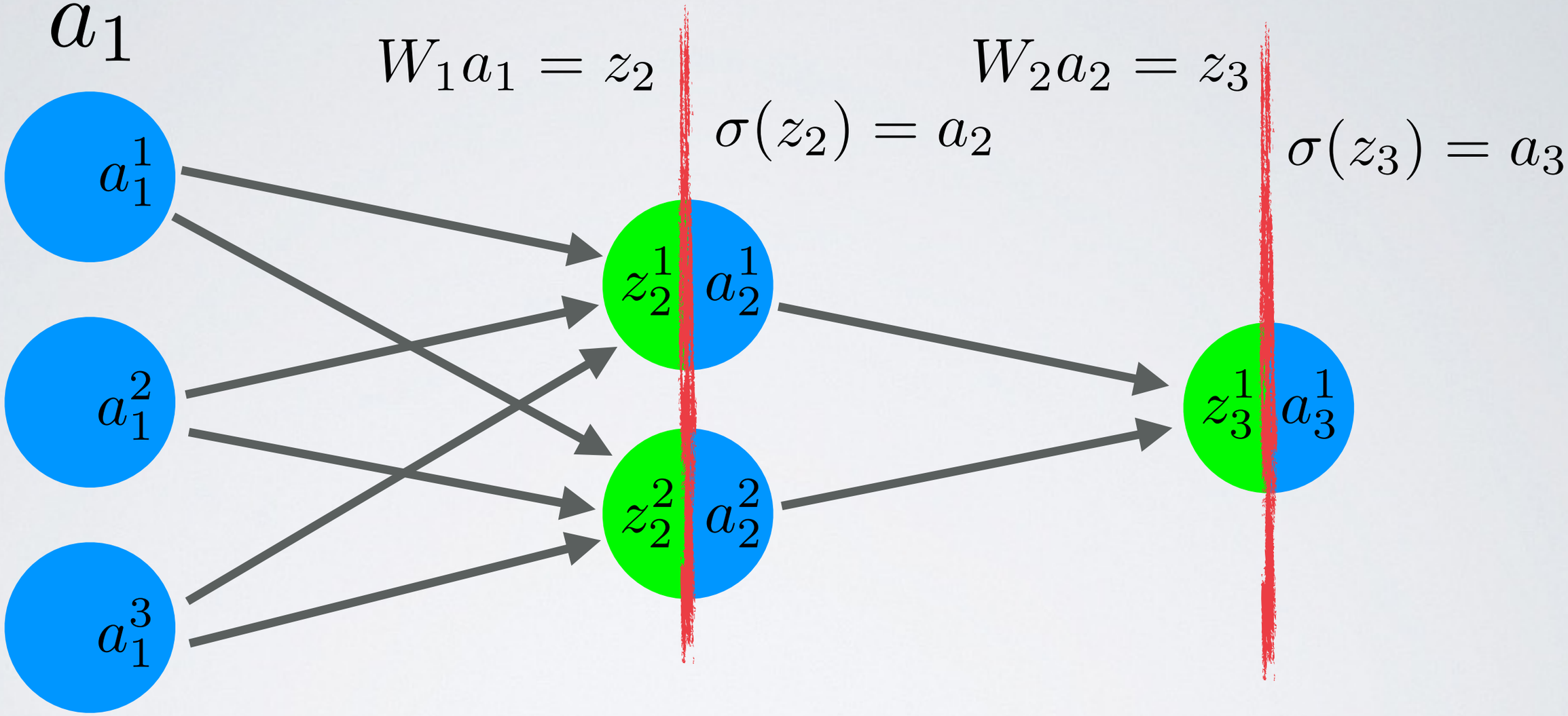
ADMM FOR NEURAL NETS



$$\text{minimize} \quad \ell(a_3)$$

$$\text{subject to} \quad z_2 = W_1 a_1, \quad a_2 = \sigma(z_2)$$
$$z_3 = W_2 a_2, \quad a_3 = \sigma(z_3)$$

ADMM FOR NEURAL NETS



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

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**

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

LAGRANGE MULTIPLIERS

Classical ADMM

$$\begin{aligned}&\text{minimize} \quad \ell(a_3)\\ &+\frac{1}{2}\|z_2 - W_1 a_1\|^2 + \frac{1}{2}\|a_2 - \sigma(z_2)\|^2 + \langle \lambda_1, z_2 - W_1 a_1\rangle + \langle \lambda_2, \langle a_2 - \sigma(z_2)\rangle\\ &+\frac{1}{2}\|z_3 - W_2 a_2\|^2 + \frac{1}{2}\|a_3 - \sigma(z_3)\|^2 + \langle \lambda_3, z_3 - W_2 a_2\rangle + \langle \lambda_4, a_3 - \sigma(z_3)\rangle\end{aligned}$$

…unstable because of non-linear constraints

Bregman Iteration

multiplier term

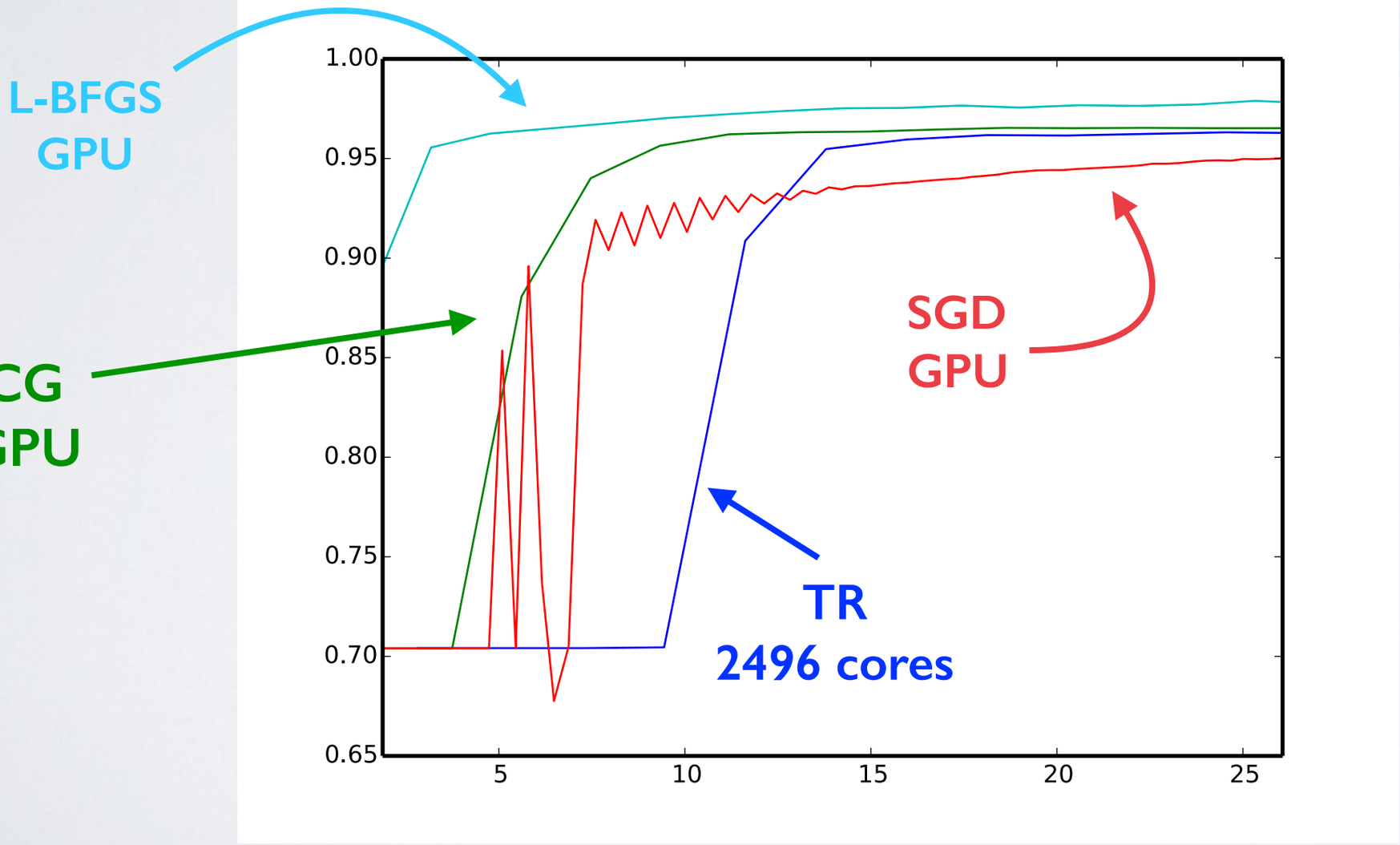
$$\begin{aligned}&\text{minimize} \quad \ell(a_3) + \langle \lambda, a_3\rangle\\ &+\frac{1}{2}\|z_2 - W_1 a_1\|^2 + \frac{1}{2}\|a_2 - \sigma(z_2)\|^2\\ &+\frac{1}{2}\|z_3 - W_2 a_2\|^2 + \frac{1}{2}\|a_3 - \sigma(z_3)\|^2\end{aligned}$$

NEURAL NETS

2 hidden layers ~150K weights

"Street View" house number classification: 120K data

2496 core ADMM vs GPU (K40, 2880 cores)



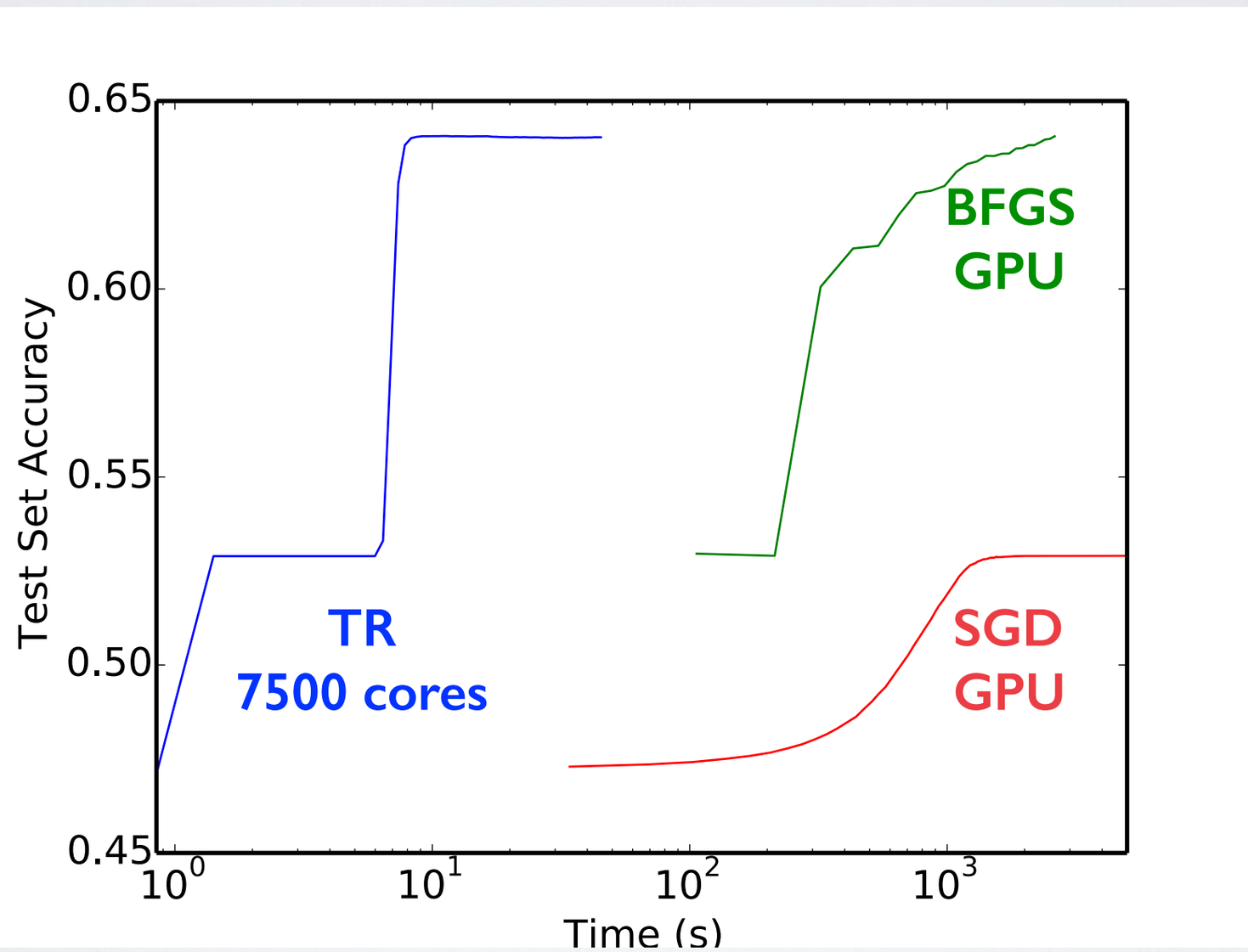
"Training Neural Networks Without Gradients: A Scalable ADMM Approach."
Taylor, Burmeister, Xu, Singh, Patel, Goldstein. ICML 2016

NEURAL NETS

3 hidden layers ~ 300K weights

"Higgs" particle classification: 10.5M points

7500 core ADMM vs GPU (K40, 2880 cores)



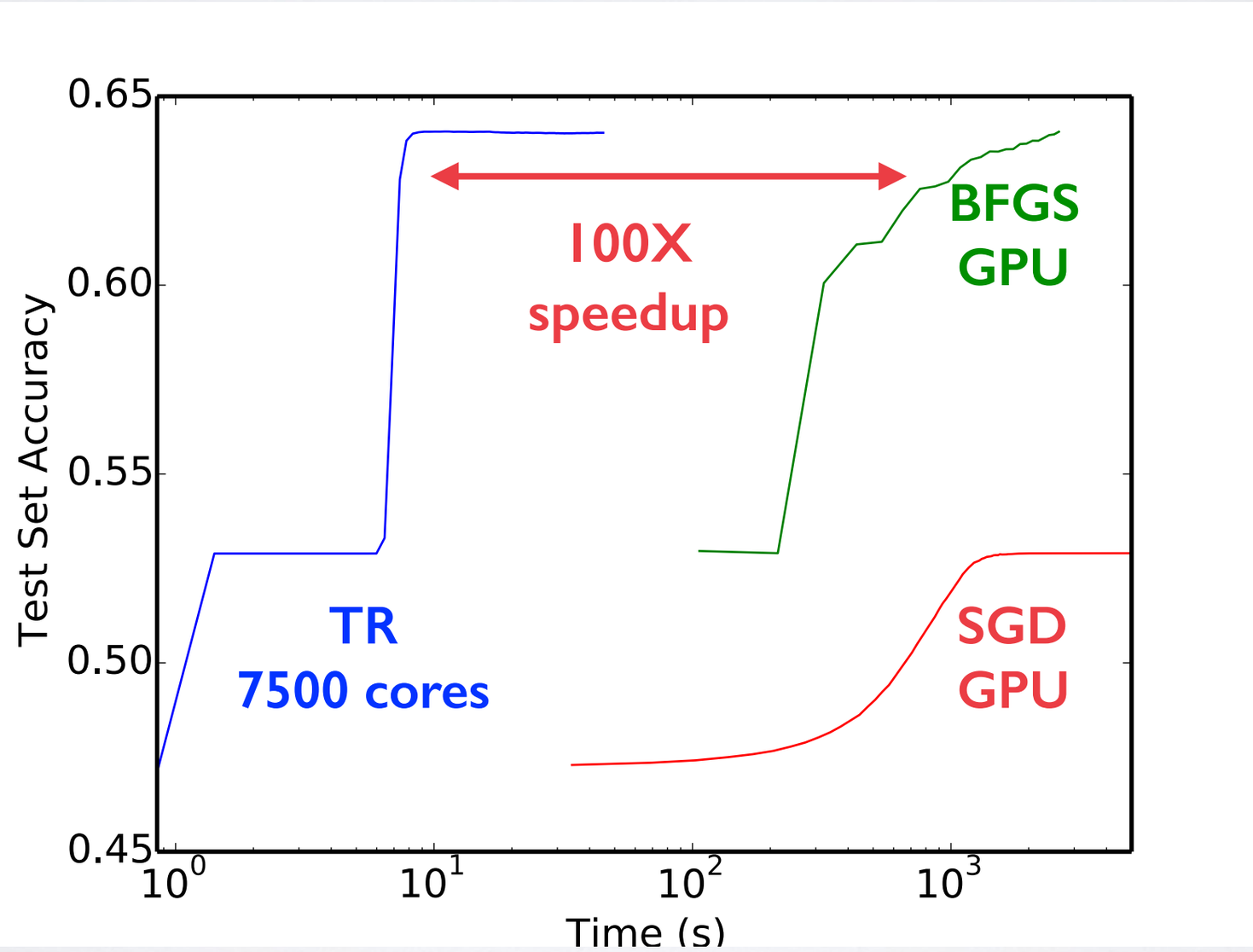
"Training Neural Networks Without Gradients: A Scalable ADMM Approach."
Taylor, Burmeister, Xu, Singh, Patel, Goldstein. ICML 2016

NEURAL NETS

3 hidden layers ~ 300K weights

"Higgs" particle classification: 10.5M points

7500 core ADMM vs GPU (K40, 2880 cores)



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

ADAPTIVE METHODS

$$\text{minimize} \quad f(x)$$

$$x^{k+1} = x^k - \tau \nabla f(x^k)$$

how to choose?

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

Augmented Lagrangian

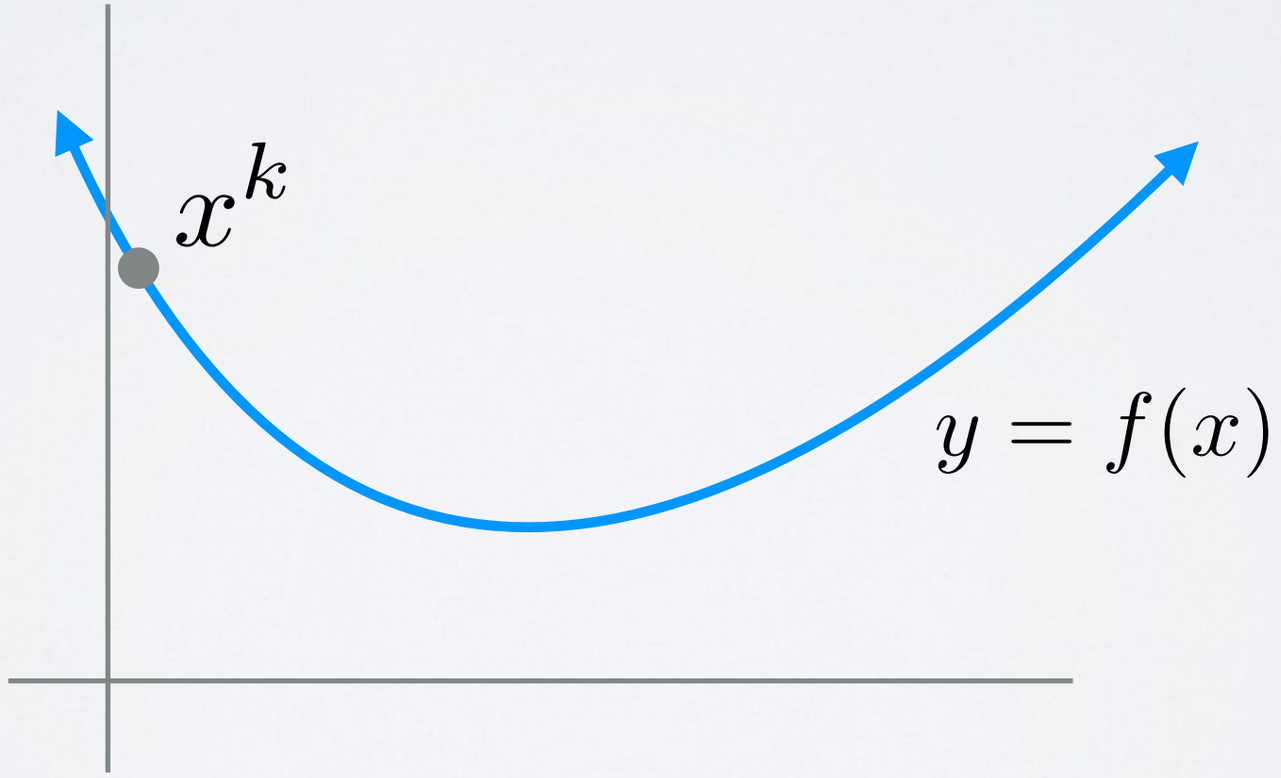
$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \langle \lambda, b - Au - Bv \rangle + \frac{\tau}{2} \|b - Au - Bv\|^2$$

SPECTRAL STEPSIZE

$$\text{minimize} \quad f(x)$$

$$x^{k+1} = x^k - \tau \nabla f(x^k)$$

"Spectral" approximation



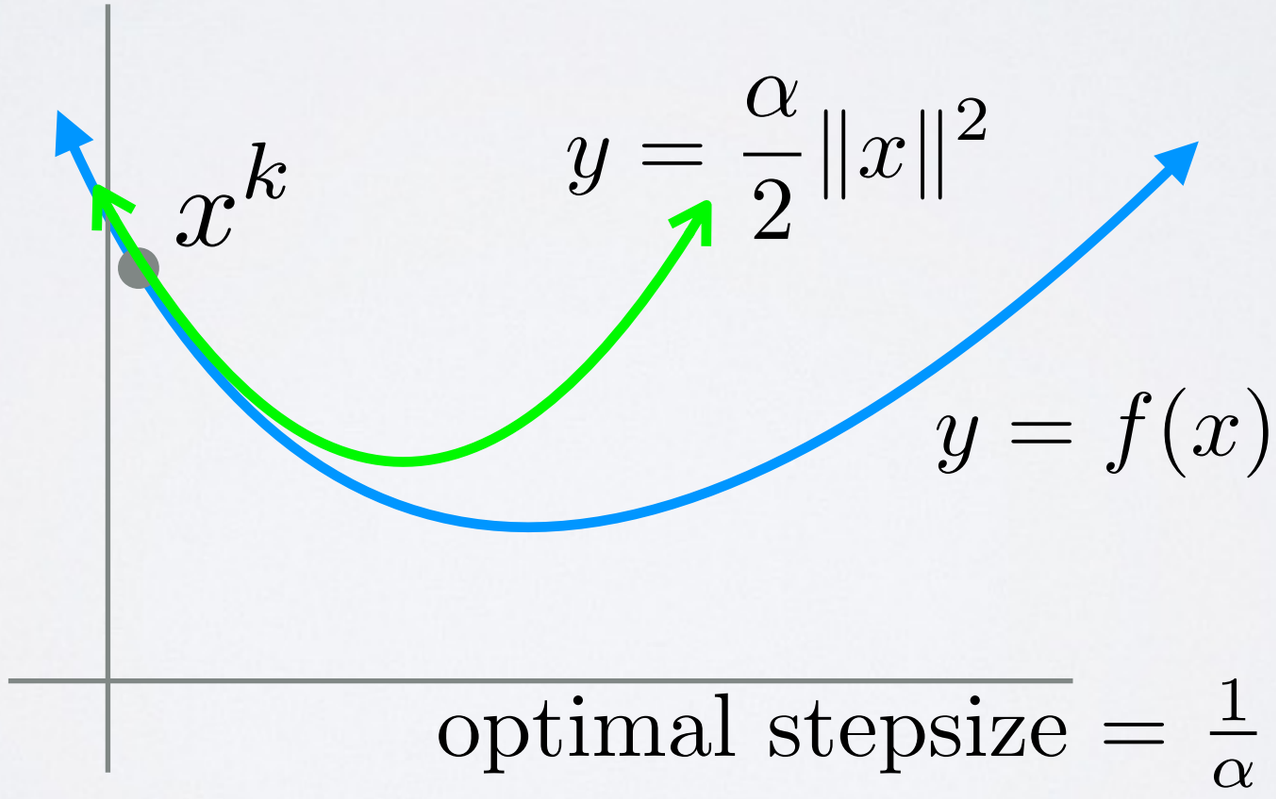
Barzilai & Borwein, "Two-point stepsize gradient methods," 1988

SPECTRAL STEPSIZE

$$\text{minimize} \quad f(x)$$

$$x^{k+1} = x^k - \tau \nabla f(x^k)$$

"Spectral" approximation



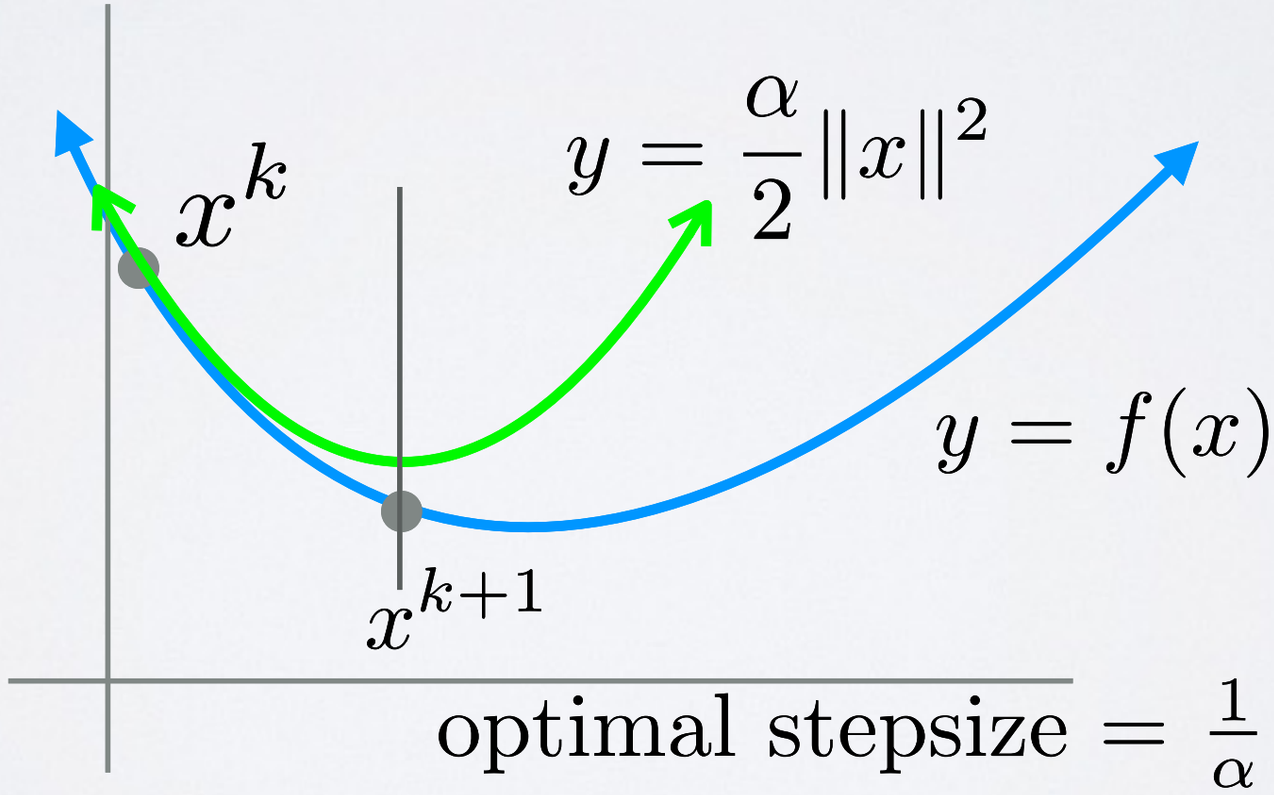
Barzilai & Borwein, "Two-point stepsize gradient methods," 1988

SPECTRAL STEPSIZE

$$\text{minimize} \quad f(x)$$

$$x^{k+1} = x^k - \tau \nabla f(x^k)$$

"Spectral" approximation



Barzilai & Borwein, "Two-point stepsize gradient methods," 1988

HOW TO GET STEPSIZE

$$\text{minimize} \quad f(x)$$

$$x^{k+1} = x^k - \tau \nabla f(x^k)$$

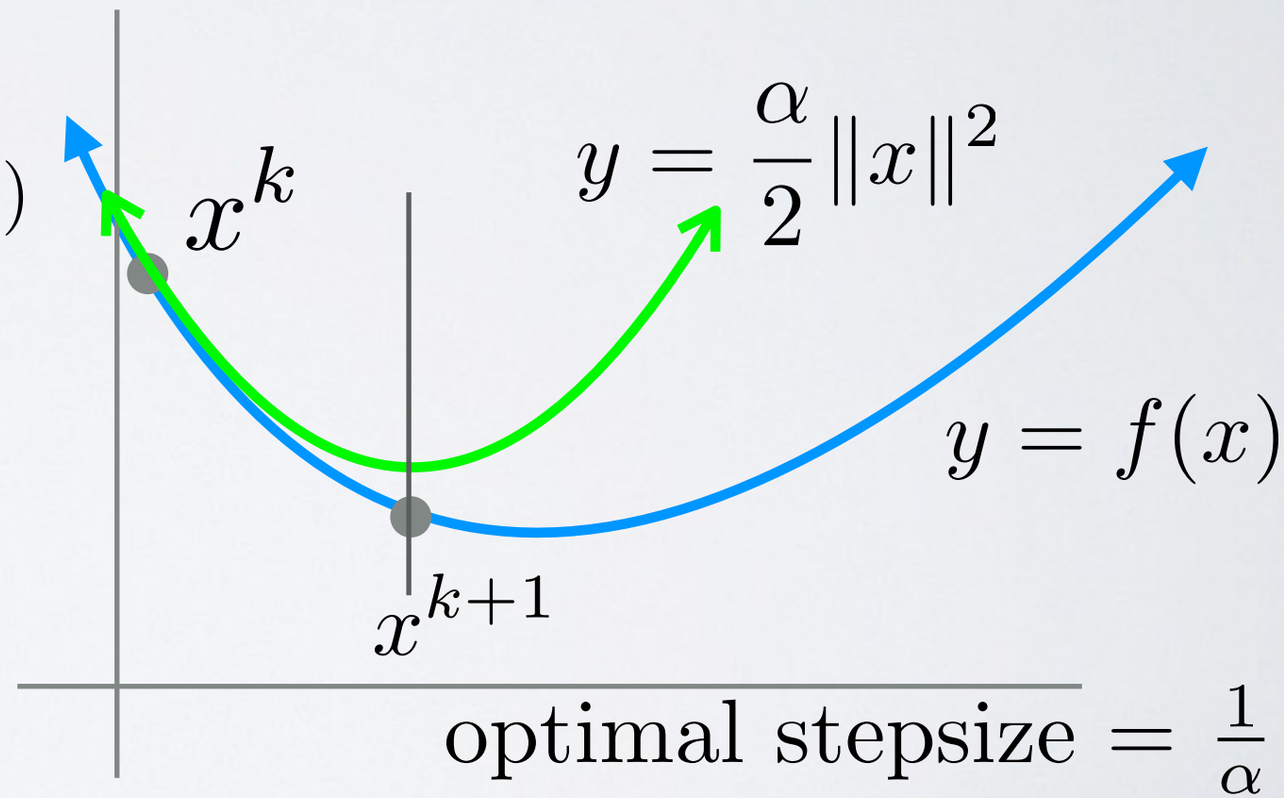
from quadratic model

$$\nabla f(x) = \alpha x$$

$$\nabla f(x^{k+1}) - \nabla f(x^k) = \alpha(x^{k+1} - x^k)$$

least squares solution

$$\alpha = \frac{(x^{k+1} - x^k)^T (\nabla f(x^{k+1}) - \nabla f(x^k))}{\|x^{k+1} - x^k\|^2}$$



Barzilai & Borwein, “Two-point stepsize gradient methods,” 1988

HOW TO GET STEPSIZE

$$\text{minimize} \quad f(x)$$

$$x^{k+1} = x^k - \tau \nabla f(x^k)$$

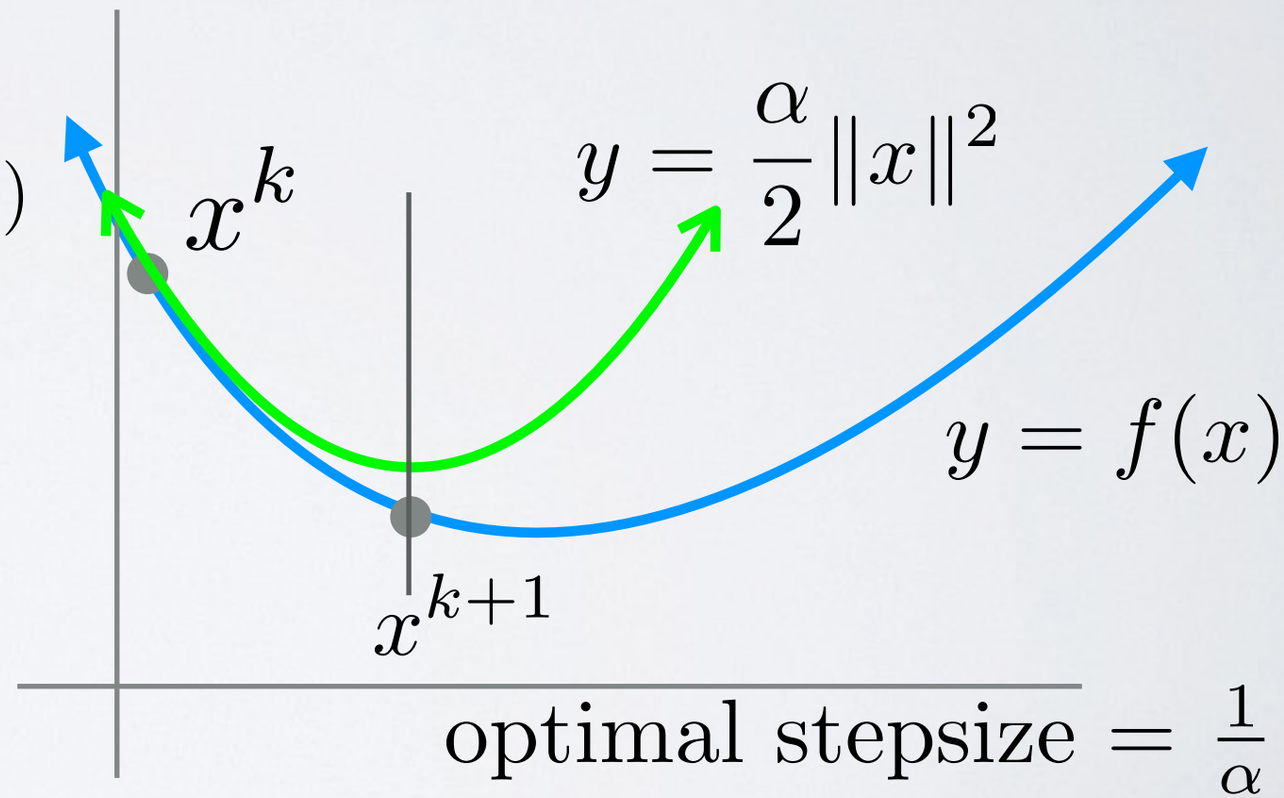
from quadratic model

$$\nabla f(x) = \alpha x$$

$$\nabla f(x^{k+1}) - \nabla f(x^k) = \alpha(x^{k+1} - x^k)$$

least squares solution

$$\alpha = \frac{(x^{k+1} - x^k)^T (\nabla f(x^{k+1}) - \nabla f(x^k))}{\|x^{k+1} - x^k\|^2}$$



Barzilai & Borwein, “Two-point stepsize gradient methods,” 1988

FORWARD-BACKWARD METHOD

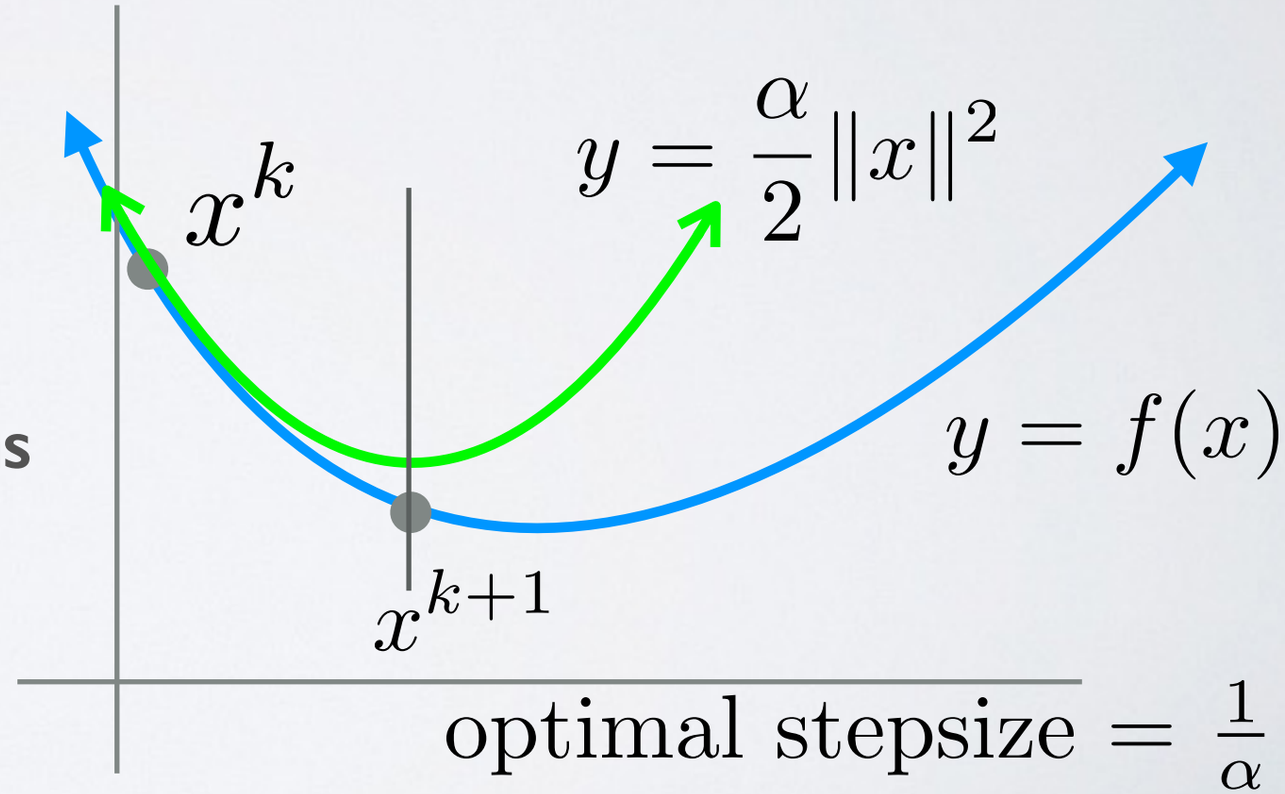
$$\text{minimize} \quad f(x)+g(x)$$

$$\hat{x}^{k+1} = x^k - \tau \nabla f(x^k)$$

$$x^{k+1} = \arg\min g(x) + \frac{1}{2\tau}\|x - \hat{x}^{k+1}\|^2$$

Advantages

- **Fast! Superlinear for some problems**
- **Automated**
- **Can solve non-differentiable problems**
- **Can handle simple set constraints**



FASTA: Fasta Adaptive Shrinkage Thresholding Algorithm

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

You provide: $\nabla f(x)$ $\text{prox}_g(x)$

FASTA does the rest!

stepsize choice!
acceleration!
stopping conditions!

preconditioners!
line search!
converge theory!

Solves: L1 least squares, sparse classification, matrix completion, democratic representations, total variation, semidefinite programs, etc…

FASTA: Fasta Adaptive Shrinkage Thresholding Algorithm

Paper: A field guide to forward backward splitting with a FASTA implementation

T Goldstein, C Studer, R Baraniuk

Solves: L1 least squares, sparse classification, matrix completion, democratic representations, total variation, semidefinite programs, etc…

ADAPTIVE ADMM

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

Lagrangian

$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \lambda^T (Au + Bv - b)$$

$$\max_{\lambda} \min_{u,v} \underbrace{[H(u) + \lambda^T Au]}_{H^*(\lambda)} + \underbrace{[G(v) + \lambda^T Bv]}_{G^*(\lambda)} - \lambda^T b$$

ADAPTIVE ADMM

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

Lagrangian

$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \lambda^T (Au + Bv - b)$$

dual problem: no constraints

$$\min_{\lambda} H^*(A^T\lambda) - \langle \lambda, b \rangle + G^*(B^T\lambda)$$

ADAPTIVE ADMM

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

Lagrangian

$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \lambda^T (Au + Bv - b)$$

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

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

Lagrangian

$$\max_{\lambda} \min_{u,v} H(u) + G(v) + \lambda^T (Au + Bv - b)$$

curvatures are "free" given ADMM iterates

dual problem: no constraints

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

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

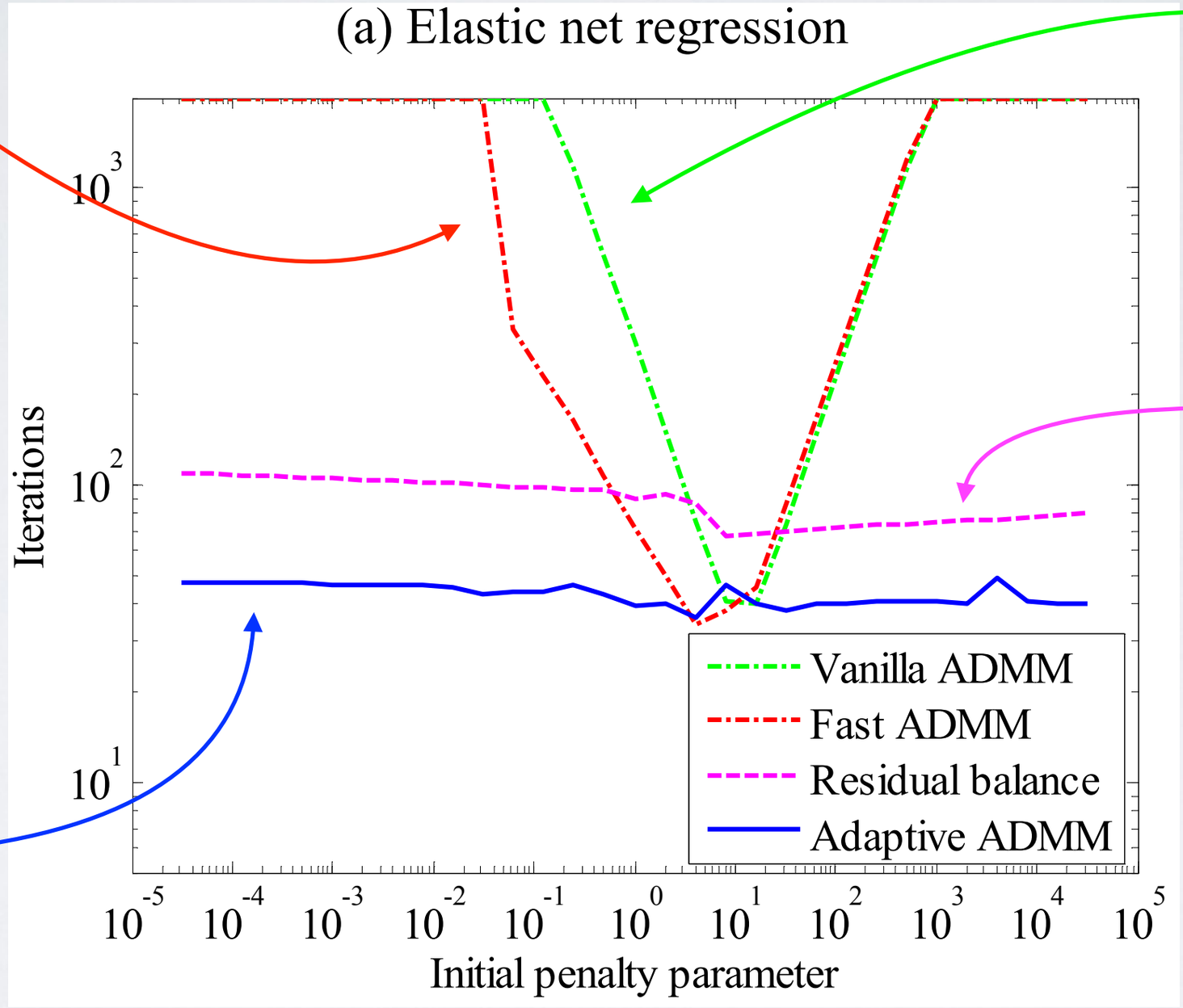
$$\alpha = \frac{(\hat{\lambda}_k - \hat{\lambda}_0)^T A(u_k - u_{k_0})}{\|\hat{\lambda}_k - \hat{\lambda}_0\|^2}$$

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

DEPENDENCE ON STEPSIZE GUESS

Fast ADMM

Vanilla ADMM

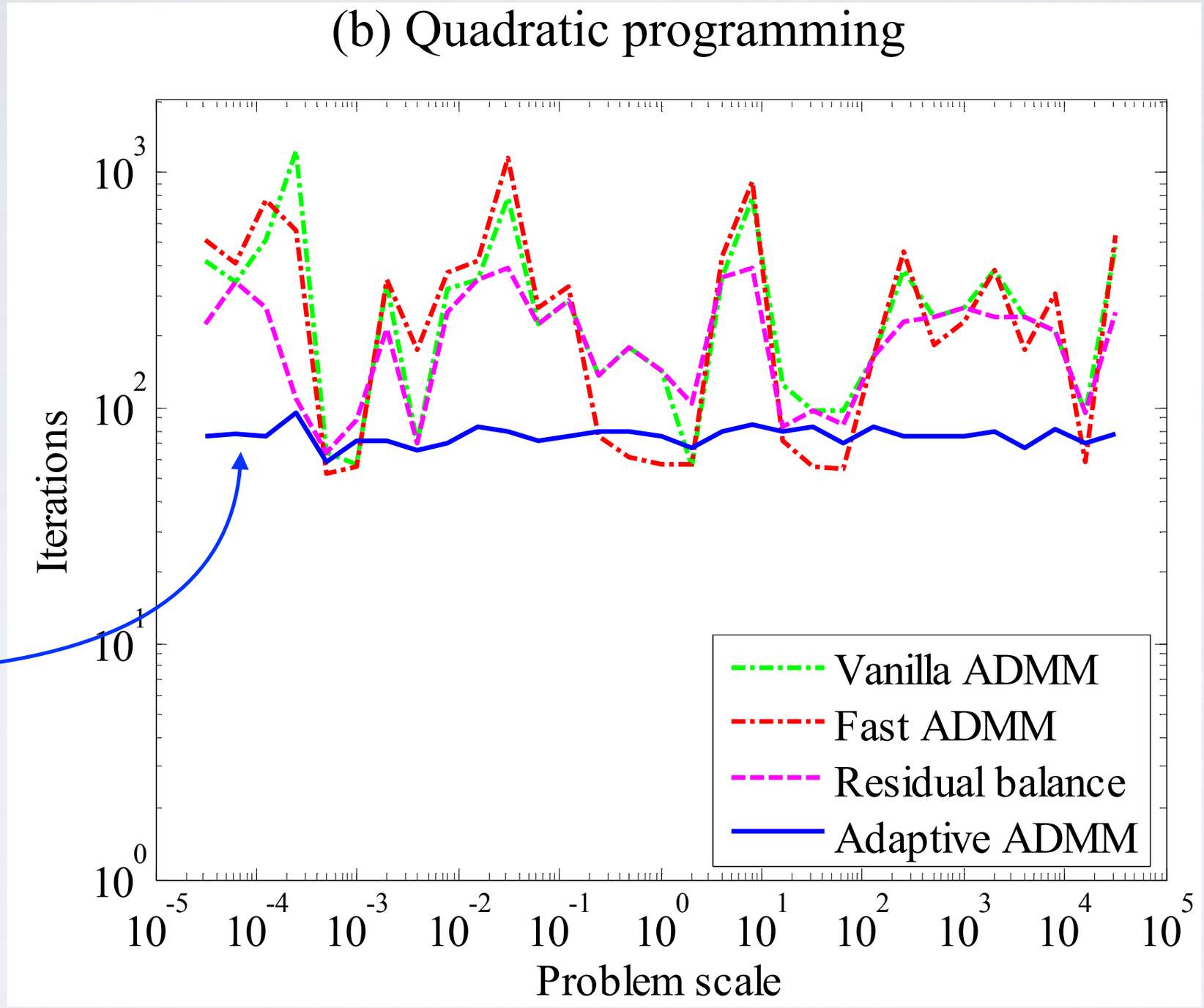


Residual Balancing

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

WRAP UP

ADMM

Complex problems : simple sub-steps
great for model fitting problems

Bells and Whistles

Distributed Variants: transpose reduction
Fast ADMM for poorly-conditioned problems
Adaptive variants for automation

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)

Questions??