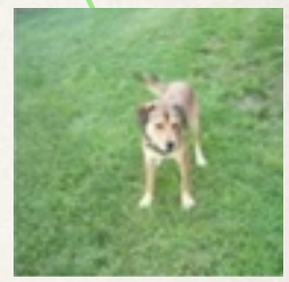
Communication Efficient

Distributed Training of Machine Learning Models

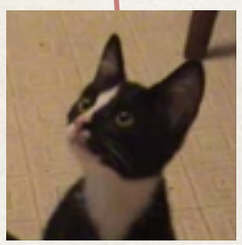
Martin Jaggi
ETH Zurich

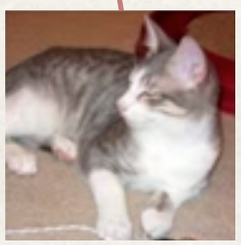
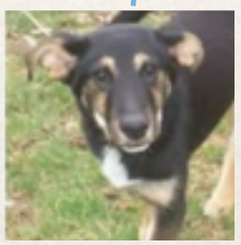
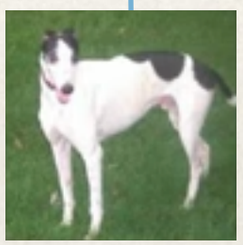
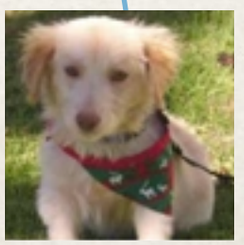
ISMP 2015 Conference, Pittsburgh, 16th July 2015

Classification

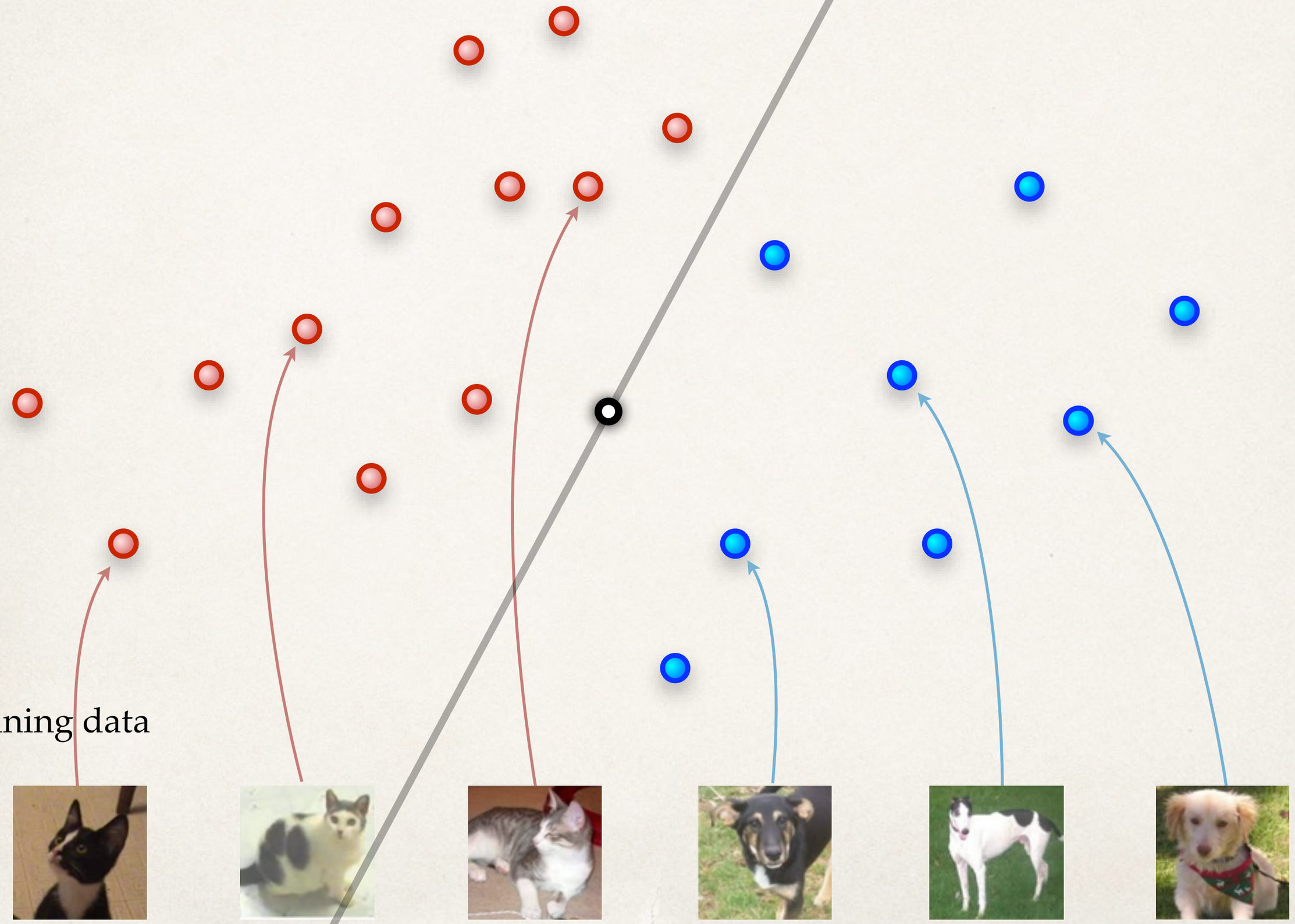


Training data

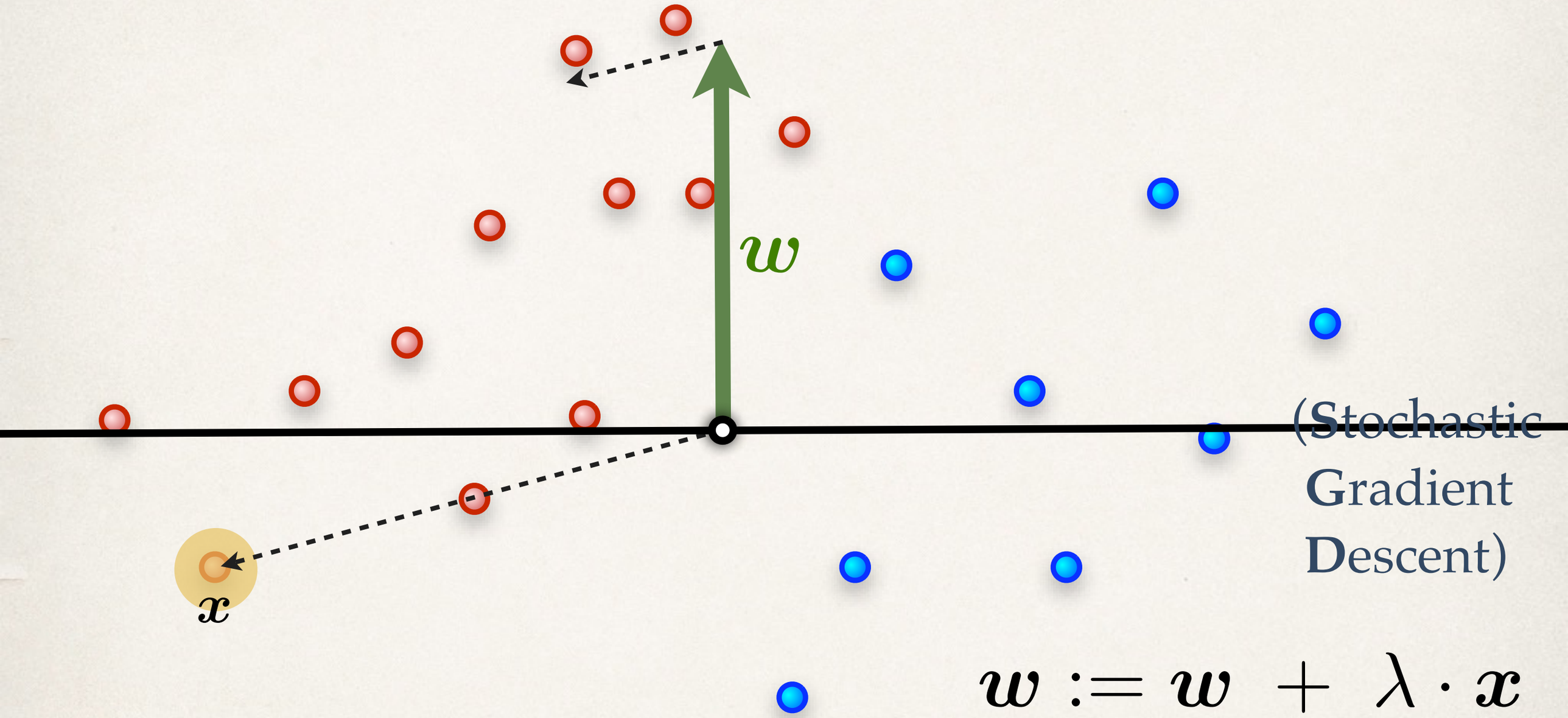


Optimization Algorithms



Optimization Algorithms



(Stochastic Gradient Descent)

$$w := w + \lambda \cdot x$$

Perceptron

(Rosenblatt 1957)

Support-Vektor-Maschine

(Cortes & Vapnik 1995)

Big Data Applications

Classification

Support Vector Machine *(SVM) (L1,L2)*
Logistic Regression *(L1,L2)*
Structured Prediction *(L1,L2)*

Regression

Ridge Regression
Least Squares variants *(L1,L2):*
Lasso, Elastic-Net *(Feature Selection, Compressed Sensing)*

$$\min_{w \in \mathbb{R}^d} \quad \left[P(\boldsymbol{w}) := \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \ell_i(\boldsymbol{w}^T \boldsymbol{x}_i) \right]$$

Big Data Applications

Classification

- Support Vector Machine *(L1,L2)*
- Logistic Regression *(L1,L2)*
- Structured Prediction *(L1,L2)*

Regression

- Ridge Regression
- Least Squares variants *(L1,L2)*
- Lasso, Elastic-Net

$$\min_{w \in \mathbb{R}^d} \left[P(\boldsymbol{w}) := \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \ell_i(\boldsymbol{w}^T \boldsymbol{x}_i) \right]$$

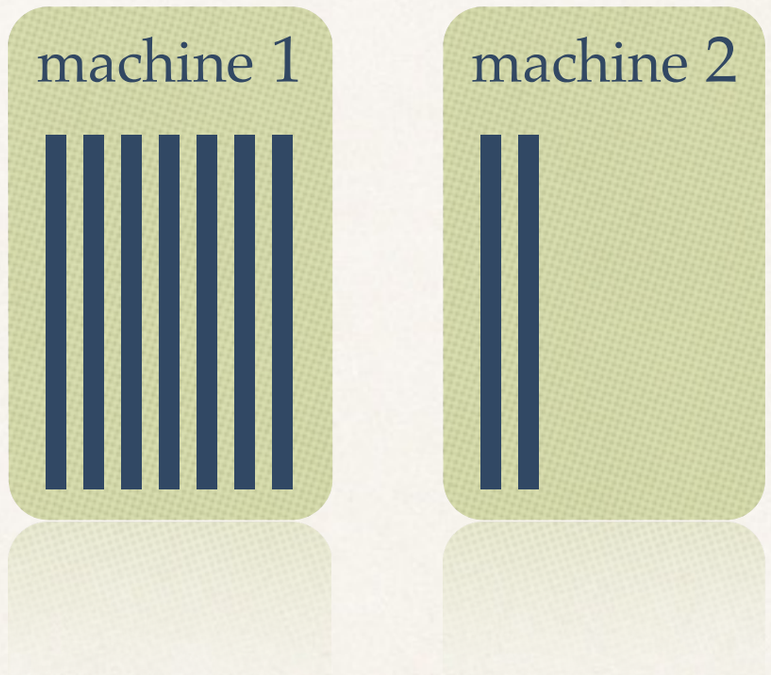
Convergence Rate: *(single machine case)*

$$P(\boldsymbol{w}^{(T)}) - P(\boldsymbol{w}^*) \leq C^{-T}[P(\boldsymbol{w}^{(0)}) - P(\boldsymbol{w}^*)]$$

ℓ_i *smooth* — *SAG, SDCA, SVRG, S2GD, SAGA*

Distributed Machine Learning

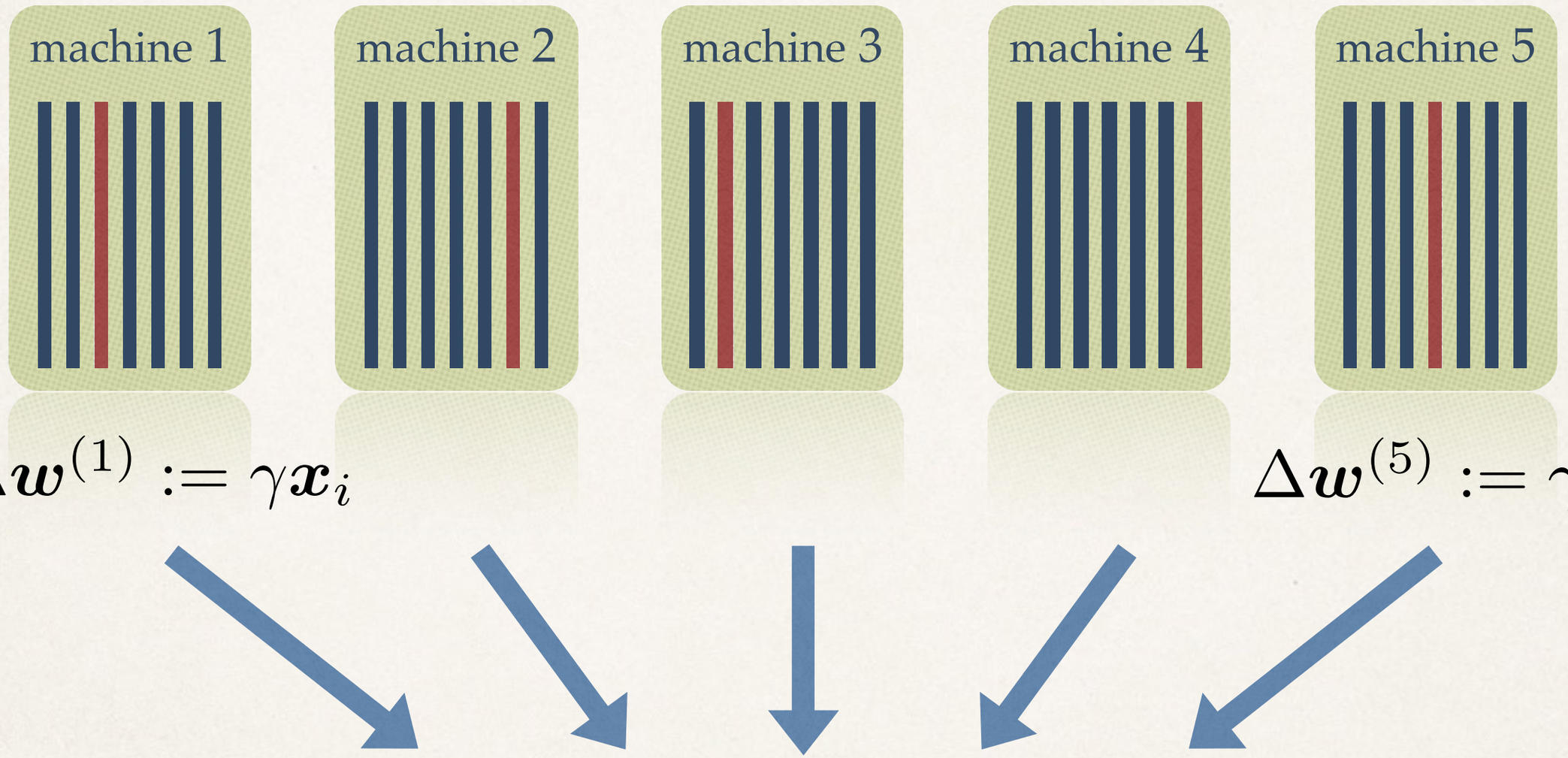
What if the data does not fit onto one computer anymore?



Does More Data Help?

Distributed Optimization

$$\boldsymbol{x}_i \in \mathbb{R}^d$$



Naive Distributed SGD

Problem 1

The Cost of Communication

$$\boldsymbol{v} \in \mathbb{R}^{100}$$

- Reading v from Memory (RAM)

  100 ns

- Sending v to another Machine

  500'000 ns

- One Typical Map-Reduce Iteration *(Hadoop)*

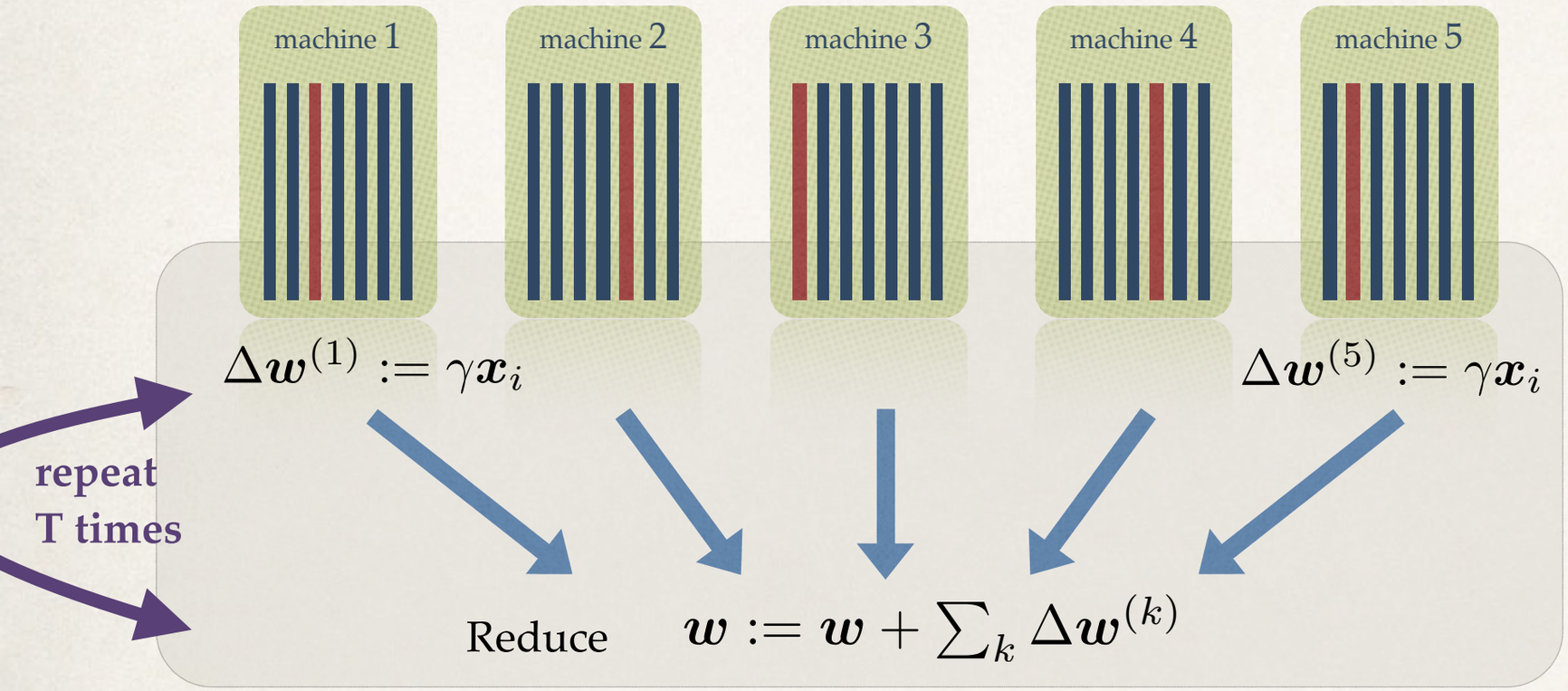
  10'000'000'000 ns

Problem 2

- Parallel Algorithms are Hard
- Single Machine Solvers are Fast

- no **reusability** of good single machine algorithms

Distributed Optimization



Naive Distributed SGD

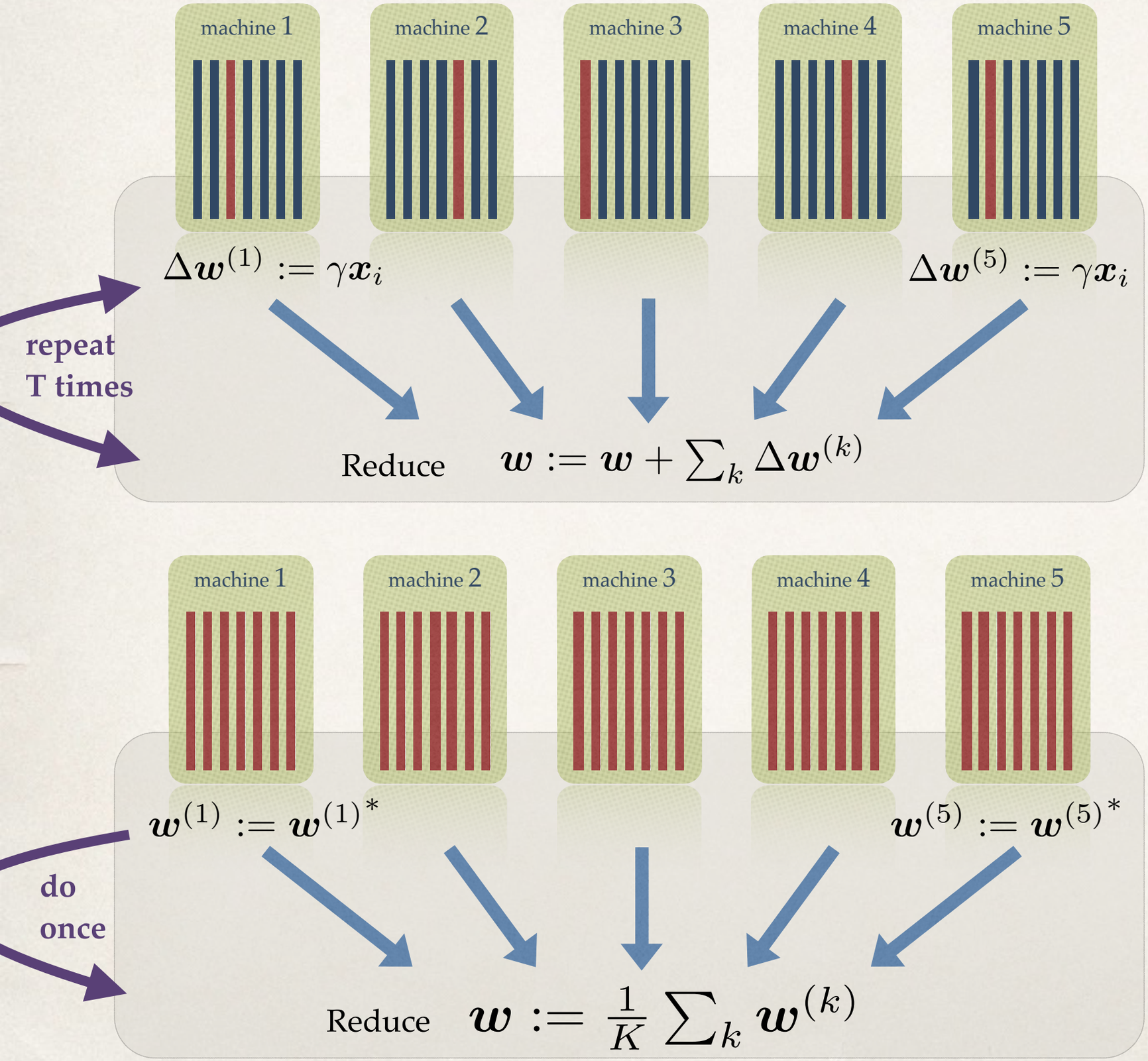
 # local datapoints read: T

 # communications: T

convergence: ✓

"always communicate"

Communication: Always / Never



Naive Distributed SGD

 # local datapoints read: T
 # communications: T
convergence: ✓

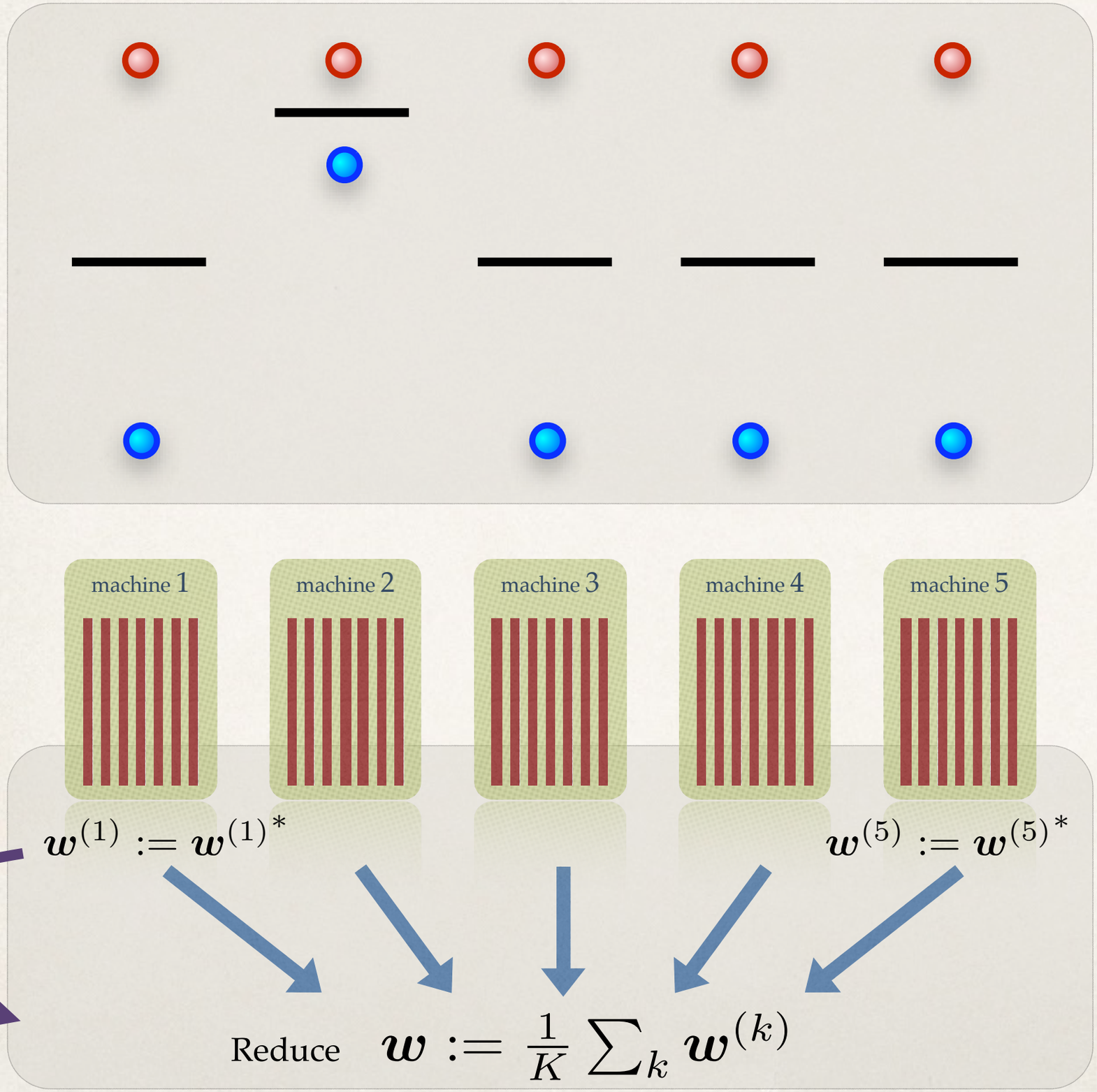
"always communicate"

One-Shot Averaged Distributed Optimization

 # local datapoints read: T
 # communications: 1
convergence: ✗

"never communicate"

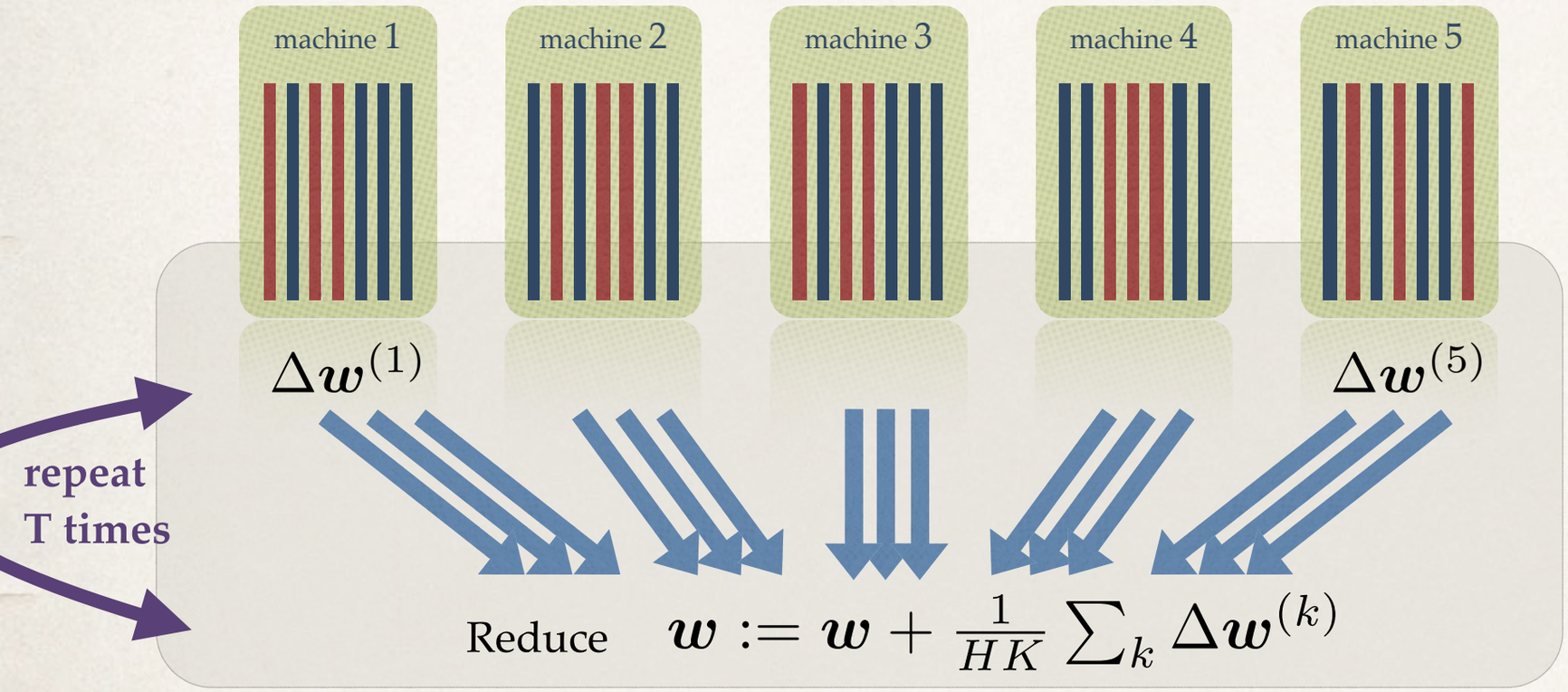
One-Shot Averaging Does Not Work



One-Shot Averaged Distributed Optimization

 # local datapoints read: T
 # communications: 1
convergence: ✗

The Middle Ground



Mini-Batch SGD / CD

 # local datapoints read: TH

 # communications: T

convergence: ✓

Primal-Dual Structure

Primal

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{n}\sum_{i=1}^{n} \ell_i(\mathbf{w}^T\mathbf{x}_i)$$

correspondence

$$\boldsymbol{w} = \frac{1}{\lambda n} A\boldsymbol{\alpha}$$

Dual

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{\lambda}{2}\left\|\frac{A\boldsymbol{\alpha}}{\lambda n}\right\|^2 - \frac{1}{n}\sum_{i=1}^{n} \ell_i^*(-\alpha_i)$$

Optimization Algorithms:

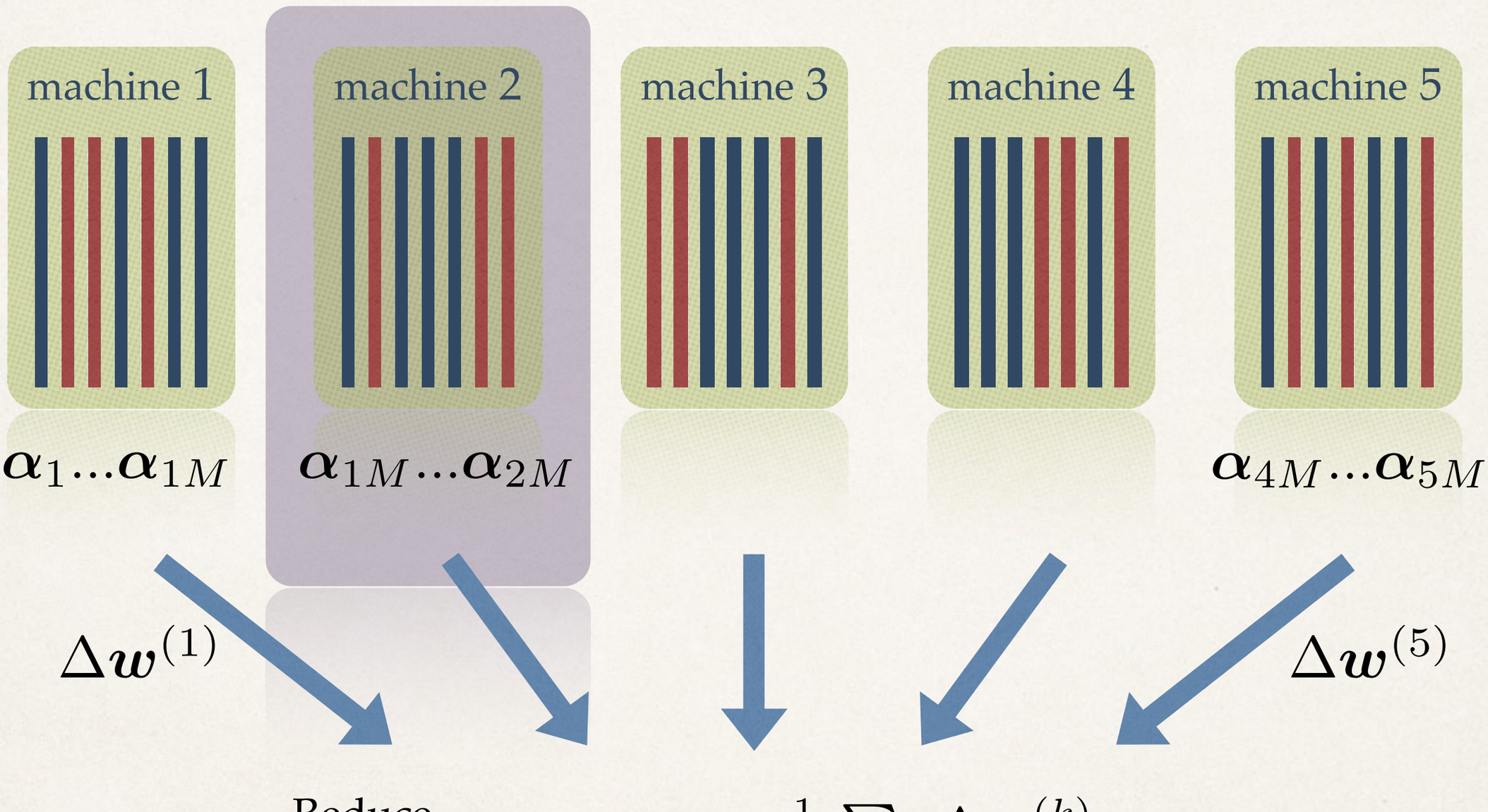
SGD

Coordinate Descent

SVM dual:	*LibLinear '08*
*struct*SVM dual:	*BCFW '13*
Lasso primal:	*GLMnet '10*
Theory:	*SDCA '13*

$\boldsymbol{w}_{(\boldsymbol{\alpha})} := A\boldsymbol{\alpha}$

Communication Efficient Distributed *Dual* Coordinate Ascent



CoCoA

 # local datapoints read: TH
 # communications: T
convergence: ✓

Primal-Dual Structure

Primal

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{n}\sum_{i=1}^{n} \ell_i(\mathbf{w}^T\mathbf{x}_i)$$

correspondence

$$\boldsymbol{w} = \frac{1}{\lambda n} A\boldsymbol{\alpha}$$

Dual

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\frac{\lambda}{2}\left\|\frac{A\boldsymbol{\alpha}}{\lambda n}\right\|^2 - \frac{1}{n}\sum_{i=1}^{n} \ell_i^*(-\alpha_i)$$

$$A_{\text{loc}}\Delta\boldsymbol{\alpha}_{[k]} + \boldsymbol{w}$$

Optimization Algorithms:

SGD

Coordinate Descent

SVM dual:	*LibLinear '08*
*struct*SVM dual:	*BCFW '13*
Lasso primal:	*GLMnet '10*
Theory:	*SDCA '13*

Local Subproblems

CoCoA

CoCoA+

$$\max_{\Delta\boldsymbol{\alpha}_{[k]}\in\mathbb{R}^n} -\frac{1}{n}\sum_{i\in\mathcal{P}_k}\ell_i^*(-\alpha_i-(\Delta\boldsymbol{\alpha}_{[k]})_i)$$

$$-\frac{1}{n}\mathbf{w}^T A\Delta\boldsymbol{\alpha}_{[k]}-\frac{\lambda}{2}\left\|\frac{1}{\lambda n}A\Delta\boldsymbol{\alpha}_{[k]}\right\|^2$$

Local Θ-Approximation

For $\Theta \in [0, 1)$, we assume the local solver finds a (possibly) randomized approximate solution satisfying:

$$\mathbb{E}\left[\mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}^*) - \mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]})\right]$$
$$\leq \Theta\left(\mathcal{G}_k^{\sigma'}(\Delta\boldsymbol{\alpha}_{[k]}^*) - \mathcal{G}_k^{\sigma'}(\mathbf{0})\right)$$

Convergence - Smooth Loss

Theorem. Let $\ell_i(.)$ be $1/\mu$ **-smooth**

Obtain suboptimality ϵ *, after T rounds, with:*

- CoCoA, averaging $\gamma = 1/K$

$$T \geq \frac{1}{1-\Theta} \frac{\lambda\mu K+1}{\lambda\mu} \log \frac{1}{\epsilon}$$

- CoCoA+, adding $\gamma = 1$

$$T \geq \frac{1}{1-\Theta} \frac{\lambda\mu+1}{\lambda\mu} \log \frac{1}{\epsilon}$$

Convergence - General Convex Loss

Theorem. Let $\ell_i(.)$ be L-**Lipschitz**
Obtain suboptimality ϵ *, after T rounds, with:*

- CoCoA, averaging $\gamma = 1/K$

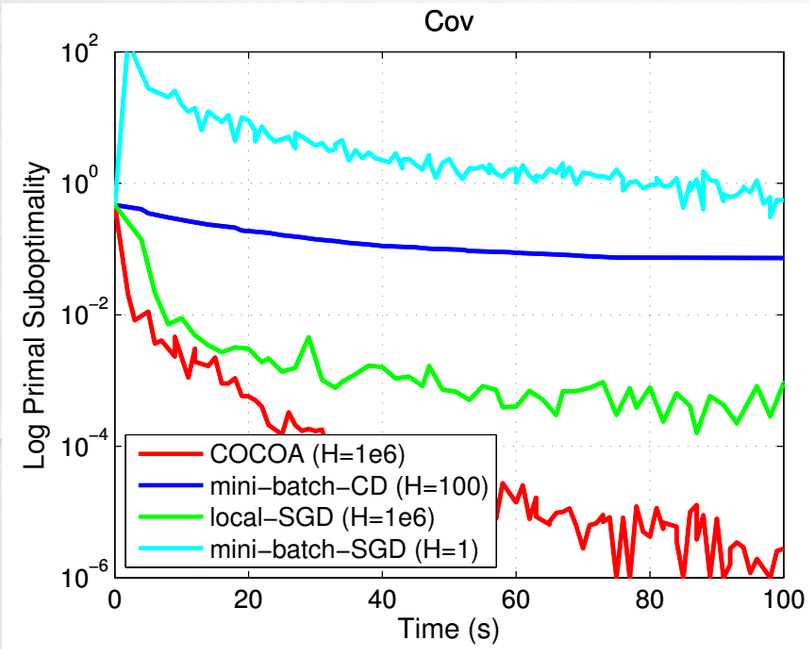
$$T \geq \tilde{\mathcal{O}}\left(\frac{K}{1-\Theta}\left(\frac{8L^2}{\lambda K \epsilon} + \tilde{c}\right)\right)$$

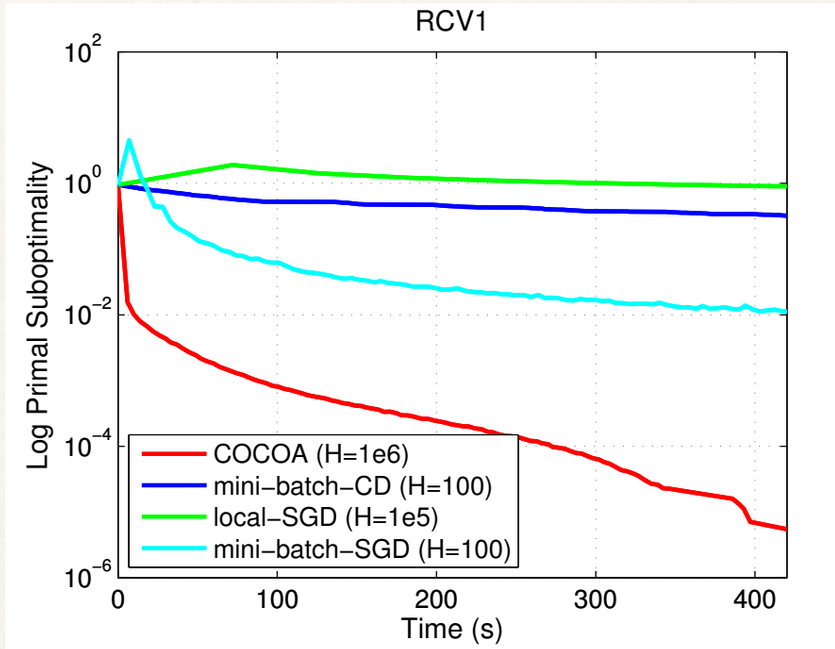
- CoCoA+, adding $\gamma = 1$

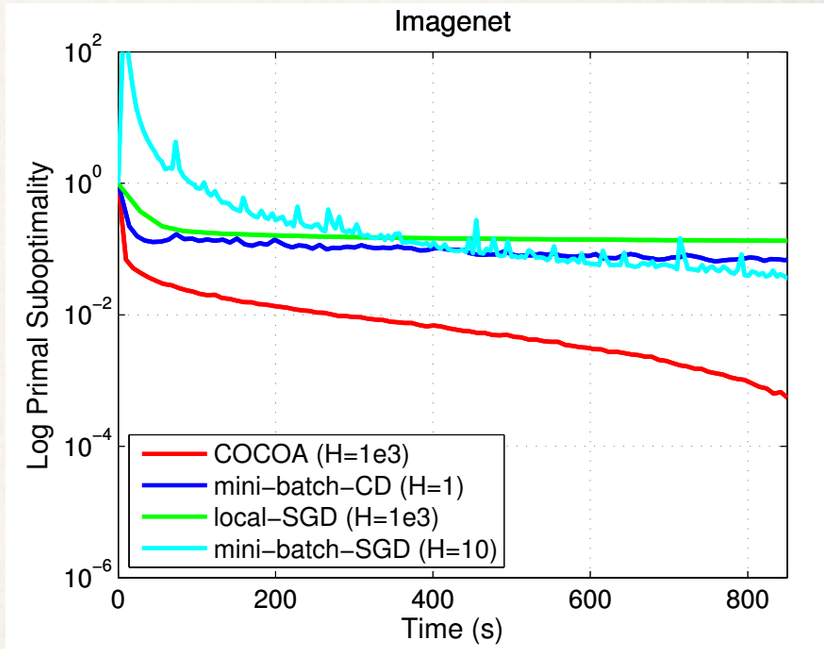
$$T \geq \tilde{\mathcal{O}}\left(\frac{1}{1-\Theta}\left(\frac{8L^2}{\lambda \epsilon} + \tilde{c}\right)\right)$$

Experiments

Dataset	**Training** n	**Features** d	**Sparsity**	λ	**Workers** K
cov	522,911	54	22.22%	1e-6	4
rcv1	677,399	47,236	0.16%	1e-6	8
imagenet	32,751	160,000	100%	1e-5	32







Experiments

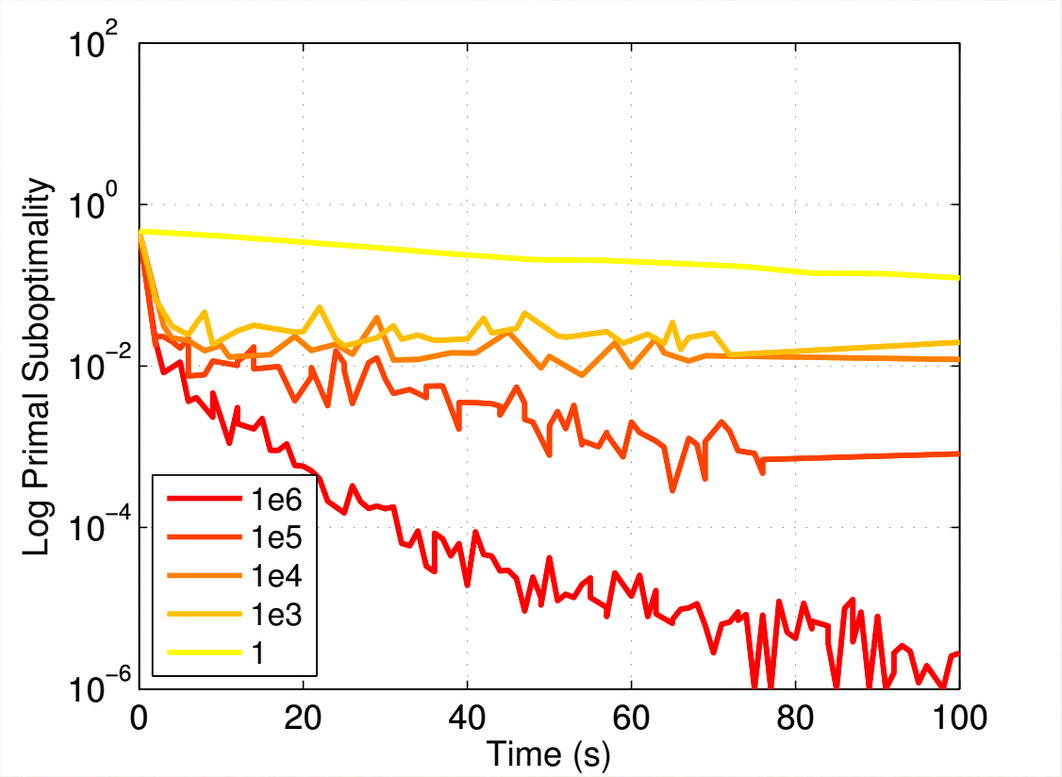


Figure 3: Effect of H on CoCoA

$H = \#\text{inner iterations}$

Dissolve struct

A Library for Distributed Structured Prediction

built on Spark

Structured SVM solver

Block Coordinate Frank-Wolfe

Distributed Optimization

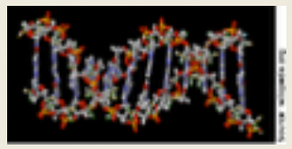
CoCoA

Open Source
Approximate Inference *allowed!*
drop-in replacement for SVM^{struct}

Applications:

Text
- Parsing
- POS tagging, chunking
- sentence alignment
- named entity recognition

Biology
Protein structure & function prediction



Vision
Horse Segmentation, OCR

more?
- Scene understanding
- object localization & recog.

Your Application?

Conclusion

- full adaptivity to the communication cost
- re-usability of good single machine solvers
- theoretical and practical efficiency

Open Research

- generalizations to *non-smooth losses*, Lasso
- purely *primal* algorithm?
- multi-level approach on heterogenous systems

Thanks

"Communication-Efficient Distributed Dual Coordinate Ascent"

CoCoA+ paper (ICML 2015)

CoCoA paper (NIPS 2014)

Spark code is available on *github*

joint work with Virginia Smith, Martin Takáč, Chenxin Ma, Simone Forte, Tribhuvanesh Orekondy, Jonathan Terhorst, Sanjay Krishnan, Aurelien Lucchi, Peter Richtarik, Thomas Hofmann, Michael I. Jordan