Randomized Algorithms for Large-scale Convex Optimization

Lijun Zhang

LAMDA group, Nanjing University, China

2016 Nanjing Workshop on Numerical Optimization with Applications

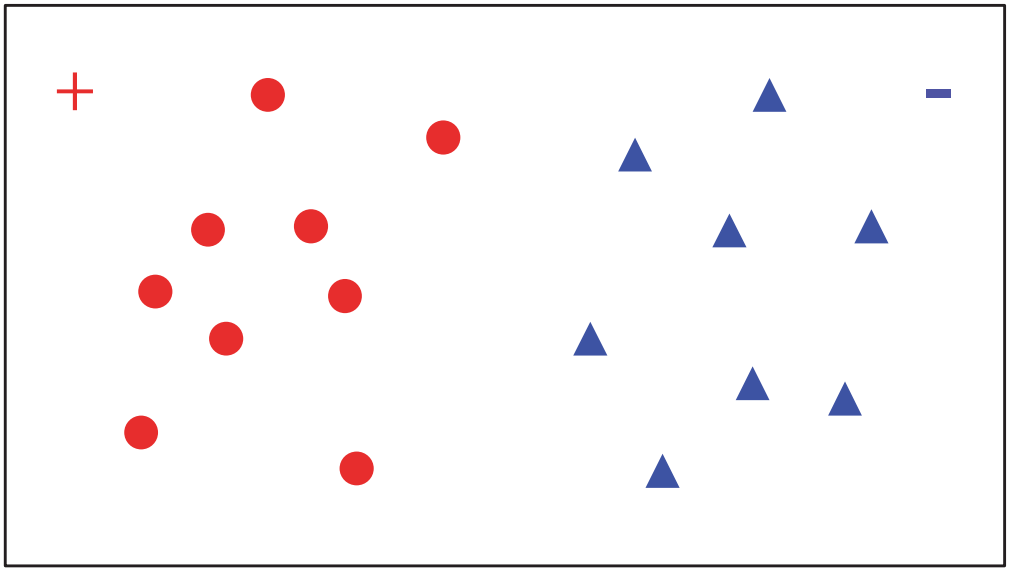
Outline

LAMDA
Learning And Mining from DatA

Learning by Optimization

Support Vector Machine (SVM)

- Modeling—Large Margin



- Optimization—Empirical Risk Minimization

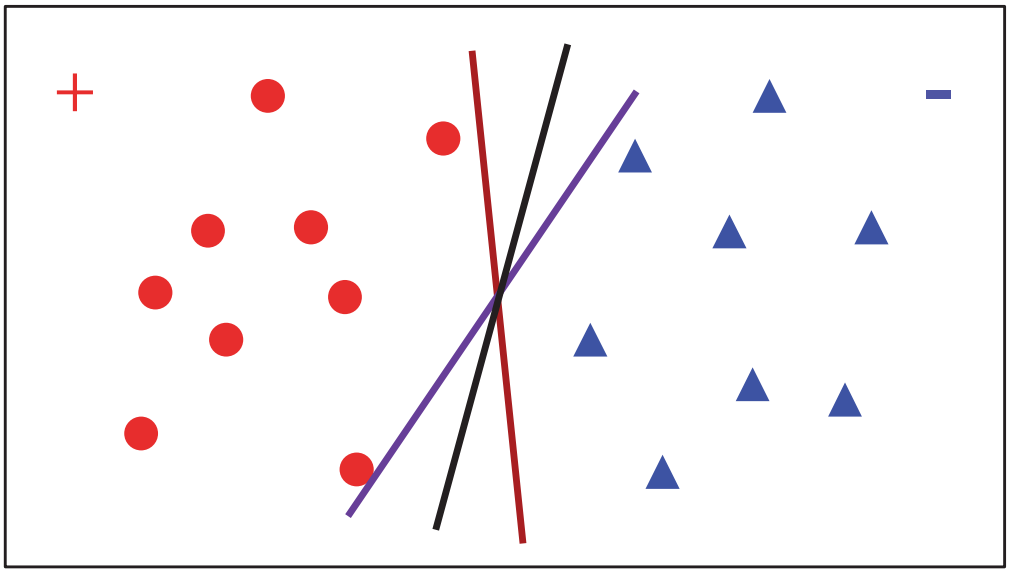
$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{x}_i^\top \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

LAMDA
Learning And Mining from DatA

Learning by Optimization

Support Vector Machine (SVM)

- Modeling—Large Margin



- Optimization—Empirical Risk Minimization

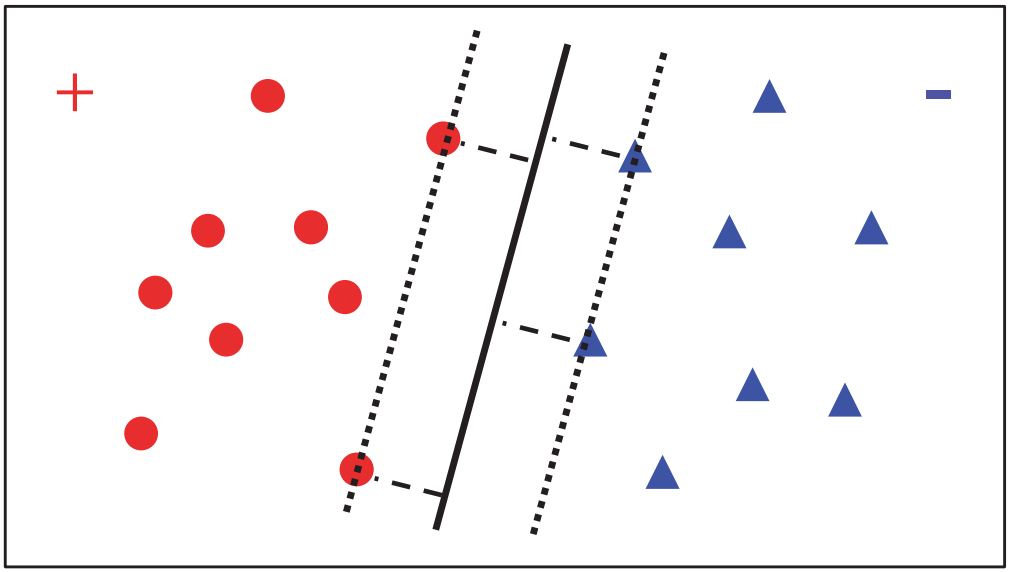
$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{x}_i^\top \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

LAMDA
Learning And Mining from DatA

Learning by Optimization

Support Vector Machine (SVM)

- Modeling—Large Margin



- Optimization—Empirical Risk Minimization

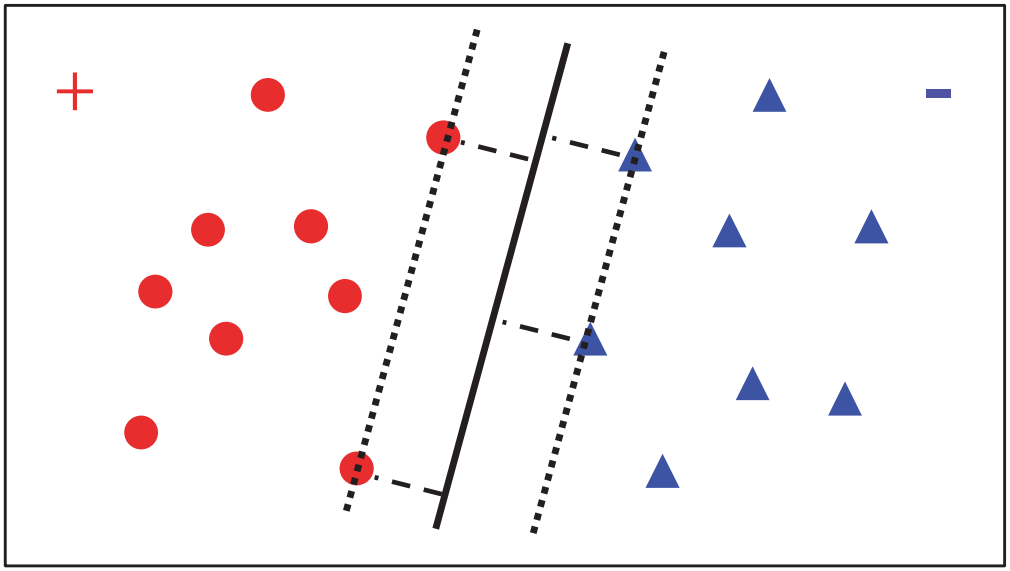
$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{x}_i^\top \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

LAMDA
Learning And Mining from DatA

Learning by Optimization

Support Vector Machine (SVM)

- Modeling—Large Margin



- Optimization—Empirical Risk Minimization

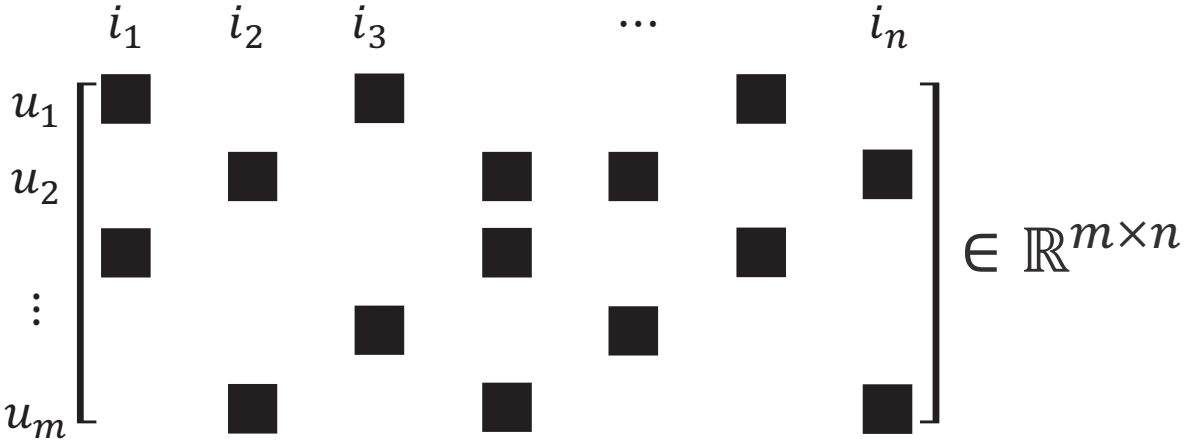
$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i \mathbf{x}_i^\top \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

LAMDA
Learning And Mining from DatA

Learning by Optimization

Collaborative Filtering (CF)

- Modeling—Low-rank Matrix Completion



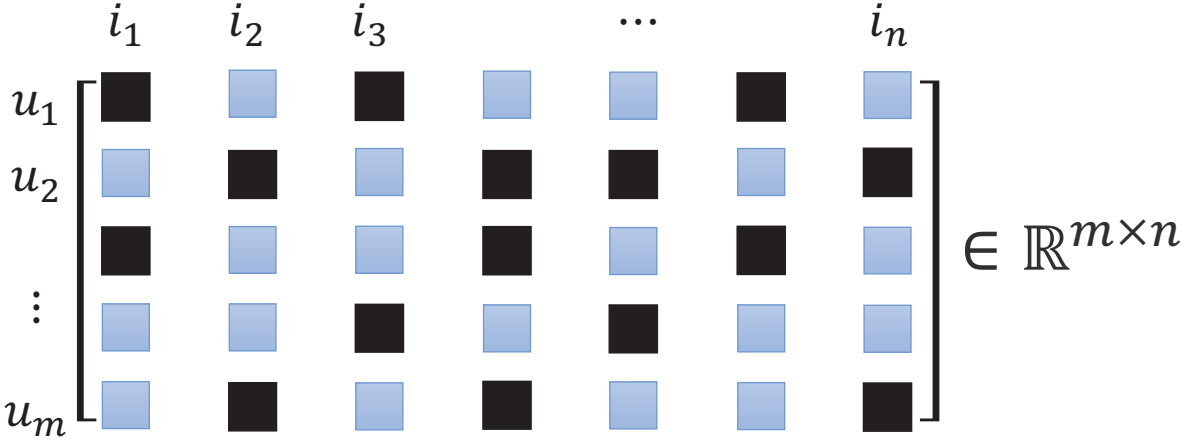
- Optimization—Constrained Nuclear Norm Minimization

$$\min_{W\in\mathbb{R}^{m\times n}} \quad \|W\|_*$$
$$\text{s.t.} \quad W_{ij} = M_{ij},\ \forall (i,j)\in\Omega$$

Learning by Optimization

Collaborative Filtering (CF)

- Modeling—Low-rank Matrix Completion



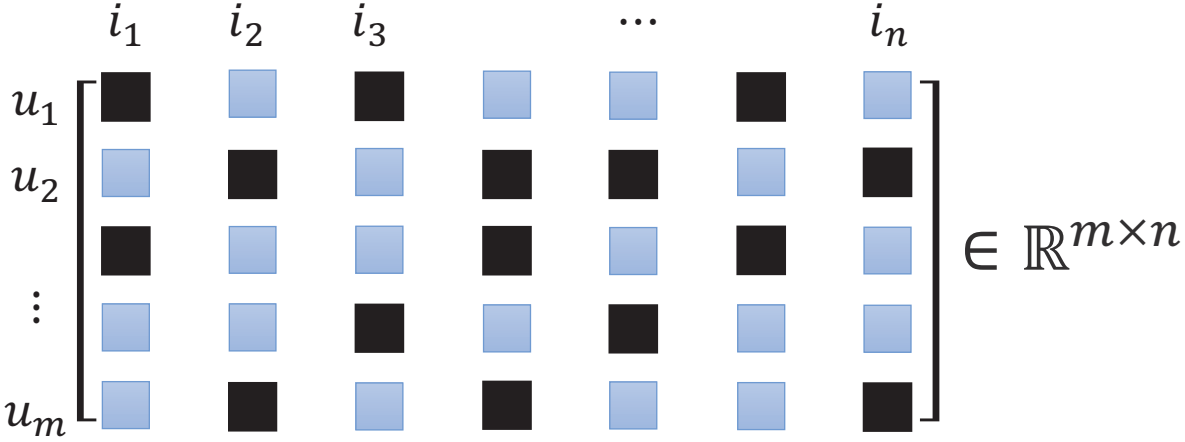
- Optimization—Constrained Nuclear Norm Minimization

$$\min_{W\in\mathbb{R}^{m\times n}} \quad \|W\|_*$$
$$\text{s.t.} \quad W_{ij} = M_{ij},\ \forall (i,j)\in\Omega$$

Learning by Optimization

Collaborative Filtering (CF)

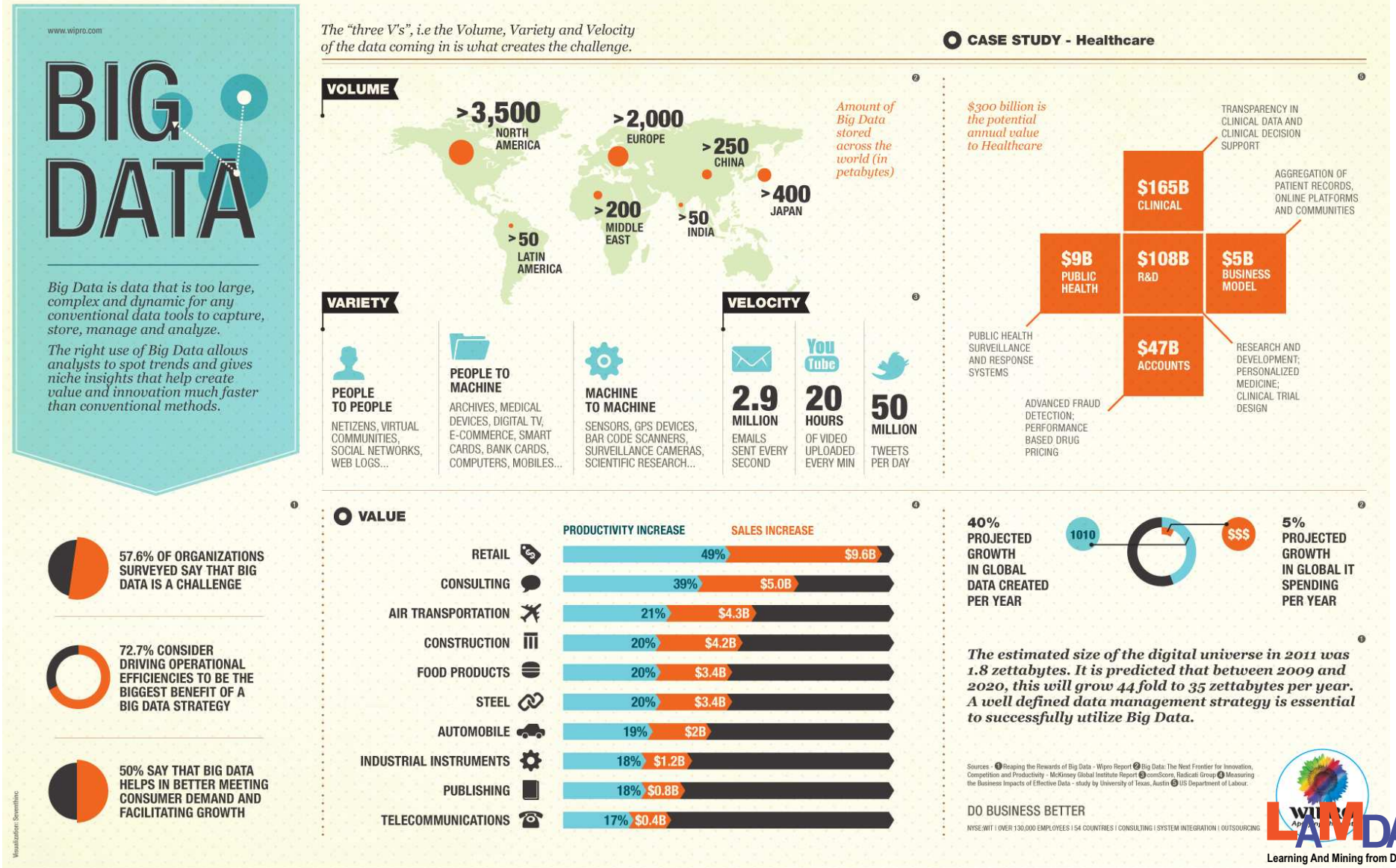
- Modeling—Low-rank Matrix Completion



- Optimization—Constrained Nuclear Norm Minimization

$$\min_{W\in\mathbb{R}^{m\times n}} \quad \|W\|_*$$
$$\text{s.t.} \quad W_{ij} = M_{ij},\ \forall (i,j)\in\Omega$$

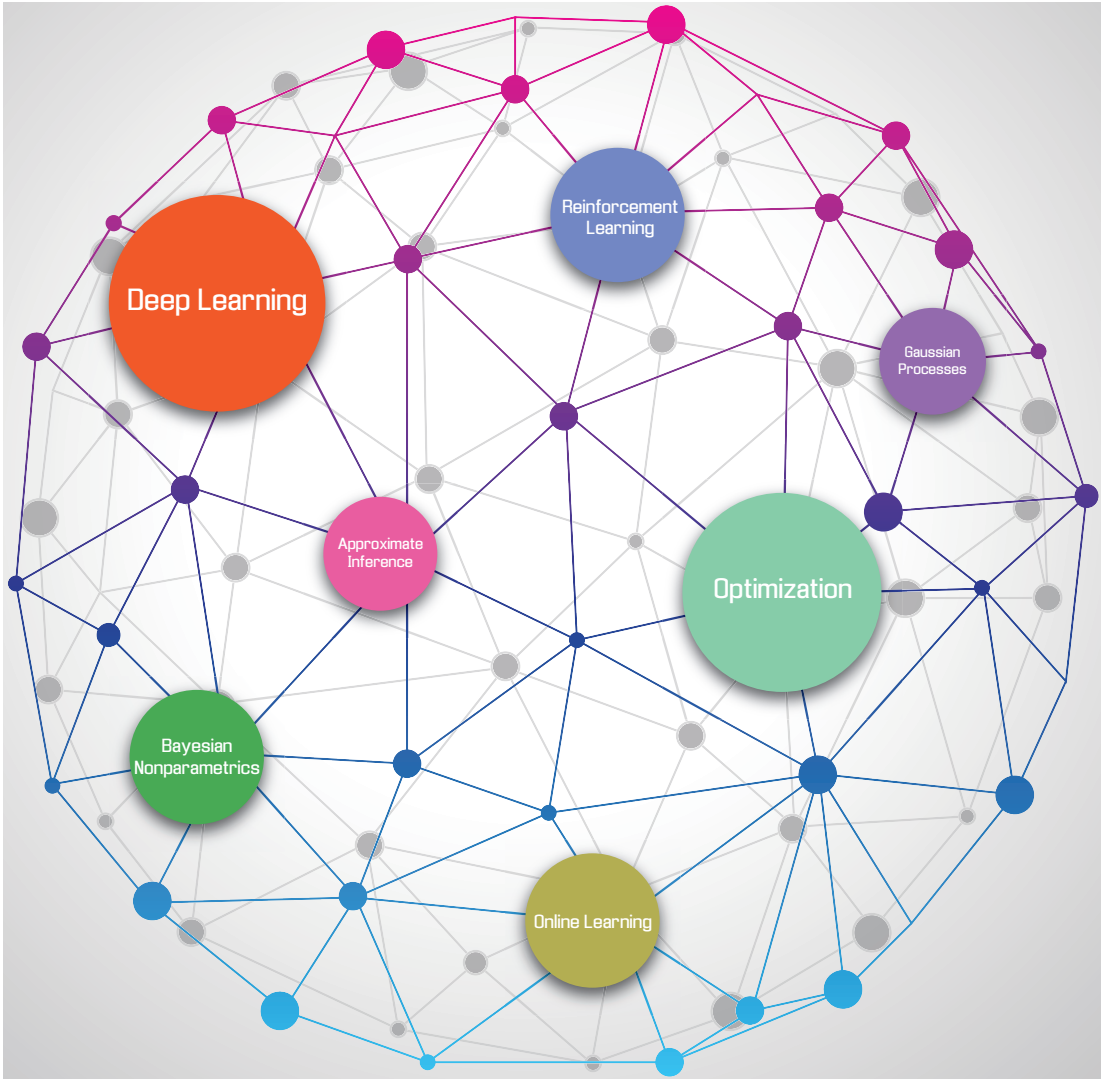
The Big Data Challenge



http://thumbnails-visually.netdna-ssl.com/big-data_50291c3b16257.jpg

Optimization in Machine Learning

The 32nd International Conference on Machine Learning (ICML 2015)

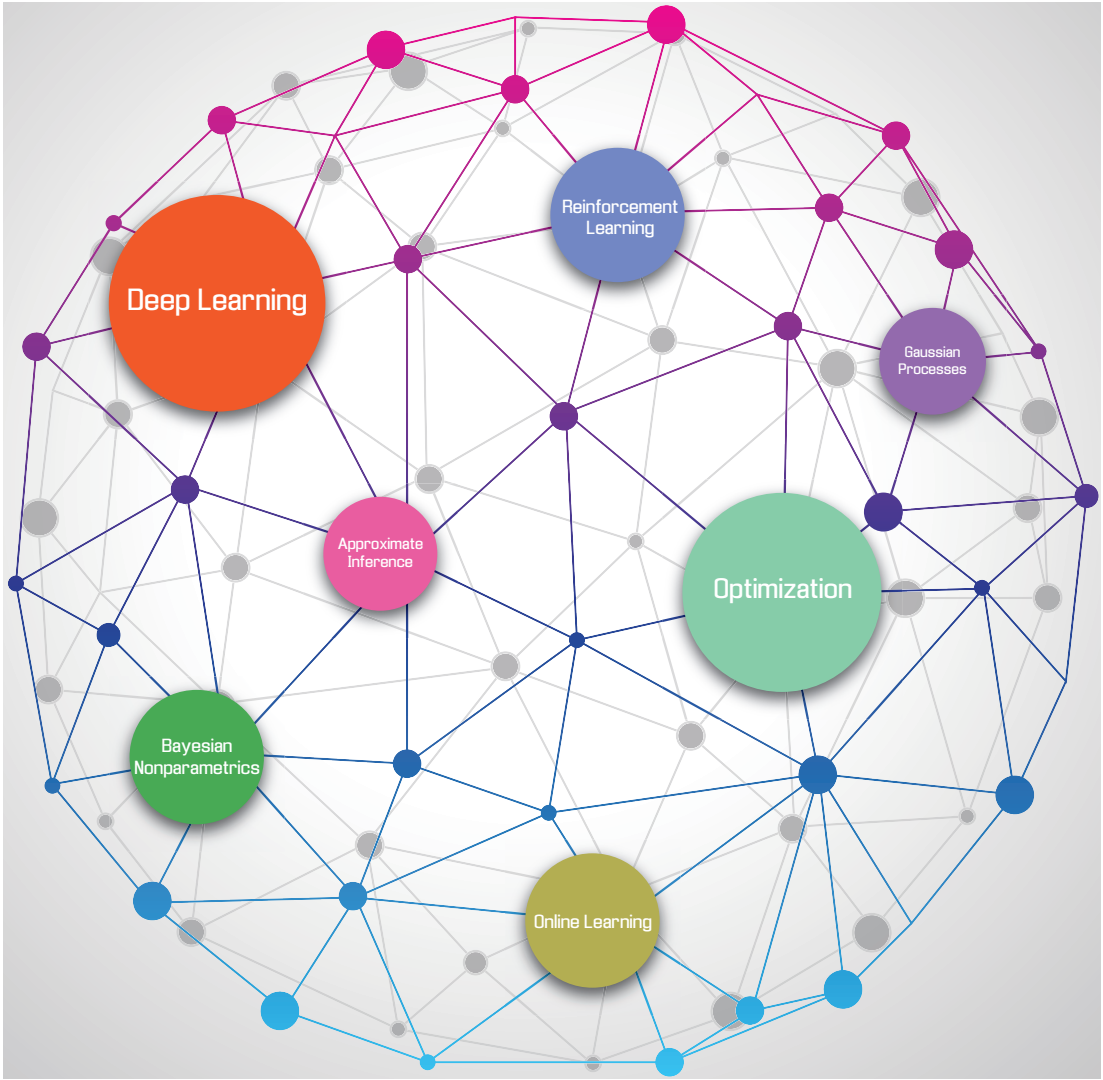


- Linear Optimization
- Convex Optimization
- Non-convex Optimization
- Combinatorial Optimization
- Stochastic Optimization
- Approximate Optimization
- Distributed Optimization
- Online Optimization

LAMDA
Learning And Mining from DatA

Optimization in Machine Learning

The 32nd International Conference on Machine Learning (ICML 2015)



- Linear Optimization
- Convex Optimization
- Non-convex Optimization
- Combinatorial Optimization
- Stochastic Optimization
- Approximate Optimization
- Distributed Optimization
- Online Optimization

LAMDA
Learning And Mining from DatA

Outline

Background

Supervised Learning

Input

- Training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$

Output

- A hypothesis $\mathbf{w}_* \in \mathcal{W} \subseteq \mathbb{R}^d$ such that

$$\mathbf{x}^\top \mathbf{w}_* \approx y \text{ or } \operatorname{sgn}(\mathbf{x}^\top \mathbf{w}_*) \approx y$$

Empirical Risk Minimization

$$\min_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

- $\ell(\cdot, \cdot)$ is a loss, e.g., hinge loss $\ell(u, v) = \max(0, 1 - uv)$
- $\Omega(\cdot)$ is a regularizer, e.g., $\lambda \|\mathbf{w}\|_2^2$ or $\lambda \|\mathbf{w}\|_1$

LAMDA Learning And Mining from DatA

Background

Supervised Learning

Input

- Training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$

Output

- A hypothesis $\mathbf{w}_* \in \mathcal{W} \subseteq \mathbb{R}^d$ such that

$$\mathbf{x}^\top \mathbf{w}_* \approx y \text{ or } \operatorname{sgn}(\mathbf{x}^\top \mathbf{w}_*) \approx y$$

Empirical Risk Minimization

$$\min_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

- $\ell(\cdot, \cdot)$ is a loss, e.g., hinge loss $\ell(u, v) = \max(0, 1 - uv)$
- $\Omega(\cdot)$ is a regularizer, e.g., $\lambda \|\mathbf{w}\|_2^2$ or $\lambda \|\mathbf{w}\|_1$

LAMDA Learning And Mining from DatA

Gradient Descent (GD)

Convex Optimization Problem

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i{}^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t)\right)$
3: $\quad \mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
4: **end for**

Convergence Rates [Nesterov, 2004]

	Convex & Smooth	Strongly Convex & Smooth
Rate	$O\left(\frac{1}{T}\right) \to O\left(\frac{1}{T^2}\right)$	$O\left(\frac{1}{\alpha^T}\right)$

LAMDA Learning And Mining from DatA

Gradient Descent (GD)

Convex Optimization Problem

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i{}^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t)\right)$
3: $\quad \mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
4: **end for**

Convergence Rates [Nesterov, 2004]

	Convex & Smooth	Strongly Convex & Smooth
Rate	$O\left(\frac{1}{T}\right) \to O\left(\frac{1}{T^2}\right)$	$O\left(\frac{1}{\alpha^T}\right)$

LAMDA Learning And Mining from DatA

Gradient Descent (GD)

Convex Optimization Problem

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i{}^\top \mathbf{w}_t) + \nabla\Omega(\mathbf{w}_t)\right)$
3: $\quad \mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
4: **end for**

Convergence Rates [Nesterov, 2004]

	Convex & Smooth	Strongly Convex & Smooth
Rate	$O\left(\frac{1}{T}\right) \to O\left(\frac{1}{T^2}\right)$	$O\left(\frac{1}{\alpha^T}\right)$

LAMDA Learning And Mining from DatA

Gradient Descent (GD)

Convex Optimization Problem

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i{}^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t)\right)$
3: $\quad \mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
4: **end for**

Computational Cost

- Time Complexity: $O(nd) + O(poly(d))$
- Space Complexity: $O(nd)$

LAMDA Learning And Mining from DatA

Gradient Descent (GD)

Convex Optimization Problem

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i{}^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t)\right)$
3: $\quad \mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
4: **end for**

Computational Cost

- Time Complexity: $O(nd) + O(poly(d))$
- Space Complexity: $O(nd)$

LAMDA Learning And Mining from DatA

Gradient Descent (GD)

Convex Optimization Problem

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i{}^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t)\right)$
3: $\quad \mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
4: **end for**

Computational Cost

- Time Complexity: $O(nd) + O(poly(d))$
- Space Complexity: $O(nd)$

LAMDA Learning And Mining from DatA

Gradient Descent (GD)

Convex Optimization Problem

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad \mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i{}^\top \mathbf{w}_t) + \nabla\Omega(\mathbf{w}_t)\right)$
3: $\quad \mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
4: **end for**

The Challenge

- Computationally expensive if both n and d are very large

LAMDA Learning And Mining from DatA

Randomized Algorithms

Random Sampling for Large n

- Referred to as *Stochastic Optimization*
- Stochastic Gradient Descent (SGD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Sample a training instance $(\mathbf{x}_i, y_i)$ randomly
3: $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t) \right)$
4: $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
5: **end for**

- Slow convergence [Zhang et al., 2013a, Mahdavi et al., 2013]

Random Projection for Large d

- Reduce the dimensionality by random projection
- Referred to as *Stochastic Approximation*
- Approximation error [Zhang et al., 2013b, Zhang et al., 2014]

LAMDA Learning And Mining from DatA

Randomized Algorithms

Random Sampling for Large n

- Referred to as *Stochastic Optimization*
- Stochastic Gradient Descent (SGD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Sample a training instance $(\mathbf{x}_i, y_i)$ randomly
3: $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t) \right)$
4: $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
5: **end for**

- Slow convergence [Zhang et al., 2013a, Mahdavi et al., 2013]

Random Projection for Large d

- Reduce the dimensionality by random projection
- Referred to as *Stochastic Approximation*
- Approximation error [Zhang et al., 2013b, Zhang et al., 2014]

LAMDA Learning And Mining from DatA

Randomized Algorithms

Random Sampling for Large n

- Referred to as *Stochastic Optimization*
- Stochastic Gradient Descent (SGD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Sample a training instance $(\mathbf{x}_i, y_i)$ randomly
3: $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t) \right)$
4: $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
5: **end for**

- Slow convergence [Zhang et al., 2013a, Mahdavi et al., 2013]

Random Projection for Large d

- Reduce the dimensionality by random projection
- Referred to as *Stochastic Approximation*
- Approximation error [Zhang et al., 2013b, Zhang et al., 2014]

LAMDA Learning And Mining from DatA

Randomized Algorithms

Random Sampling for Large n

- Referred to as *Stochastic Optimization*
- Stochastic Gradient Descent (SGD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Sample a training instance $(\mathbf{x}_i, y_i)$ randomly
3: $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t)\right)$
4: $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
5: **end for**

- Slow convergence [Zhang et al., 2013a, Mahdavi et al., 2013]

Random Projection for Large d

- Reduce the dimensionality by random projection
- Referred to as *Stochastic Approximation*
- Approximation error [Zhang et al., 2013b, Zhang et al., 2014]

LAMDA Learning And Mining from DatA

Randomized Algorithms

Random Sampling for Large n

- Referred to as *Stochastic Optimization*
- Stochastic Gradient Descent (SGD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Sample a training instance $(\mathbf{x}_i, y_i)$ randomly
3: $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left(\nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) + \nabla \Omega(\mathbf{w}_t)\right)$
4: $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$
5: **end for**

- Slow convergence [Zhang et al., 2013a, Mahdavi et al., 2013]

Random Projection for Large d

- Reduce the dimensionality by random projection
- Referred to as *Stochastic Approximation*
- Approximation error [Zhang et al., 2013b, Zhang et al., 2014]

Optimization after Random Projection

The Optimization Problem in $\mathbb{R}^d$

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\mathbf{x}_i^\top\mathbf{w})+\frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

The Procedure

1. Reduce the dimensionality $\widehat{\mathbf{x}}_i = A^\top\mathbf{x}_i \in \mathbb{R}^m$, where $A \in \mathbb{R}^{d\times m}$ and $A_{ij} \sim \mathcal{N}(0, 1/m)$
2. Solve the primal problem in $\mathbb{R}^m$

$$\min_{\mathbf{z}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\mathbf{z}^\top\widehat{\mathbf{x}}_i)+\frac{\lambda}{2}\|\mathbf{z}\|_2^2$$

   Time and space complexities are reduced from $O(nd)$ to $O(nm)$
3. Compute $\widehat{\mathbf{w}} \in \mathbb{R}^d$ by $\widehat{\mathbf{w}} = A\mathbf{z}_*$

LAMDA Learning And Mining from DatA

Optimization after Random Projection

The Optimization Problem in $\mathbb{R}^d$

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

The Procedure

1. Reduce the dimensionality $\widehat{\mathbf{x}}_i = A^\top \mathbf{x}_i \in \mathbb{R}^m$, where $A \in \mathbb{R}^{d\times m}$ and $A_{ij} \sim \mathcal{N}(0, 1/m)$
2. Solve the primal problem in $\mathbb{R}^m$

$$\min_{\mathbf{z}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, \mathbf{z}^\top \widehat{\mathbf{x}}_i) + \frac{\lambda}{2}\|\mathbf{z}\|_2^2$$

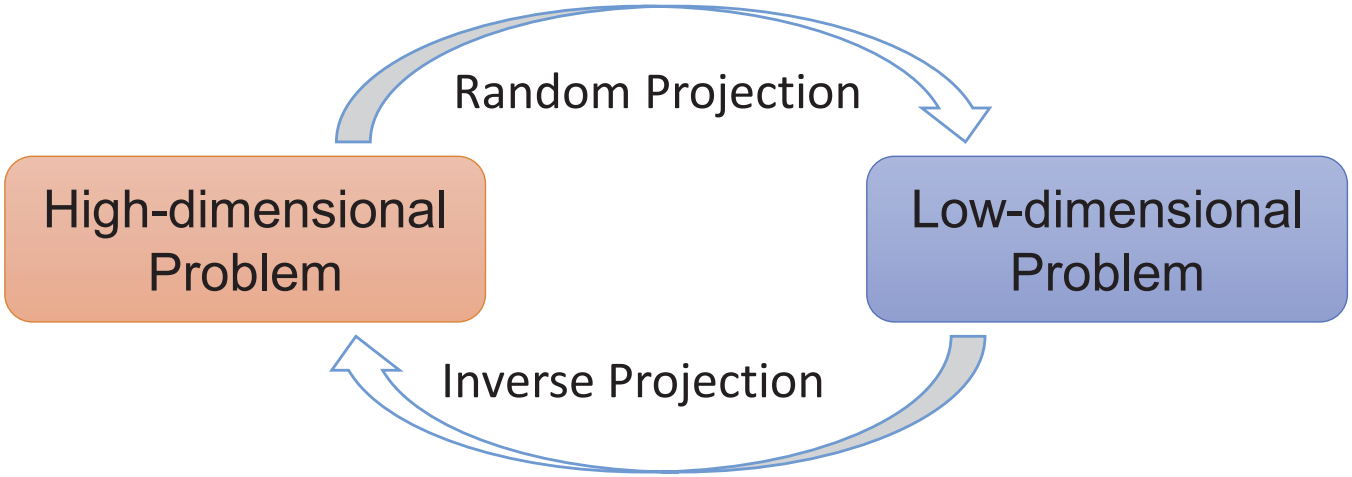
   Time and space complexities are reduced from $O(nd)$ to $O(nm)$
3. Compute $\widehat{\mathbf{w}} \in \mathbb{R}^d$ by $\widehat{\mathbf{w}} = A\mathbf{z}_*$

LAMDA Learning And Mining from DatA

Optimization after Random Projection

The Overall Idea

$$\mathbb{R}^d \ni \mathbf{x}_i \rightarrow A^\top \mathbf{x}_i \in \mathbb{R}^m$$



$$\mathbb{R}^d \ni A\mathbf{z}_* \leftarrow \mathbf{z}_* \in \mathbb{R}^m$$

Theorem 1 (A Bad News [Zhang et al., 2014])

With a probability at least $1 - 2^{-d+m}$, *we have*

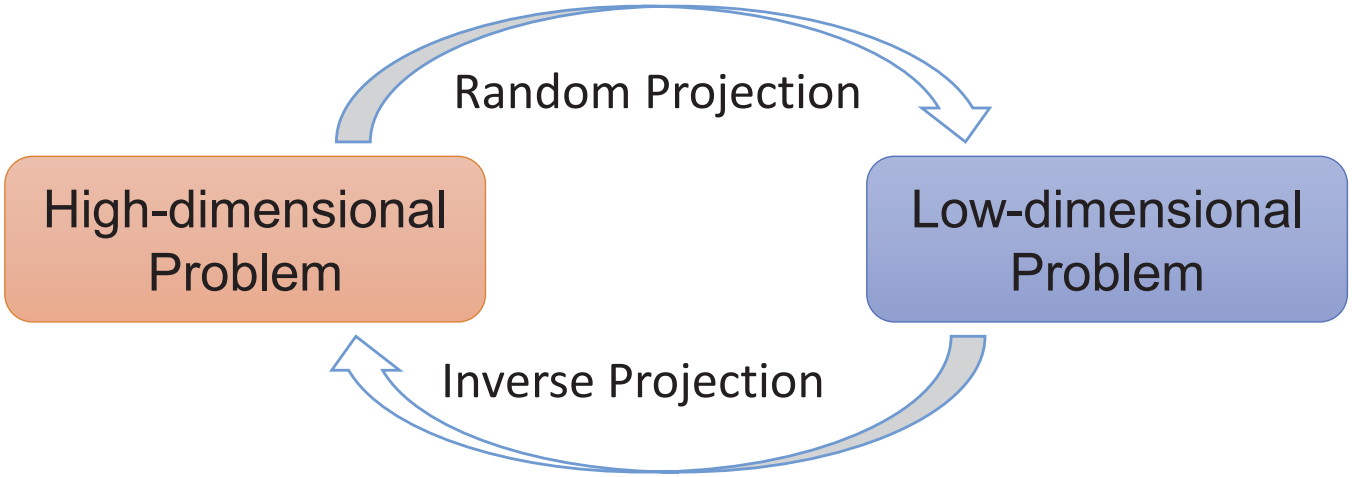
$$\|\widehat{\mathbf{w}} - \mathbf{w}_*\|_2 \geq \frac{1}{2c}\sqrt{\frac{d-m}{d}}\|\mathbf{w}_*\|_2$$

LAMDA Learning And Mining from DatA

Optimization after Random Projection

The Overall Idea

$$\mathbb{R}^d \ni \mathbf{x}_i \rightarrow A^\top \mathbf{x}_i \in \mathbb{R}^m$$



$$\mathbb{R}^d \ni A\mathbf{z}_* \leftarrow \mathbf{z}_* \in \mathbb{R}^m$$

Theorem 1 (A Bad News [Zhang et al., 2014])

With a probability at least $1 - 2^{-d+m}$, *we have*

$$\|\widehat{\mathbf{w}} - \mathbf{w}_*\|_2 \geq \frac{1}{2c}\sqrt{\frac{d-m}{d}}\|\mathbf{w}_*\|_2$$

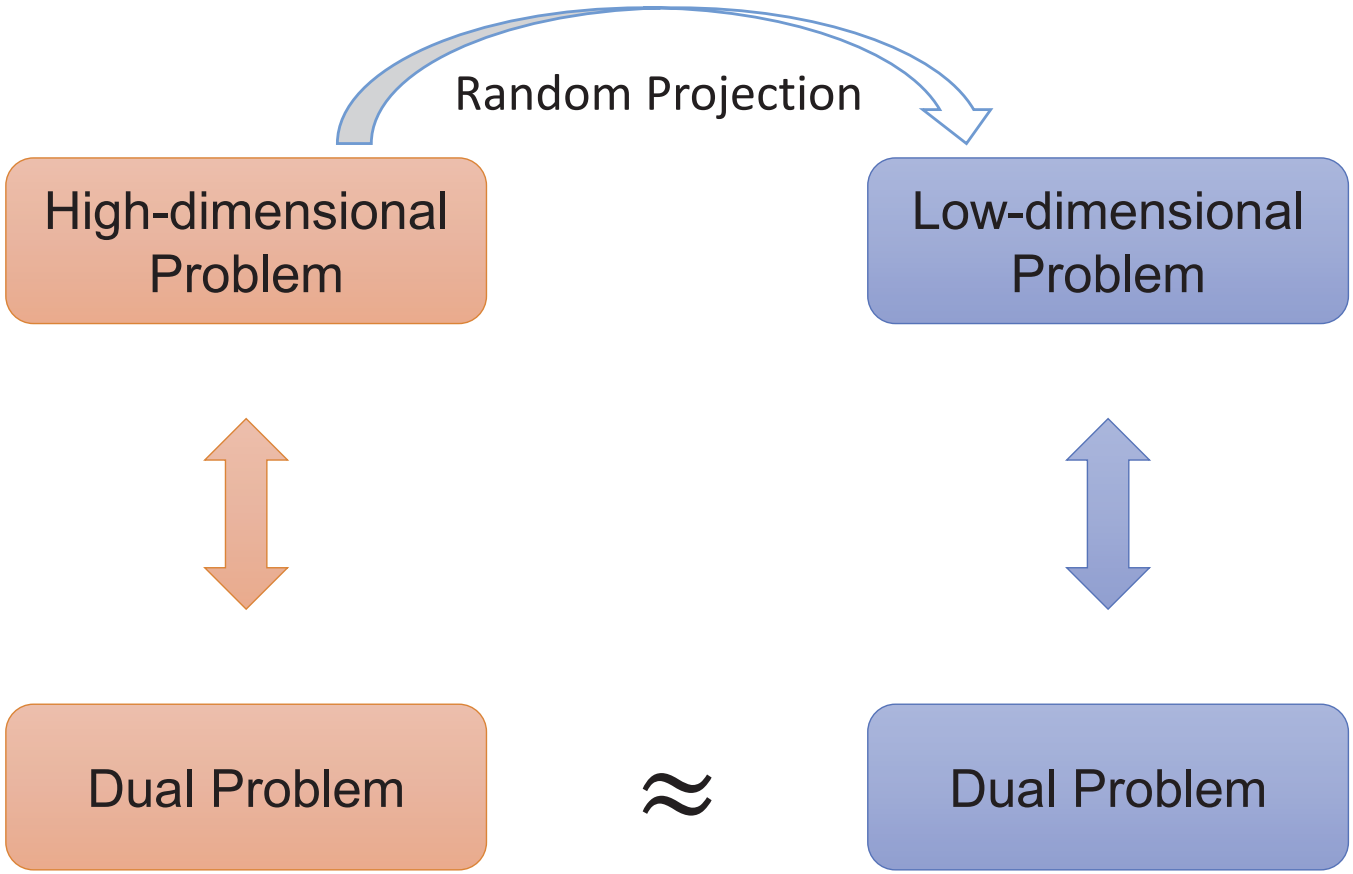
LAMDA Learning And Mining from DatA

Outline

Dual Random Projection

Use Dual Problems to Connect Primal Problems

$$\mathbb{R}^d \ni \mathbf{x}_i \rightarrow A^\top \mathbf{x}_i \in \mathbb{R}^m$$



LAMDA
Learning And Mining from DatA

The Primal Problem in $\mathbb{R}^d$ and Its Dual Problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\mathbf{x}_i^\top\mathbf{w})+\frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

$$\max_{\boldsymbol{\alpha}\in\Omega^n} -\sum_{i=1}^{n}\ell_*(\alpha_i)-\frac{1}{2n\lambda}(\boldsymbol{\alpha}\circ\mathbf{y})^\top X^\top X(\boldsymbol{\alpha}\circ\mathbf{y})$$

The Primal Problem in $\mathbb{R}^m$ and Its Dual Problem

$$\min_{\mathbf{z}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\widehat{\mathbf{x}}_i^\top\mathbf{z})+\frac{\lambda}{2}\|\mathbf{z}\|_2^2$$

$$\max_{\boldsymbol{\beta}\in\Omega^n} -\sum_{i=1}^{n}\ell_*(\beta_i)-\frac{1}{2\lambda n}(\boldsymbol{\beta}\circ\mathbf{y})^\top X^\top AA^\top X(\boldsymbol{\beta}\circ\mathbf{y})$$

The Primal Problem in $\mathbb{R}^d$ and Its Dual Problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\mathbf{x}_i^\top\mathbf{w})+\frac{\lambda}{2}\|\mathbf{w}\|_2^2$$

$$\max_{\boldsymbol{\alpha}\in\Omega^n} -\sum_{i=1}^{n}\ell_*(\alpha_i)-\frac{1}{2n\lambda}(\boldsymbol{\alpha}\circ\mathbf{y})^\top X^\top X(\boldsymbol{\alpha}\circ\mathbf{y})$$

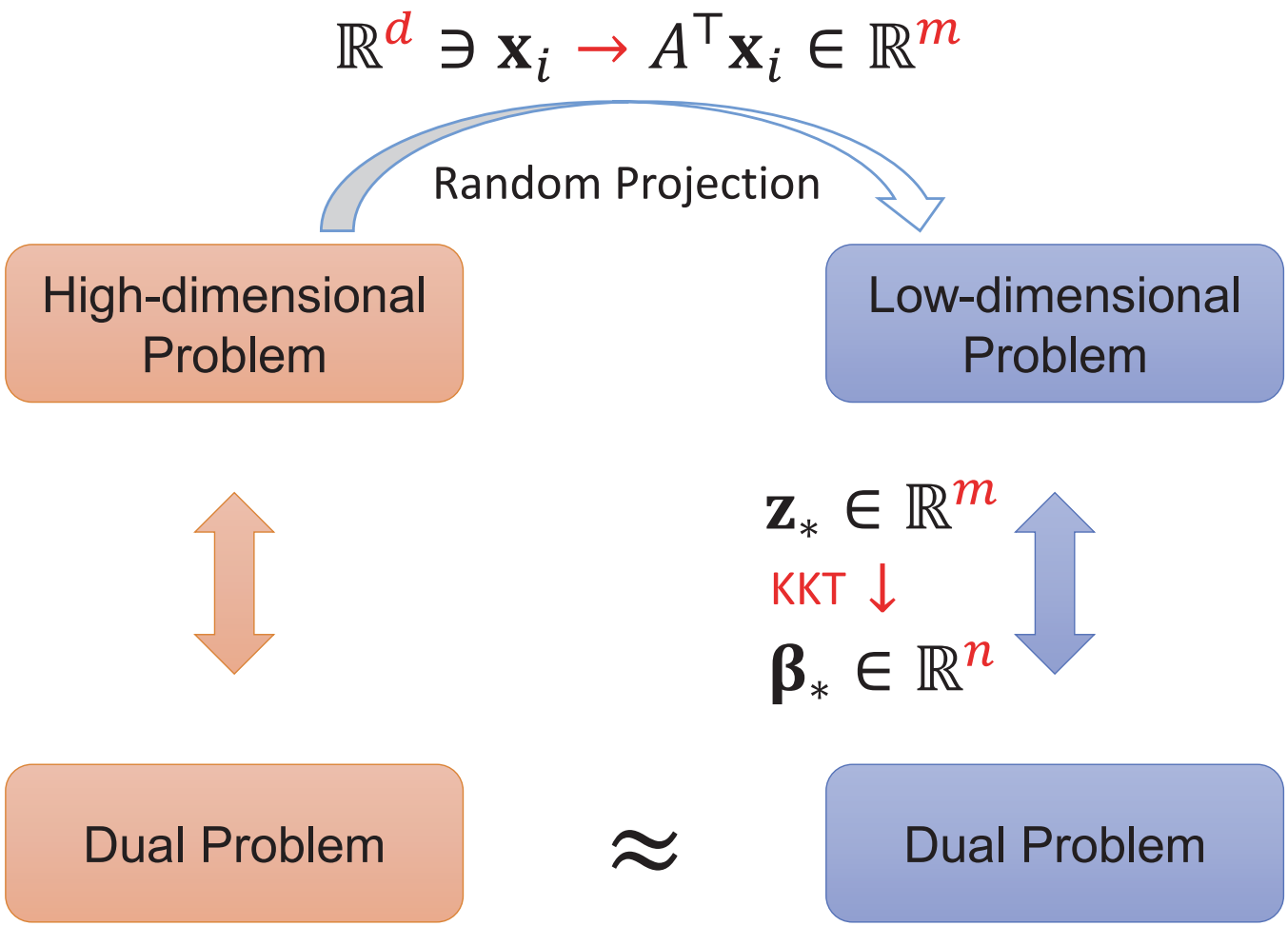
The Primal Problem in $\mathbb{R}^m$ and Its Dual Problem

$$\min_{\mathbf{z}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\widehat{\mathbf{x}}_i^\top\mathbf{z})+\frac{\lambda}{2}\|\mathbf{z}\|_2^2$$

$$\max_{\boldsymbol{\beta}\in\Omega^n} -\sum_{i=1}^{n}\ell_*(\beta_i)-\frac{1}{2\lambda n}(\boldsymbol{\beta}\circ\mathbf{y})^\top X^\top AA^\top X(\boldsymbol{\beta}\circ\mathbf{y})$$

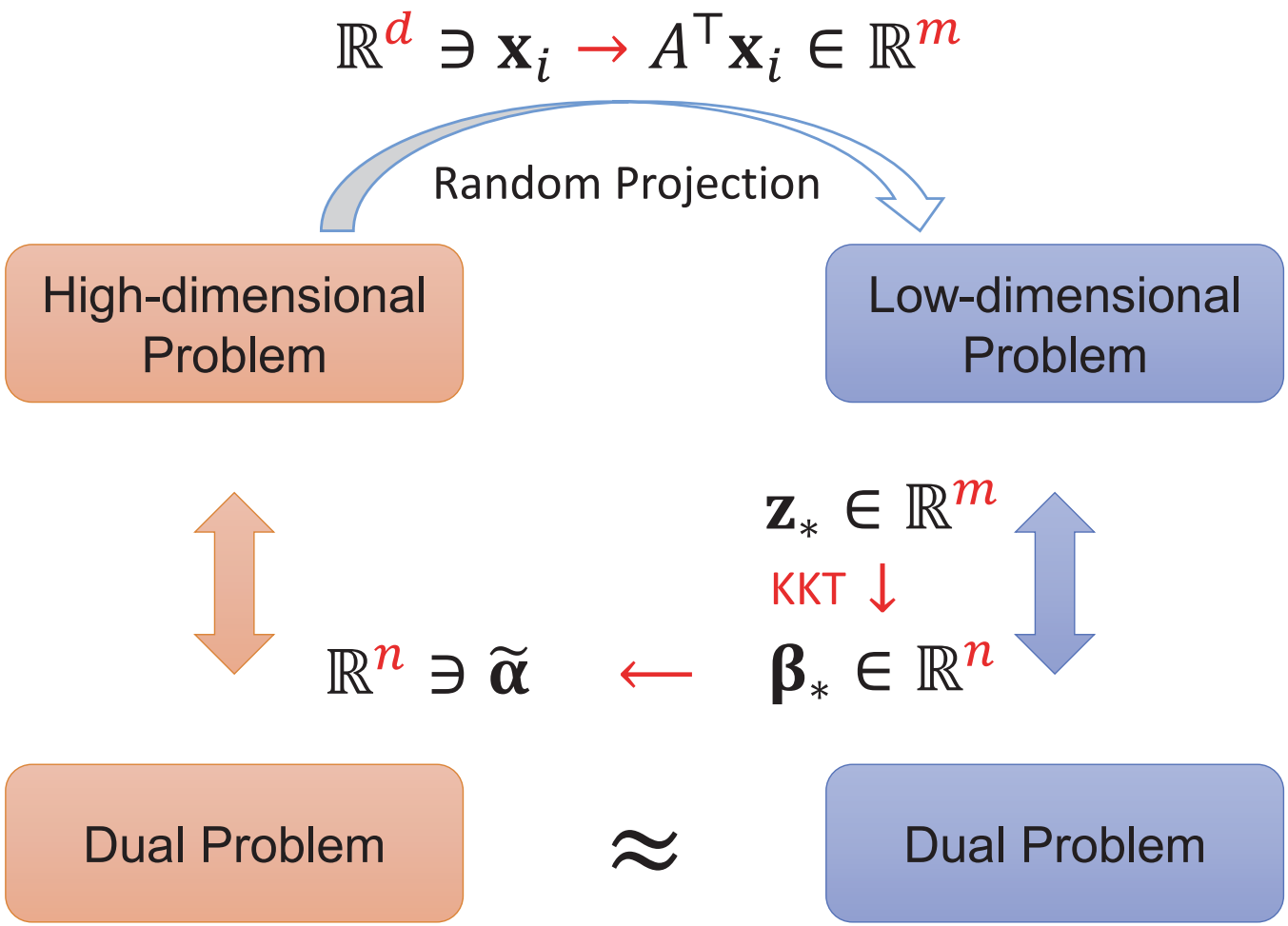
Dual Random Projection

Use Dual Problems to Connect Primal Problems



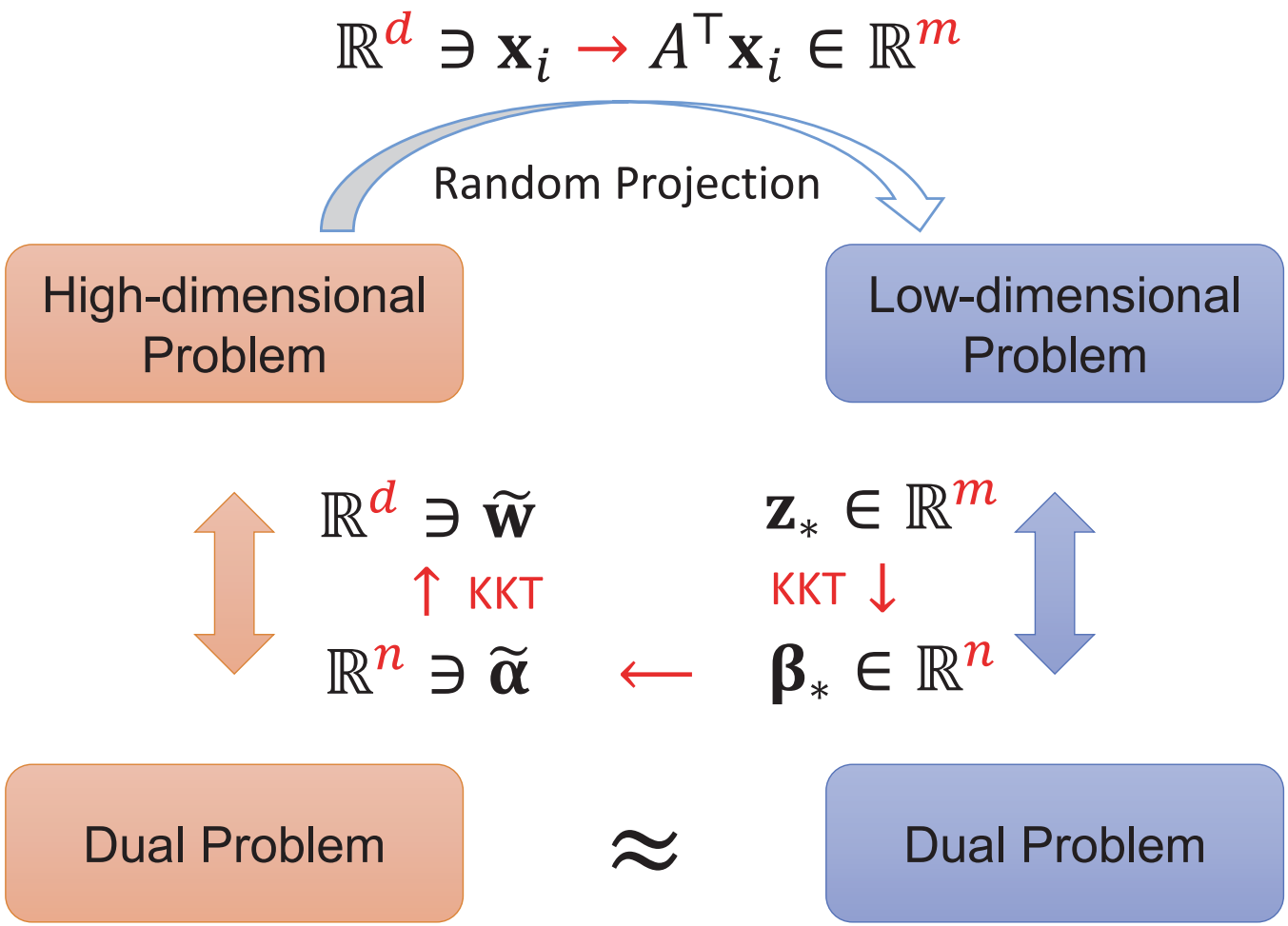
Dual Random Projection

Use Dual Problems to Connect Primal Problems



Dual Random Projection

Use Dual Problems to Connect Primal Problems



LAMDA Learning And Mining from DatA

Theoretical Guarantees

Low-rank Assumption

$r = \text{rank}(X) \ll \min(d, n)$.

Theorem 2 (A Good News[Zhang et al., 2013b, Zhang et al., 2014])

For any $0 < \epsilon \leq 1/2$, *with a probability at least* $1 - \delta$, *we have*

$$\|\widetilde{\mathbf{w}} - \mathbf{w}_*\|_2 \leq \frac{\epsilon}{1-\epsilon}\|\mathbf{w}_*\|_2,$$

provided

$$m \geq \frac{(r+1)\log(2r/\delta)}{c\epsilon^2},$$

where constant c *is at least* $1/4$.

Implication

To accurately recover $\mathbf{w}_*$, the number of required random projections is $\Omega(r \log r)$.

LAMDA Learning And Mining from DatA

Theoretical Guarantees

Low-rank Assumption

$r = \text{rank}(X) \ll \min(d, n)$.

Theorem 2 (A Good News[Zhang et al., 2013b, Zhang et al., 2014])

For any $0 < \epsilon \leq 1/2$, *with a probability at least* $1 - \delta$, *we have*

$$\|\widetilde{\mathbf{w}} - \mathbf{w}_*\|_2 \leq \frac{\epsilon}{1-\epsilon}\|\mathbf{w}_*\|_2,$$

provided

$$m \geq \frac{(r+1)\log(2r/\delta)}{c\epsilon^2},$$

where constant c *is at least* $1/4$.

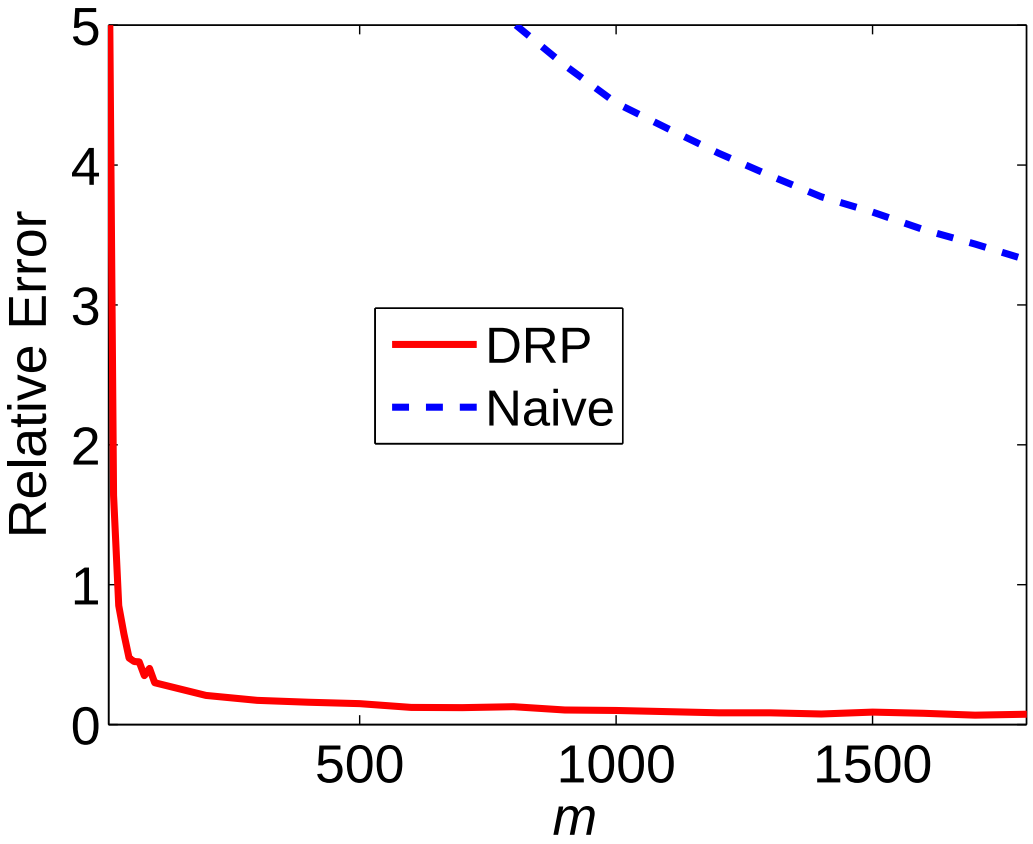
Implication

To accurately recover $\mathbf{w}_*$, the number of required random projections is $\Omega(r \log r)$.

LAMDA
Learning And Mining from DatA

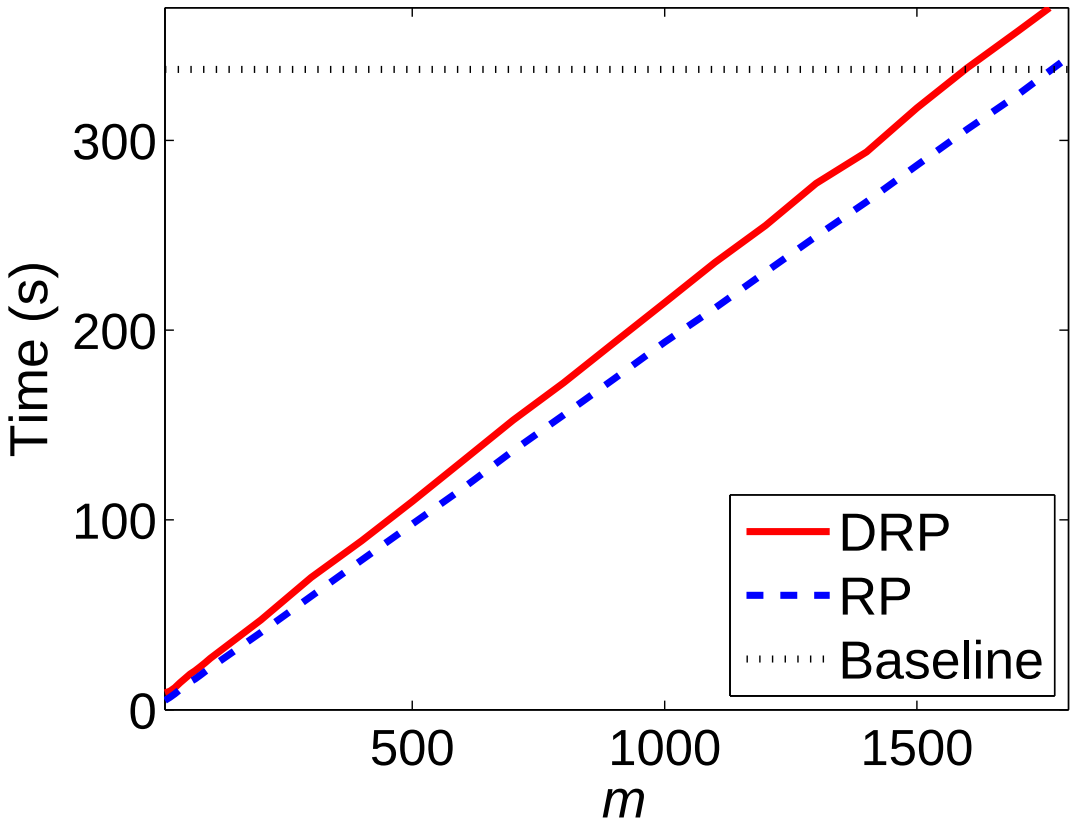
Experimental Results I

- A $20,000 \times 50,000$ data matrix with rank 10.
- The reconstruction error



Experimental Results II

- A $20,000 \times 50,000$ data matrix with rank 10.
- The running time



Outline

LAMDA
Learning And Mining from DatA

Background

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

- Both m and n can be very large
- The optimal solution W_* is low-rank

Low-rank Matrix Completion

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = \sum_{(i,j) \in \Omega} (W_{ij} - M_{ij})^2 + \lambda \|W\|_*$$

- There are m users and n items
- $M \in \mathbb{R}^{m \times n}$ is the underlying user-item rating matrix
- Ω is the set of observed indices

LAMDA Learning And Mining from DatA

Background

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

- Both m and n can be very large
- The optimal solution W_* is low-rank

Low-rank Matrix Completion

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = \sum_{(i,j) \in \Omega} (W_{ij} - M_{ij})^2 + \lambda \|W\|_*$$

- There are m users and n items
- $M \in \mathbb{R}^{m \times n}$ is the underlying user-item rating matrix
- Ω is the set of observed indices

LAMDA
Learning And Mining from DatA

Background

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

- Both m and n can be very large
- The optimal solution W_* is low-rank

Low-rank Matrix Approximation

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = d(W, M) + \lambda \|W\|_*$$

- $M \in \mathbb{R}^{m \times n}$ is a given matrix
- $d(\cdot, \cdot)$ is a distance function, such as

$$d(W, M) = \|W - M\|_F^2, \quad \|W - M\|_1$$

LAMDA Learning And Mining from DatA

Background

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

- Both m and n can be very large
- The optimal solution W_* is low-rank

Multi-class Classification

$$\min_{W \in \mathbb{R}^{d \times k}} \frac{1}{n} \sum_{i=1}^{n} \log \left(1 + \sum_{j \in \mathcal{Y} \setminus \{y_i\}} \exp \left(\mathbf{w}_j^\top \mathbf{x}_i - \mathbf{w}_{y_i}^\top \mathbf{x}_i \right) \right) + \lambda \|W\|_*$$

- $W = [\mathbf{w}_1, \ldots, \mathbf{w}_k] \in \mathbb{R}^{d \times k}$ is a set of k classifiers
- $\mathcal{Y} = \{1, 2, \ldots, k\}$ is the set of class labels
- $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are training data

LAMDA Learning And Mining from DatA

Proximal Gradient Descent (PGD)

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

Proximal Gradient Descent (PGD)

1: **for** $t = 1, 2, \ldots, T$ **do**

2:

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \|W - (W_t - \eta_t \nabla f(W_t))\|_F^2 + \eta_t \lambda \|W\|_*$$

3: **end for**

Convergence Rates [Nesterov, 2007]

	Convex & Smooth	Strongly Convex & Smooth
Rate	$O\left(\frac{1}{T}\right) \rightarrow O\left(\frac{1}{T^2}\right)$	$O\left(\frac{1}{\alpha^T}\right)$

LAMDA Learning And Mining from DatA

Proximal Gradient Descent (PGD)

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

Proximal Gradient Descent (PGD)

1: **for** $t = 1, 2, \ldots, T$ **do**

2:

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \|W - (W_t - \eta_t \nabla f(W_t))\|_F^2 + \eta_t \lambda \|W\|_*$$

3: **end for**

Convergence Rates [Nesterov, 2007]

	Convex & Smooth	Strongly Convex & Smooth
Rate	$O\left(\frac{1}{T}\right) \rightarrow O\left(\frac{1}{T^2}\right)$	$O\left(\frac{1}{\alpha^T}\right)$

LAMDA Learning And Mining from DatA

Proximal Gradient Descent (PGD)

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

Proximal Gradient Descent (PGD)

1: **for** $t = 1, 2, \ldots, T$ **do**

2:

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \|W - (W_t - \eta_t \nabla f(W_t))\|_F^2 + \eta_t \lambda \|W\|_*$$

3: **end for**

Convergence Rates [Nesterov, 2007]

	Convex & Smooth	Strongly Convex & Smooth
Rate	$O\left(\frac{1}{T}\right) \to O\left(\frac{1}{T^2}\right)$	$O\left(\frac{1}{\alpha^T}\right)$

LAMDA Learning And Mining from DatA

Implementation Of PGD

Solution of PGD [Cai et al., 2010]

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \|W - (W_t - \eta_t \nabla f(W_t))\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[W_t - \eta_t \nabla f(W_t)]$$

Singular Value Shrinkage (SVS) Operator

For a matrix $Y \in \mathbb{R}^{m \times n}$ with singular value decomposition (SVD)

$$Y = U \Sigma V^\top$$

where

$$U = [\mathbf{u}_1, \ldots, \mathbf{u}_r],\ \Sigma = \operatorname{diag}[\sigma_1, \ldots, \sigma_r],\ \text{and}\ V = [\mathbf{v}_1, \ldots, \mathbf{v}_r].$$

The SVS operator with threshold λ is defined as

$$\mathcal{D}_\lambda[Y] = \sum_{i:\sigma_i \geq \lambda} (\sigma_i - \lambda) \mathbf{u}_i \mathbf{v}_i^\top$$

LAMDA Learning And Mining from DatA

Implementation Of PGD

Solution of PGD [Cai et al., 2010]

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \|W - (W_t - \eta_t \nabla f(W_t))\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[W_t - \eta_t \nabla f(W_t)]$$

Singular Value Shrinkage (SVS) Operator

For a matrix $Y \in \mathbb{R}^{m \times n}$ with singular value decomposition (SVD)

$$Y = U \Sigma V^\top$$

where

$$U = [\mathbf{u}_1, \ldots, \mathbf{u}_r],\ \Sigma = \operatorname{diag}[\sigma_1, \ldots, \sigma_r],\ \text{and}\ V = [\mathbf{v}_1, \ldots, \mathbf{v}_r].$$

The SVS operator with threshold λ is defined as

$$\mathcal{D}_\lambda[Y] = \sum_{i:\sigma_i \geq \lambda} (\sigma_i - \lambda) \mathbf{u}_i \mathbf{v}_i^\top$$

LAMDA Learning And Mining from DatA

Implementation Of PGD

Solution of PGD [Cai et al., 2010]

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \|W - (W_t - \eta_t \nabla f(W_t))\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[W_t - \eta_t \nabla f(W_t)]$$

Computational Cost

- Space Complexity: $O(mn)$
- Time Complexity: $O(mn^2)$ or $O(m^2 n)$

Implementation Of PGD

Solution of PGD [Cai et al., 2010]

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \|W - (W_t - \eta_t \nabla f(W_t))\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[W_t - \eta_t \nabla f(W_t)]$$

Computational Cost

- Space Complexity: $O(mn)$
- Time Complexity: $O(mn^2)$ or $O(m^2 n)$

The Challenge

- Expensive if both m and n are very large

Outline

LAMDA
Learning And Mining from DatA

Stochastic Proximal Gradient Descent (SPGD)

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

The Algorithm [Zhang et al., 2015]

1: **for** $t = 1, 2, \ldots, T$ **do**

2: Generate a low-rank stochastic gradient $\widehat{G}_t$ of $f(\cdot)$ at W_t

3:

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \left\| W - (W_t - \eta_t \widehat{G}_t) \right\|_F^2 + \eta_t \lambda \|W\|_*$$

4: **end for**

5: **return** W_{T+1}

Efficient Implementation

- The above problem can be solved by incremental SVD with $O((m+n)r)$ space and $O((m+n)r^2)$ time

LAMDA Learning And Mining from DatA

Stochastic Proximal Gradient Descent (SPGD)

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

The Algorithm [Zhang et al., 2015]

1: **for** $t = 1, 2, \ldots, T$ **do**

2:

3: Generate a low-rank stochastic gradient $\widehat{G}_t$ of $f(\cdot)$ at W_t

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \left\| W - (W_t - \eta_t \widehat{G}_t) \right\|_F^2 + \eta_t \lambda \|W\|_*$$

4: **end for**

5: **return** W_{T+1}

Efficient Implementation

- The above problem can be solved by incremental SVD with $O((m+n)r)$ space and $O((m+n)r^2)$ time

LAMDA Learning And Mining from DatA

Stochastic Proximal Gradient Descent (SPGD)

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

The Algorithm [Zhang et al., 2015]

1: **for** $t = 1, 2, \ldots, T$ **do**

2:

3: Generate a low-rank stochastic gradient $\widehat{G}_t$ of $f(\cdot)$ at W_t

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \left\| W - (W_t - \eta_t \widehat{G}_t) \right\|_F^2 + \eta_t \lambda \|W\|_*$$

4: **end for**

5: **return** W_{T+1}

Efficient Implementation

- The above problem can be solved by incremental SVD with $O((m+n)r)$ space and $O((m+n)r^2)$ time

LAMDA Learning And Mining from DatA

Stochastic Proximal Gradient Descent (SPGD)

Nuclear Norm Regularized Optimization over Matrices

$$\min_{W \in \mathbb{R}^{m \times n}} F(W) = f(W) + \lambda \|W\|_*$$

The Algorithm [Zhang et al., 2015]

1: **for** $t = 1, 2, \ldots, T$ **do**

2:

3: Generate a low-rank stochastic gradient $\widehat{G}_t$ of $f(\cdot)$ at W_t

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \left\| W - (W_t - \eta_t \widehat{G}_t) \right\|_F^2 + \eta_t \lambda \|W\|_*$$

4: **end for**

5: **return** W_{T+1}

Efficient Implementation

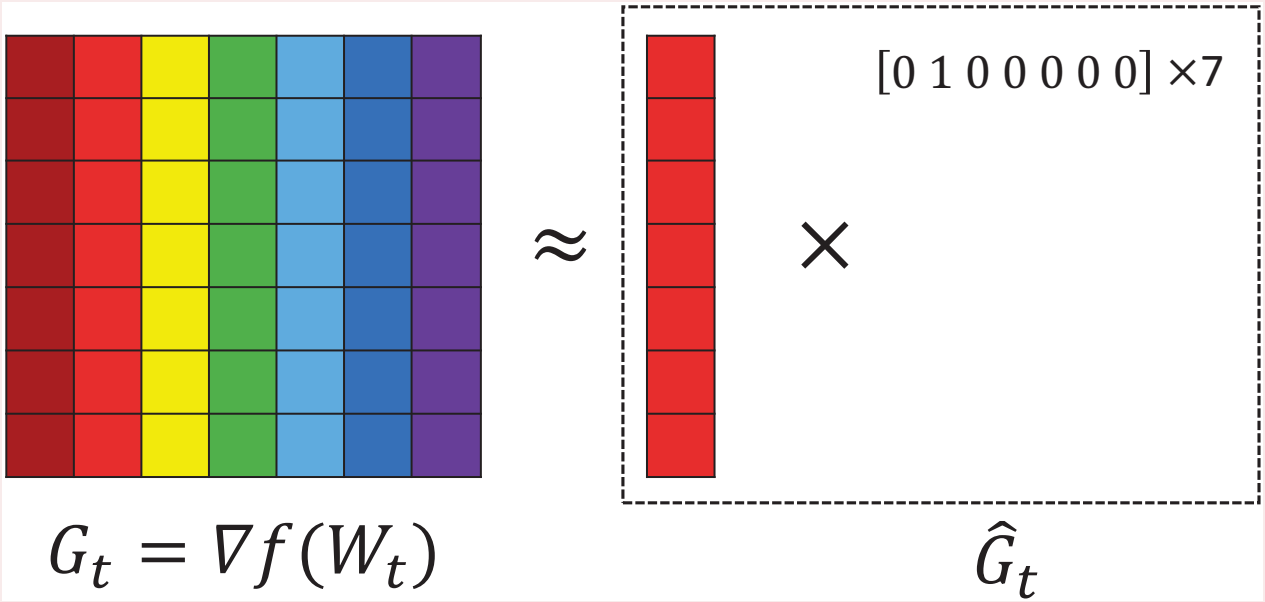
- The above problem can be solved by incremental SVD with $O((m+n)r)$ space and $O((m+n)r^2)$ time

LAMDA Learning And Mining from DatA

Implementation—Low-rank Stochastic Gradient $\widehat{G}_t$

Random Sampling [Avron et al., 2012]

- Low-rank



- Stochastic $G_t = \mathrm{E}[\widehat{G}_t]$

Random Projection [Chen et al., 2014]

- Generate a random matrix $Z \in \mathbb{R}^{n \times k}$ such that $\mathrm{E}[ZZ^\top] = I$

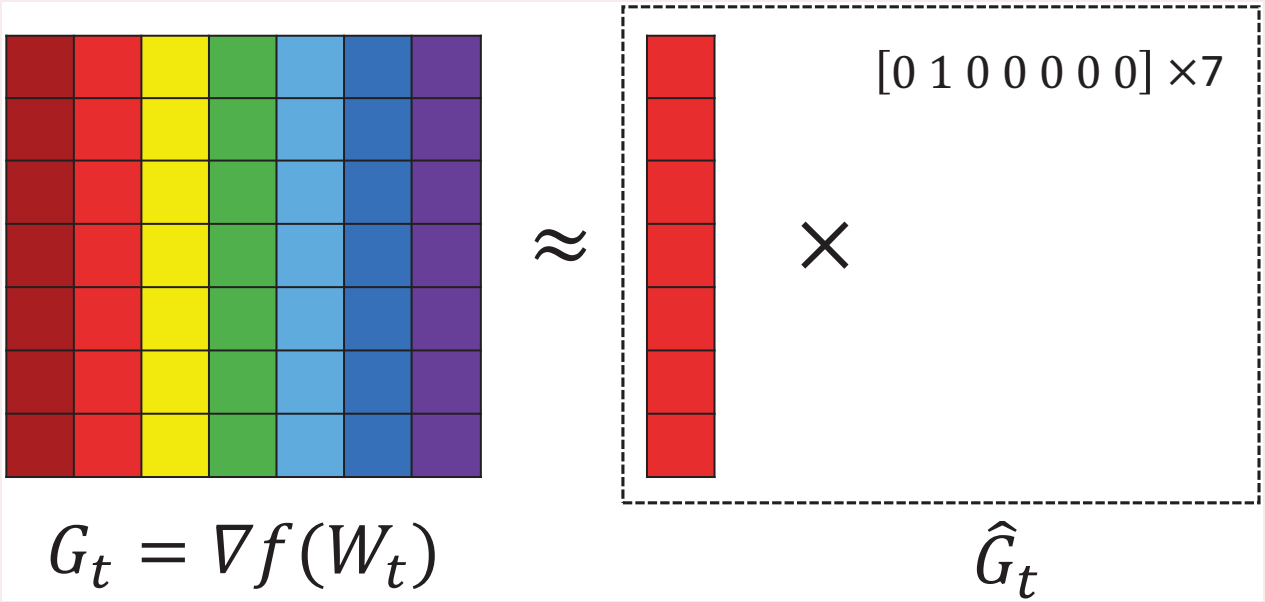
$$\widehat{G}_t = G_t Z Z^\top = (G_t Z) Z^\top$$

LAMDA Learning And Mining from DatA

Implementation—Low-rank Stochastic Gradient $\widehat{G}_t$

Random Sampling [Avron et al., 2012]

- Low-rank



- Stochastic $G_t = \mathrm{E}[\widehat{G}_t]$

Random Projection [Chen et al., 2014]

- Generate a random matrix $Z \in \mathbb{R}^{n \times k}$ such that $\mathrm{E}[ZZ^\top] = I$

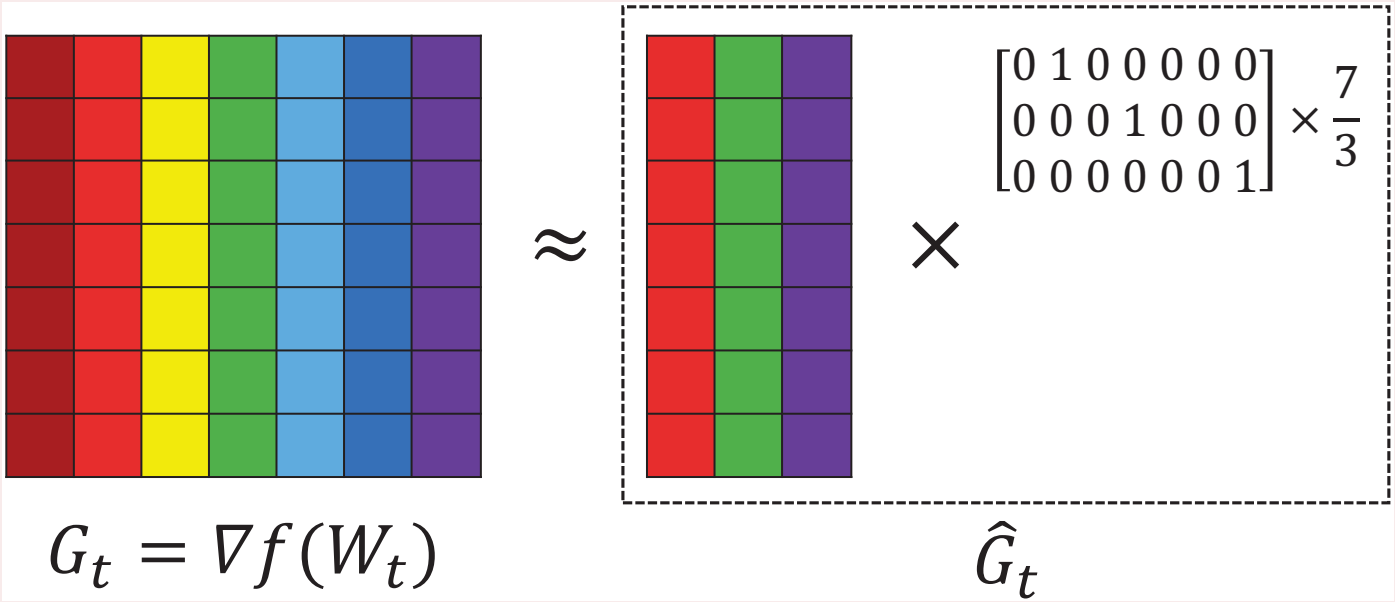
$$\widehat{G}_t = G_t Z Z^\top = (G_t Z) Z^\top$$

LAMDA Learning And Mining from DatA

Implementation—Low-rank Stochastic Gradient $\widehat{G}_t$

Random Sampling [Avron et al., 2012]

- Low-rank



- Stochastic $G_t = \mathrm{E}[\widehat{G}_t]$

Random Projection [Chen et al., 2014]

- Generate a random matrix $Z \in \mathbb{R}^{n \times k}$ such that $\mathrm{E}[ZZ^\top] = I$

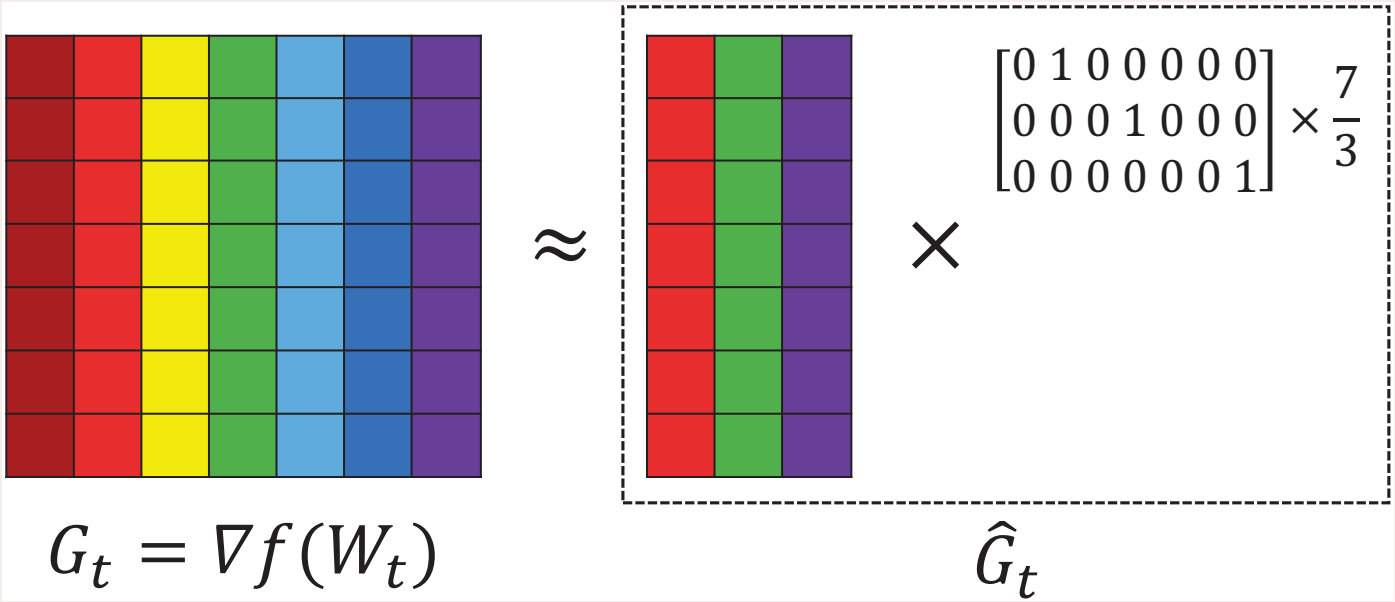
$$\widehat{G}_t = G_t Z Z^\top = (G_t Z) Z^\top$$

LAMDA Learning And Mining from DatA

Implementation—Low-rank Stochastic Gradient $\widehat{G}_t$

Random Sampling [Avron et al., 2012]

- Low-rank



- Stochastic $G_t = \mathrm{E}[\widehat{G}_t]$

Random Projection [Chen et al., 2014]

- Generate a random matrix $Z \in \mathbb{R}^{n \times k}$ such that $\mathrm{E}[ZZ^\top] = I$

$$\widehat{G}_t = G_t Z Z^\top = (G_t Z) Z^\top$$

LAMDA Learning And Mining from DatA

Implementation—SPGD

Solution of SPGD [Cai et al., 2010]

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \left\| W - (W_t - \eta_t \widehat{G}_t) \right\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[W_t - \eta_t \widehat{G}_t]$$

Procedure

1. Calculate the SVD of $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
   - Incremental SVD [Brand, 2006, Section 2]
2. Calculate $W_{t+1} = \mathcal{D}_{\lambda \eta_t}[W'_{t+1}]$

Computational Complexities

- Time complexity: $O\left((m+n)(r+k)^2 + (r+k)^3\right)$
- Space complexity: $O\left((m+n)(r+k)\right)$

LAMDA Learning And Mining from DatA

Implementation—SPGD

Solution of SPGD [Cai et al., 2010]

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \left\| W - (W_t - \eta_t \widehat{G}_t) \right\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[W_t - \eta_t \widehat{G}_t]$$

Procedure

1. Calculate the SVD of $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
   - Incremental SVD [Brand, 2006, Section 2]
2. Calculate $W_{t+1} = \mathcal{D}_{\lambda \eta_t}[W'_{t+1}]$

Computational Complexities

- Time complexity: $O\left((m+n)(r+k)^2 + (r+k)^3\right)$
- Space complexity: $O\left((m+n)(r+k)\right)$

LAMDA Learning And Mining from DatA

Implementation—SPGD

Solution of SPGD [Cai et al., 2010]

$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2} \left\| W - (W_t - \eta_t \widehat{G}_t) \right\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[W_t - \eta_t \widehat{G}_t]$$

Procedure

1. Calculate the SVD of $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
   - Incremental SVD [Brand, 2006, Section 2]
2. Calculate $W_{t+1} = \mathcal{D}_{\lambda \eta_t}[W'_{t+1}]$

Computational Complexities

- Time complexity: $O\left((m+n)(r+k)^2 + (r+k)^3\right)$
- Space complexity: $O\left((m+n)(r+k)\right)$

LAMDA Learning And Mining from DatA

Convergence Rates

Theorem 3 (General convex functions [Zhang et al., 2015])

Assume $\mathrm{E}[\|\widehat{G}_t\|_F^2]$ *is upper bounded. Setting* $\eta_t = 1/\sqrt{t}$*, we have*

$$\mathrm{E}\left[F(W_T) - F(W_*)\right] \leq O\left(\frac{\log T}{\sqrt{T}}\right)$$

where W_* *is the optimal solution.*

Theorem 4 (Strongly convex functions [Zhang et al., 2015])

Suppose $f(\cdot)$ *is strongly convex, and* $\mathrm{E}[\|\widehat{G}_t\|_F^2]$ *is upper bounded. Setting* $\eta_t = 1/(\mu t)$*, we have*

$$\mathrm{E}\left[F(W_T) - F(W_*)\right] \leq O\left(\frac{\log T}{T}\right)$$

where W_* *is the optimal solution.*

LAMDA Learning And Mining from DatA

Convergence Rates

Theorem 3 (General convex functions [Zhang et al., 2015])

Assume $\mathrm{E}[\|\widehat{G}_t\|_F^2]$ *is upper bounded. Setting* $\eta_t = 1/\sqrt{t}$*, we have*

$$\mathrm{E}\left[F(W_T) - F(W_*)\right] \leq O\left(\frac{\log T}{\sqrt{T}}\right)$$

where W_* *is the optimal solution.*

Theorem 4 (Strongly convex functions [Zhang et al., 2015])

Suppose $f(\cdot)$ *is strongly convex, and* $\mathrm{E}[\|\widehat{G}_t\|_F^2]$ *is upper bounded. Setting* $\eta_t = 1/(\mu t)$*, we have*

$$\mathrm{E}\left[F(W_T) - F(W_*)\right] \leq O\left(\frac{\log T}{T}\right)$$

where W_* *is the optimal solution.*

LAMDA Learning And Mining from DatA

Kernel PCA

Traditional Algorithm of Kernel PCA

1. Construct a kernel matrix $K \in R^{n \times n}$ with $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$
2. Calculate the top eigenvectors and eigenvalues of K

- Space complexity: $O(n^2)$; Time complexity: $O(n^3)$

Nuclear Norm Regularized Least Squares

$$\min_{W \in \mathbb{R}^{n \times n}} \frac{1}{2}\|W - K\|_F^2 + \lambda \|W\|_*$$

- Top eigensystems of $K = U \Lambda U^\top$ can be recovered from

$$W_* = \mathcal{D}_\lambda[K] = \sum_{i:\lambda_i > \lambda} (\lambda_i - \lambda)\mathbf{u}_i \mathbf{u}_i^\top$$

LAMDA Learning And Mining from DatA

Kernel PCA

Traditional Algorithm of Kernel PCA

1. Construct a kernel matrix $K \in R^{n \times n}$ with $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$
2. Calculate the top eigenvectors and eigenvalues of K

- Space complexity: $O(n^2)$; Time complexity: $O(n^3)$

Nuclear Norm Regularized Least Squares

$$\min_{W \in \mathbb{R}^{n \times n}} \frac{1}{2}\|W - K\|_F^2 + \lambda \|W\|_*$$

- Top eigensystems of $K = U \Lambda U^\top$ can be recovered from

$$W_* = \mathcal{D}_\lambda[K] = \sum_{i:\lambda_i > \lambda} (\lambda_i - \lambda)\mathbf{u}_i \mathbf{u}_i^\top$$

LAMDA Learning And Mining from DatA

Application to Kernel PCA

Nuclear Norm Regularized Least Squares

$$\min_{W \in \mathbb{R}^{n \times n}} \frac{1}{2}\|W - K\|_F^2 + \lambda \|W\|_*$$

Stochastic Proximal Gradient Descent [Zhang et al., 2016]

1: **for** $t = 1, 2, \ldots, T$ **do**

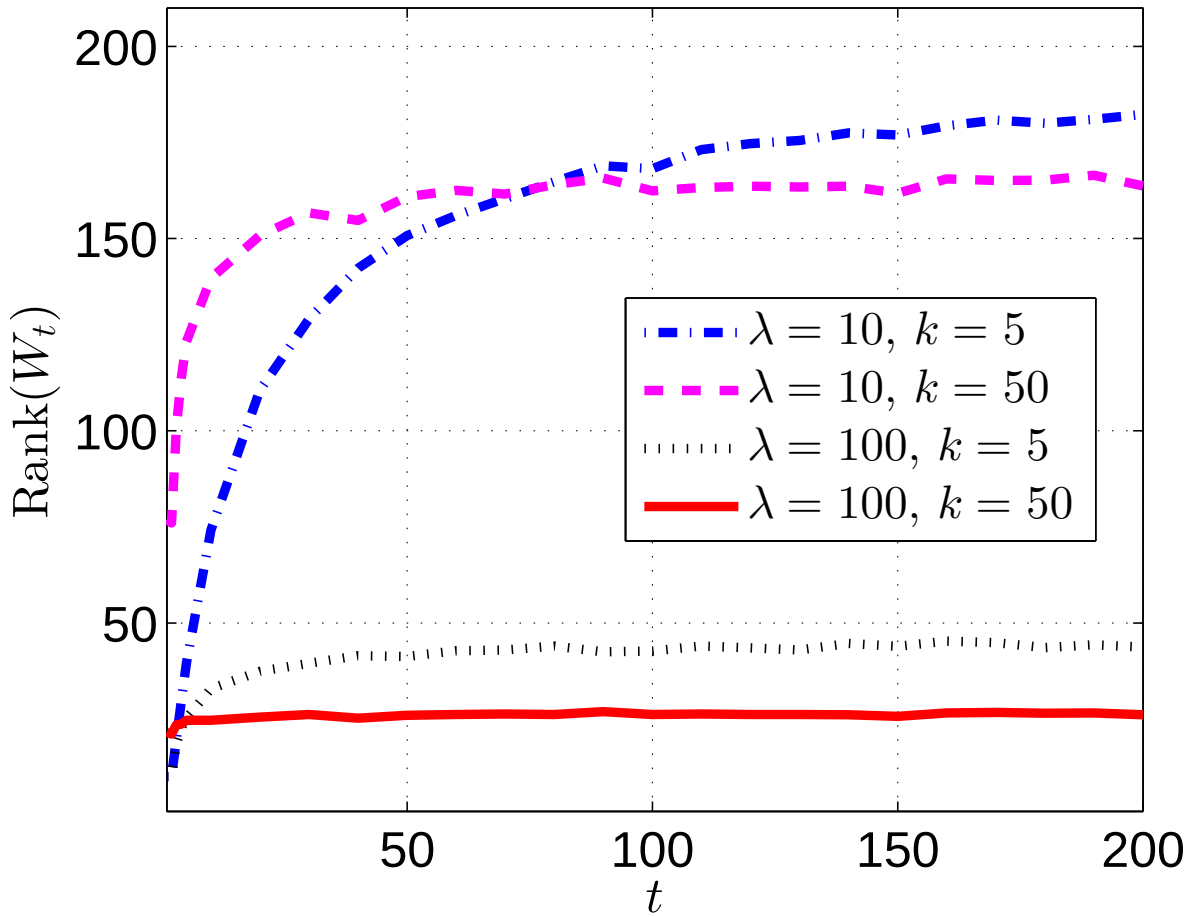
2: Calculate a low-rank stochastic gradient $\widehat{G}_t = W_t - \xi$

3:
$$\mathrm{rank}(\xi) \ll n \text{ and } \mathrm{E}[\xi] = K$$
$$W_{t+1} = \operatorname*{argmin}_{W \in \mathbb{R}^{m \times n}} \frac{1}{2}\left\|W - (W_t - \eta_t \widehat{G}_t)\right\|_F^2 + \eta_t \lambda \|W\|_*$$
$$= \mathcal{D}_{\lambda \eta_t}[(1 - \eta_t) W_t - \eta_t \xi]$$

4: **end for**

Experimental Results (I)

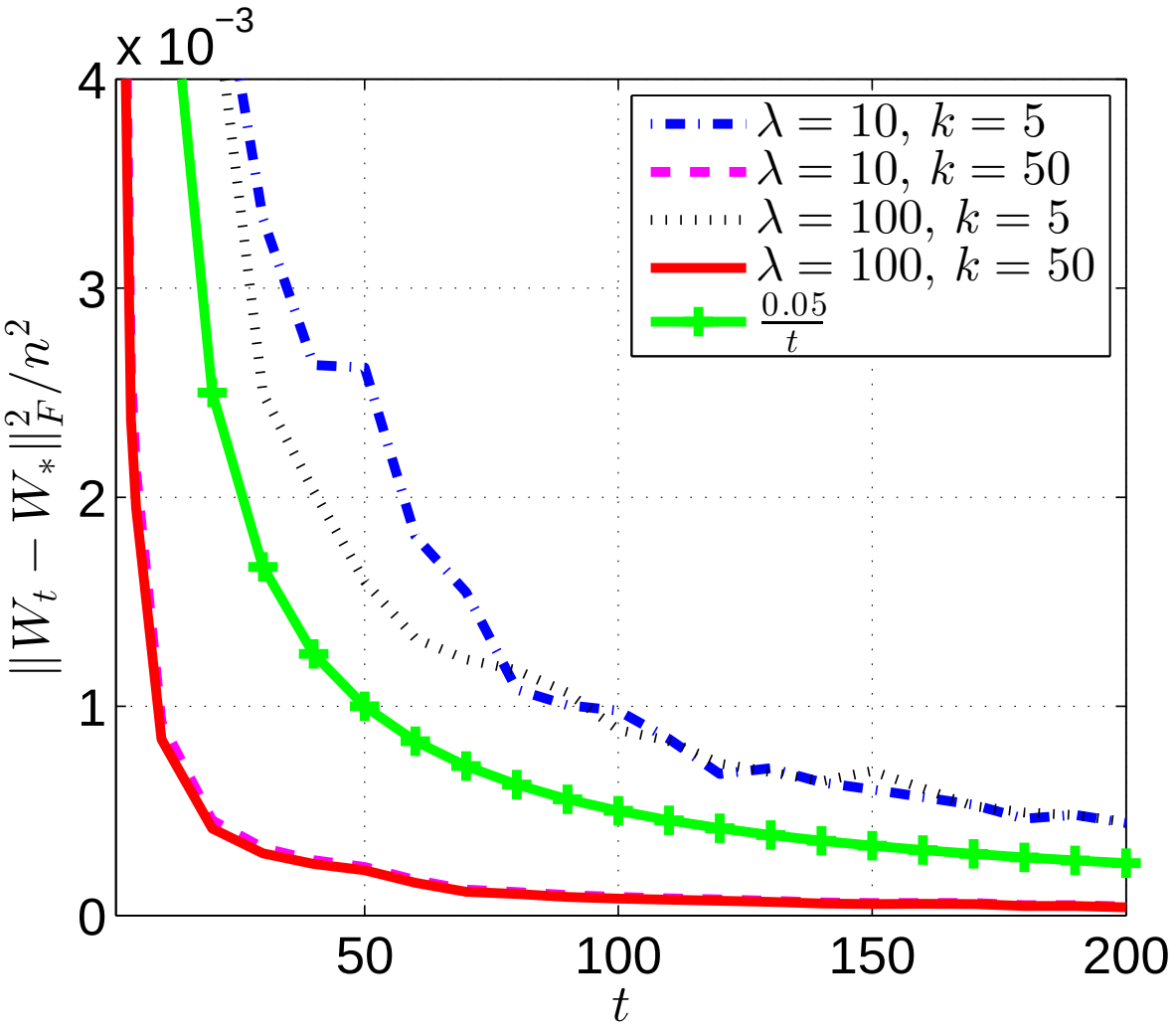
- The Magic data set ($n = 19,020$)
- The rank of each iterate



LAMDA
Learning And Mining from DatA

Experimental Results (II)

- The Magic data set ($n = 19,020$)
- The optimization error



LAMDA Learning And Mining from DatA

Conclusion and Future Work

Randomized Algorithms

- ℓ_2-norm Regularized Convex Optimization
  - An approximate algorithm based on random projection
- Nuclear-norm Regularized Convex Optimization
  - An iterative algorithm based on random sampling

Future Work

- Design random algorithms for non-convex problems

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

- Non-convex loss: ramp loss
- Non-convex regularizer: $\|\cdot\|_0$, $\mathrm{Rank}(\cdot)$

LAMDA
Learning And Mining from DatA

Conclusion and Future Work

Randomized Algorithms

- ℓ_2-norm Regularized Convex Optimization
  - An approximate algorithm based on random projection
- Nuclear-norm Regularized Convex Optimization
  - An iterative algorithm based on random sampling

Future Work

- Design random algorithms for non-convex problems

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w}) + \Omega(\mathbf{w})$$

- Non-convex loss: ramp loss
- Non-convex regularizer: $\|\cdot\|_0$, $\mathrm{Rank}(\cdot)$

LAMDA
Learning And Mining from DatA

Reference I

Thanks!

Avron, H., Kale, S., Kasiviswanathan, S., and Sindhwani, V. (2012).
Efficient and practical stochastic subgradient descent for nuclear norm regularization.
In *Proceedings of the 29th International Conference on Machine Learning*, pages 1231–1238.

Brand, M. (2006).
Fast low-rank modifications of the thin singular value decomposition.
Linear Algebra and its Applications, 415(1):20–30.

Cai, J.-F., Candès, E. J., and Shen, Z. (2010).
A singular value thresholding algorithm for matrix completion.
SIAM Journal on Optimization, 20(4):1956–1982.

Chen, J., Yang, T., and Zhu, S. (2014).
Efficient low-rank stochastic gradient descent methods for solving semidefinite programs.
In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 122–130.

Mahdavi, M., Zhang, L., and Jin, R. (2013).
Mixed optimization for smooth functions.
In *Advance in Neural Information Processing Systems 26 (NIPS)*, pages 674–682.

Nesterov, Y. (2004).
Introductory lectures on convex optimization: a basic course, volume 87 of *Applied optimization*.
Kluwer Academic Publishers.

LAMDA
Learning And Mining from DatA

Reference II

Nesterov, Y. (2007).

Gradient methods for minimizing composite objective function.
Core discussion papers.

Zhang, L., Mahdavi, M., and Jin, R. (2013a).

Linear convergence with condition number independent access of full gradients.
In *Advance in Neural Information Processing Systems 26*, pages 980–988.

Zhang, L., Mahdavi, M., Jin, R., Yang, T., and Zhu, S. (2013b).

Recovering the optimal solution by dual random projection.
In *Proceedings of the 26th Annual Conference on Learning Theory (COLT)*, pages 135–157.

Zhang, L., Mahdavi, M., Jin, R., Yang, T., and Zhu, S. (2014).

Random projections for classification: A recovery approach.
IEEE Transactions on Information Theory, 60(11):7300–7316.

Zhang, L., Yang, T., Jin, R., and Zhou, Z.-H. (2015).

Stochastic proximal gradient descent for nuclear norm regularization.
ArXiv e-prints, arXiv:1511.01664.

Zhang, L., Yang, T., Yi, J., Jin, R., and Zhou, Z.-H. (2016).

Stochastic optimization for kernel pca.
In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

LAMDA
Learning And Mining from DatA