Stochastic Optimization for Large-scale Machine Learning

Lijun Zhang

LAMDA group, Nanjing University, China

The 13rd Chinese Workshop on Machine Learning and Applications

Outline

LAMDA
Learning And Mining from DatA

Outline

What is Stochastic Optimization? (I)

Definition 1: A Special Objective [Nemirovski et al., 2009]

$$\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) = \mathrm{E}_{\xi}\left[\ell(\mathbf{w},\xi)\right] = \int_{\Xi} \ell(\mathbf{w},\xi) dP(\xi)$$

where ξ is a random variable

- It is possible to generate an i.i.d. sample $\xi_1, \xi_2, \ldots$

Risk Minimization

$$\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) = \mathrm{E}_{(\mathbf{x},y)}\left[\ell(\mathbf{w},(\mathbf{x},y))\right]$$

$\ell(\cdot,\cdot)$ is a loss function, e.g., hinge loss $\ell(u,v) = \max(0, 1-uv)$

- Training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are i.i.d.

LAMDA Learning And Mining from DatA

What is Stochastic Optimization? (I)

Definition 1: A Special Objective [Nemirovski et al., 2009]

$$\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) = \mathrm{E}_{\xi}\left[\ell(\mathbf{w},\xi)\right] = \int_{\Xi} \ell(\mathbf{w},\xi) dP(\xi)$$

where ξ is a random variable

- It is possible to generate an i.i.d. sample $\xi_1, \xi_2, \ldots$

Risk Minimization

$$\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) = \mathrm{E}_{(\mathbf{x},y)}\left[\ell(\mathbf{w},(\mathbf{x},y))\right]$$

$\ell(\cdot,\cdot)$ is a loss function, e.g., hinge loss $\ell(u,v) = \max(0, 1-uv)$

- Training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$ are i.i.d.

LAMDA Learning And Mining from DatA

What is Stochastic Optimization? (II)

Definition 2: A Special Access Model [Hazan and Kale, 2011]

$$\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$$

- There exists a stochastic oracle that produces unbiased gradient $\mathbf{o}(\cdot)$

$$\mathrm{E}[\mathbf{o}(\mathbf{w})] \in \partial f(\mathbf{w}) \text{ or } \mathrm{E}[\mathbf{o}(\mathbf{w})] = \nabla f(\mathbf{w})$$

Empirical Risk Minimization

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

- Sampling a $(\mathbf{x}_t, y_t)$ from $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ randomly

$$\mathrm{E}[\partial \ell(y_t, \mathbf{x}_t^\top \mathbf{w})] \subseteq \partial \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w})$$

LAMDA Learning And Mining from DatA

What is Stochastic Optimization? (II)

Definition 2: A Special Access Model [Hazan and Kale, 2011]

$$\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$$

- There exists a stochastic oracle that produces unbiased gradient $\mathbf{o}(\cdot)$

$$\mathrm{E}[\mathbf{o}(\mathbf{w})] \in \partial f(\mathbf{w}) \text{ or } \mathrm{E}[\mathbf{o}(\mathbf{w})] = \nabla f(\mathbf{w})$$

Empirical Risk Minimization

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

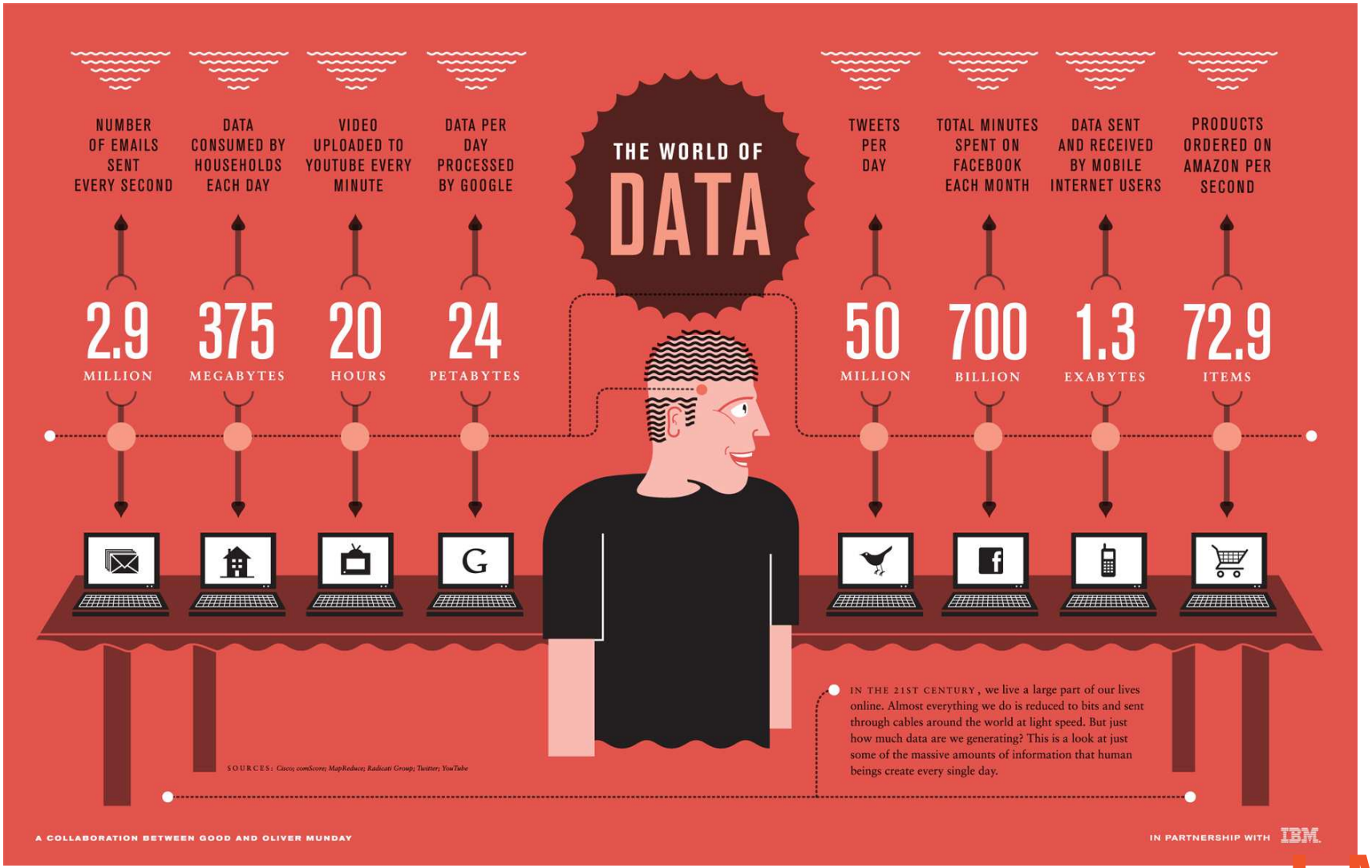
- Sampling a $(\mathbf{x}_t, y_t)$ from $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ randomly

$$\mathrm{E}[\partial \ell(y_t, \mathbf{x}_t^\top \mathbf{w})] \subseteq \partial \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, \mathbf{x}_i^\top \mathbf{w})$$

LAMDA Learning And Mining from DatA

Outline

Why Stochastic Optimization? (I)



https://infographiclist.files.wordpress.com/2011/09/world-of-data.jpeg

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (II)

Empirical Risk Minimization

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

- n is the number of training data
- d is the dimensionality

Deterministic Optimization—Gradient Descent (GD)

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

The Challenge

- Time complexity per iteration: $O(nd) + O(poly(d))$

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (II)

Empirical Risk Minimization

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

- n is the number of training data
- d is the dimensionality

Deterministic Optimization—Gradient Descent (GD)

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

The Challenge

- Time complexity per iteration: $O(nd) + O(poly(d))$

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (III)

Empirical Risk Minimization

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

Stochastic Optimization—Stochastic Gradient Descent (SGD)

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

The Advantage—*Time Reduction*

- Time complexity per iteration: $O(d) + O(poly(d))$

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (III)

Empirical Risk Minimization

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

Stochastic Optimization—Stochastic Gradient Descent (SGD)

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

The Advantage—*Time Reduction*

- Time complexity per iteration: $O(d) + O(poly(d))$

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (IV)

Optimization over Large Matrices

$$\min_{W \in \mathcal{W} \subseteq \mathbb{R}^{m \times n}} F(W)$$

- Matrix completion, distance metric learning, multi-task learning, multi-class learning

Deterministic Optimization—Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $W'_{t+1} = W_t - \eta_t \nabla F(W_t)$
3: $W_{t+1} = \Pi_{\mathcal{W}}(W'_{t+1})$
4: **end for**

The Challenge

- Space complexity per iteration: $O(mn)$

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (IV)

Optimization over Large Matrices

$$\min_{W \in \mathcal{W} \subseteq \mathbb{R}^{m \times n}} F(W)$$

- Matrix completion, distance metric learning, multi-task learning, multi-class learning

Deterministic Optimization—Gradient Descent (GD)

1: **for** $t = 1, 2, \ldots, T$ **do**
2: $\quad W'_{t+1} = W_t - \eta_t \nabla F(W_t)$
3: $\quad W_{t+1} = \Pi_{\mathcal{W}}(W'_{t+1})$
4: **end for**

The Challenge

- Space complexity per iteration: $O(mn)$

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (V)

Optimization over Large Matrices

$$\min_{W \in \mathcal{W} \subseteq \mathbb{R}^{m \times n}} F(W)$$

Stochastic Optimization—Stochastic Gradient Descent (SGD)

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: $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
4: $W_{t+1} = \Pi_{\mathcal{W}}(W'_{t+1})$
5: **end for**

The Advantage—*Space Reduction*

- Space complexity per iteration could be: $O((m+n)r)$

LAMDA Learning And Mining from DatA

Why Stochastic Optimization? (V)

Optimization over Large Matrices

$$\min_{W \in \mathcal{W} \subseteq \mathbb{R}^{m \times n}} F(W)$$

Stochastic Optimization—Stochastic Gradient Descent (SGD)

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: $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
4: $W_{t+1} = \Pi_{\mathcal{W}}(W'_{t+1})$
5: **end for**

The Advantage—*Space Reduction*

- Space complexity per iteration could be: $O((m+n)r)$

LAMDA Learning And Mining from DatA

Outline

LAMDA
Learning And Mining from DatA

Stochastic Gradient Descent (SGD)

Empirical Risk Minimization

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

The Algorithm

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

Advantage v.s. Disadvantage

- Time complexity per iteration is low: $O(d) + O(poly(d))$
- The iteration complexity is much higher than GD

LAMDA Learning And Mining from DatA

Stochastic Gradient Descent (SGD)

Empirical Risk Minimization

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

The Algorithm

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

Advantage v.s. Disadvantage

- Time complexity per iteration is low: $O(d) + O(poly(d))$
- The iteration complexity is much higher than GD

LAMDA Learning And Mining from DatA

The Problem

Iteration Complexity

The number of iterations T to ensure

$$f(\mathbf{w}_T) - \min_{\mathbf{w}\in\Omega} f(\mathbf{w}) \leq \epsilon$$

Iteration Complexity of GD and SGD

	Convex & Smooth	Strongly Convex & Smooth
GD	$O\left(\frac{1}{\sqrt{\epsilon}}\right)$	$O\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\mu\epsilon}\right)$

Total Time Complexity of GD and SGD

	Convex & Smooth	Strongly Convex & Smooth
GD	$O\left(\frac{n}{\sqrt{\epsilon}}\right)$	$O\left(n\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\mu\epsilon}\right)$

LAMDA Learning And Mining from DatA

The Problem

Iteration Complexity

The number of iterations T to ensure

$$f(\mathbf{w}_T) - \min_{\mathbf{w}\in\Omega} f(\mathbf{w}) \leq \epsilon$$

Iteration Complexity of GD and SGD

	Convex & Smooth	Strongly Convex & Smooth
GD	$O\left(\frac{1}{\sqrt{\epsilon}}\right)$	$O\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\mu\epsilon}\right)$

Total Time Complexity of GD and SGD

	Convex & Smooth	Strongly Convex & Smooth
GD	$O\left(\frac{n}{\sqrt{\epsilon}}\right)$	$O\left(n\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\mu\epsilon}\right)$

LAMDA Learning And Mining from DatA

The Problem

Iteration Complexity

The number of iterations T to ensure

$$f(\mathbf{w}_T) - \min_{\mathbf{w}\in\Omega} f(\mathbf{w}) \leq \epsilon$$

Iteration Complexity of GD and SGD $\epsilon = 10^{-6}$

	Convex & Smooth		Strongly Convex & Smooth	
GD	$O\left(\frac{1}{\sqrt{\epsilon}}\right)$	10^3	$O\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$	$6\sqrt{\kappa}$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	10^{12}	$O\left(\frac{1}{\mu\epsilon}\right)$	$\frac{10^6}{\mu}$

Total Time Complexity of GD and SGD

	Convex & Smooth	Strongly Convex & Smooth
GD	$O\left(\frac{n}{\sqrt{\epsilon}}\right)$	$O\left(n\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\mu\epsilon}\right)$

LAMDA Learning And Mining from DatA

The Problem

Iteration Complexity

The number of iterations T to ensure

$$f(\mathbf{w}_T) - \min_{\mathbf{w}\in\Omega} f(\mathbf{w}) \leq \epsilon$$

Iteration Complexity of GD and SGD $\epsilon = 10^{-6}$

	Convex & Smooth		Strongly Convex & Smooth	
GD	$O\left(\frac{1}{\sqrt{\epsilon}}\right)$	10^3	$O\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$	$6\sqrt{\kappa}$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	10^{12}	$O\left(\frac{1}{\mu\epsilon}\right)$	$\frac{10^6}{\mu}$

Total Time Complexity of GD and SGD

	Convex & Smooth	Strongly Convex & Smooth
GD	$O\left(\frac{n}{\sqrt{\epsilon}}\right)$	$O\left(n\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\mu\epsilon}\right)$

LAMDA Learning And Mining from DatA

The Problem

Iteration Complexity

The number of iterations T to ensure

$$f(\mathbf{w}_T) - \min_{\mathbf{w}\in\Omega} f(\mathbf{w}) \leq \epsilon$$

Iteration Complexity of GD and SGD $\epsilon = 10^{-6}$

	Convex & Smooth		Strongly Convex & Smooth	
GD	$O\left(\frac{1}{\sqrt{\epsilon}}\right)$	10^3	$O\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$	$6\sqrt{\kappa}$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	10^{12}	$O\left(\frac{1}{\mu\epsilon}\right)$	$\frac{10^6}{\mu}$

Total Time Complexity of GD and SGD $\epsilon = 10^{-6}$

	Convex & Smooth		Strongly Convex & Smooth	
GD	$O\left(\frac{n}{\sqrt{\epsilon}}\right)$	$10^3 n$	$O\left(n\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$	$6n\sqrt{\kappa}$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	10^{12}	$O\left(\frac{1}{\mu\epsilon}\right)$	$\frac{10^6}{\mu}$

LAMDA Learning And Mining from DatA

The Problem

Iteration Complexity

The number of iterations T to ensure

$$f(\mathbf{w}_T) - \min_{\mathbf{w}\in\Omega} f(\mathbf{w}) \leq \epsilon$$

Iteration Complexity of GD and SGD $\epsilon = 10^{-6}$

	Convex & Smooth		Strongly Convex & Smooth	
GD	$O\left(\frac{1}{\epsilon}\right)$	10^6	$O\left(\log\frac{1}{\epsilon}\right)$	6
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	10^{12}	$O\left(\frac{1}{\epsilon}\right)$	10^6

SGD may be slower than GD if ϵ is very small.

Total Time Complexity of GD and SGD $\epsilon = 10^{-6}$

	Convex & Smooth		Strongly Convex & Smooth	
GD	$O\left(\frac{n}{\sqrt{\epsilon}}\right)$	$10^3 n$	$O\left(n\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$	$6n\sqrt{\kappa}$
SGD	$O\left(\frac{1}{\epsilon^2}\right)$	10^{12}	$O\left(\frac{1}{\mu\epsilon}\right)$	$\frac{10^6}{\mu}$

LAMDA Learning And Mining from DatA

Outline

LAMDA
Learning And Mining from DatA

Motivations

Reason of Slow Convergence Rate

The step size of SGD is a decreasing sequence

- $\eta_t = \frac{1}{\sqrt{t}}$ for convex function
- $\eta_t = \frac{1}{t}$ for strongly convex function

Reason of Decreasing Step Size

$$\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t)$$

Stochastic gradients introduce a constant error

The Key Idea

- Control the variance of stochastic gradients
- Choose a fixed step size η

LAMDA Learning And Mining from DatA

Motivations

Reason of Slow Convergence Rate

The step size of SGD is a decreasing sequence

- $\eta_t = \frac{1}{\sqrt{t}}$ for convex function
- $\eta_t = \frac{1}{t}$ for strongly convex function

Reason of Decreasing Step Size

$$\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t)$$

Stochastic gradients introduce a constant error

The Key Idea

- Control the variance of stochastic gradients
- Choose a fixed step size η

LAMDA Learning And Mining from DatA

Motivations

Reason of Slow Convergence Rate

The step size of SGD is a decreasing sequence

- $\eta_t = \frac{1}{\sqrt{t}}$ for convex function
- $\eta_t = \frac{1}{t}$ for strongly convex function

Reason of Decreasing Step Size

$$\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t)$$

Stochastic gradients introduce a constant error

The Key Idea

- Control the variance of stochastic gradients
- Choose a fixed step size η

LAMDA Learning And Mining from DatA

Mixed Gradient Descent (I)

Mixed Gradient of $\mathbf{w}_t$

$$\mathbf{m}(\mathbf{w}_t) = \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_0) + \nabla f(\mathbf{w}_0)$$

where $(\mathbf{x}_t, y_t)$ is a random sample, $\mathbf{w}_0$ is a initial solution, and

$$\nabla f(\mathbf{w}_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_0)$$

The Properties of Mixed Gradient

- It is still a unbiased estimate of true gradient

$$\mathrm{E}[\mathbf{m}(\mathbf{w}_t)] = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) = \nabla f(\mathbf{w}_t)$$

- The variance is controlled by the distance

$$\|\nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_0)\|_2 \leq L \|\mathbf{w}_t - \mathbf{w}_0\|_2$$

LAMDA Learning And Mining from DatA

Mixed Gradient Descent (I)

Mixed Gradient of $\mathbf{w}_t$

$$\mathbf{m}(\mathbf{w}_t) = \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_0) + \nabla f(\mathbf{w}_0)$$

where $(\mathbf{x}_t, y_t)$ is a random sample, $\mathbf{w}_0$ is a initial solution, and

$$\nabla f(\mathbf{w}_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_0)$$

The Properties of Mixed Gradient

- It is still a unbiased estimate of true gradient

$$\mathrm{E}[\mathbf{m}(\mathbf{w}_t)] = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) = \nabla f(\mathbf{w}_t)$$

- The variance is controlled by the distance

$$\|\nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_0)\|_2 \leq L \|\mathbf{w}_t - \mathbf{w}_0\|_2$$

LAMDA Learning And Mining from DatA

Mixed Gradient Descent (I)

Mixed Gradient of $\mathbf{w}_t$

$$\mathbf{m}(\mathbf{w}_t) = \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_0) + \nabla f(\mathbf{w}_0)$$

where $(\mathbf{x}_t, y_t)$ is a random sample, $\mathbf{w}_0$ is a initial solution, and

$$\nabla f(\mathbf{w}_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_0)$$

The Properties of Mixed Gradient

- It is still a unbiased estimate of true gradient

$$\mathrm{E}[\mathbf{m}(\mathbf{w}_t)] = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) = \nabla f(\mathbf{w}_t)$$

- The variance is controlled by the distance

$$\|\nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_0)\|_2 \leq L \|\mathbf{w}_t - \mathbf{w}_0\|_2$$

LAMDA Learning And Mining from DatA

Mixed Gradient Descent (I)

Mixed Gradient of $\mathbf{w}_t$

$$\mathbf{m}(\mathbf{w}_t) = \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_0) + \nabla f(\mathbf{w}_0)$$

where $(\mathbf{x}_t, y_t)$ is a random sample, $\mathbf{w}_0$ is a initial solution, and

$$\nabla f(\mathbf{w}_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_0)$$

The Properties of Mixed Gradient

- It is still a unbiased estimate of true gradient

$$\mathrm{E}[\mathbf{m}(\mathbf{w}_t)] = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) = \nabla f(\mathbf{w}_t)$$

- The variance is controlled by the distance

$$\|\nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_0)\|_2 \leq L \|\mathbf{w}_t - \mathbf{w}_0\|_2$$

LAMDA Learning And Mining from DatA

Mixed Gradient Descent (I)

Mixed Gradient of $\mathbf{w}_t$

$$\mathbf{m}(\mathbf{w}_t) = \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t{}^\top \mathbf{w}_0) + \nabla f(\mathbf{w}_0)$$

where $(\mathbf{x}_t, y_t)$ is a random sample, $\mathbf{w}_0$ is a initial solution, and

$$\nabla f(\mathbf{w}_0) = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_0)$$

The Properties of Mixed Gradient

- It is still a unbiased estimate of true gradient

$$\mathrm{E}[\mathbf{m}(\mathbf{w}_t)] = \frac{1}{n} \sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_t) = \nabla f(\mathbf{w}_t)$$

- The variance is controlled by the distance

$$\|\nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_0)\|_2 \leq L \|\mathbf{w}_t - \mathbf{w}_0\|_2$$

LAMDA Learning And Mining from DatA

Mixed Gradient Descent (II)

The Algorithm [Zhang et al., 2013]

1: Compute the true gradient of $\mathbf{w}_0$

$$\nabla f(\mathbf{w}_0) = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell(y_i, \mathbf{x}_i^\top \mathbf{w}_0)$$

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

3: Sample a training example $(\mathbf{x}_t, y_t)$ randomly

4: Compute the mixed gradient of $\mathbf{w}_t$

$$\mathbf{m}(\mathbf{w}_t) = \nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_t) - \nabla \ell(y_t, \mathbf{x}_t^\top \mathbf{w}_0) + \nabla f(\mathbf{w}_0)$$

5: $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta \mathbf{m}(\mathbf{w}_t)$

6: $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}'_{t+1})$

7: **end for**

LAMDA Learning And Mining from DatA

Theoretical Guarantees

Theorem 1 ([Zhang et al., 2013])

Suppose the objective function is smooth and strongly convex. To find an ϵ-optimal solution, the mixed gradient descent needs

	True Gradient	*Stochastic Gradient*
MGD	$O\left(\log \frac{1}{\epsilon}\right)$	$O\left(\kappa^2 \log \frac{1}{\epsilon}\right)$

In contrast, SGD needs $O(1/\mu\epsilon)$ stochastic gradients.

Extensions

- For unbounded domain, $O(\kappa^2 \log 1/\epsilon))$ can be improved to $O(\kappa \log 1/\epsilon)$ [Johnson and Zhang, 2013]
- For smooth and convex function, $O(\log 1/\epsilon)$ true gradients and $O(1/\epsilon)$ stochastic gradients are needed [Mahdavi et al., 2013]

LAMDA Learning And Mining from DatA

Theoretical Guarantees

Theorem 1 ([Zhang et al., 2013])

Suppose the objective function is smooth and strongly convex. To find an ϵ-optimal solution, the mixed gradient descent needs

	True Gradient	*Stochastic Gradient*
MGD	$O\left(\log \frac{1}{\epsilon}\right)$	$O\left(\kappa^2 \log \frac{1}{\epsilon}\right)$

In contrast, SGD needs $O(1/\mu\epsilon)$ stochastic gradients.

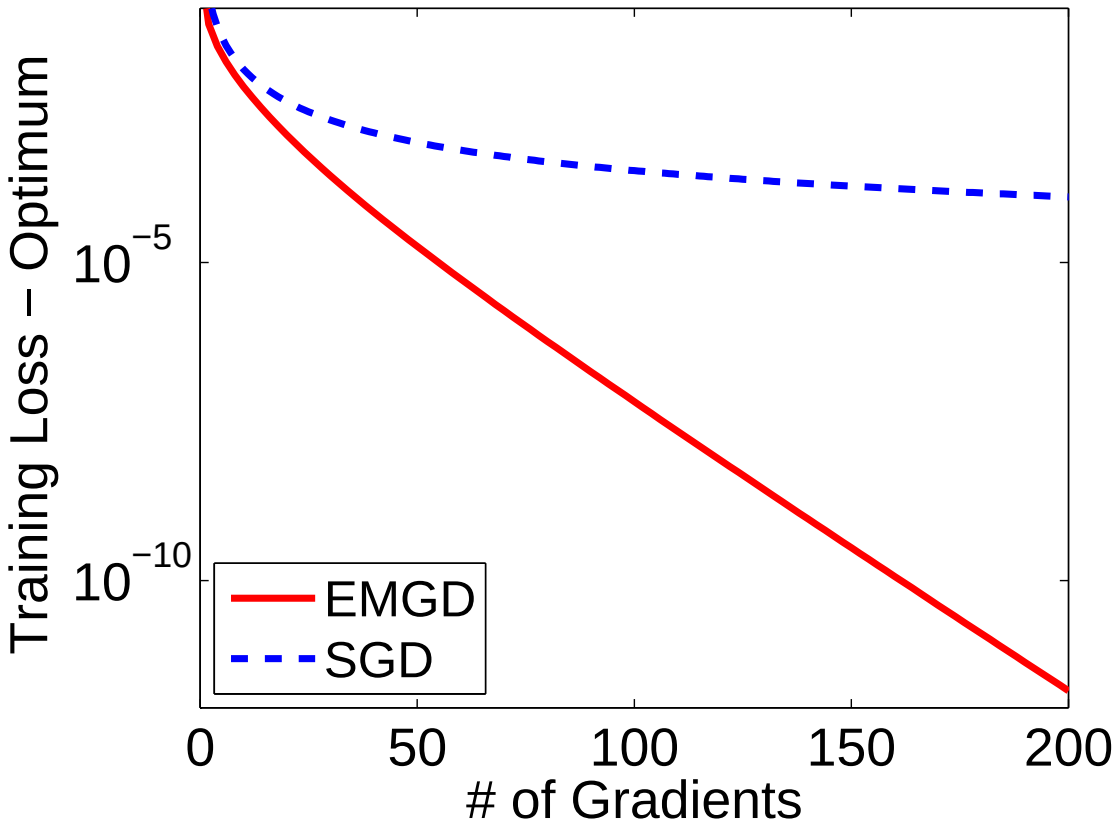
Extensions

- For unbounded domain, $O(\kappa^2 \log 1/\epsilon))$ can be improved to $O(\kappa \log 1/\epsilon)$ [Johnson and Zhang, 2013]
- For smooth and convex function, $O(\log 1/\epsilon)$ true gradients and $O(1/\epsilon)$ stochastic gradients are needed [Mahdavi et al., 2013]

LAMDA Learning And Mining from DatA

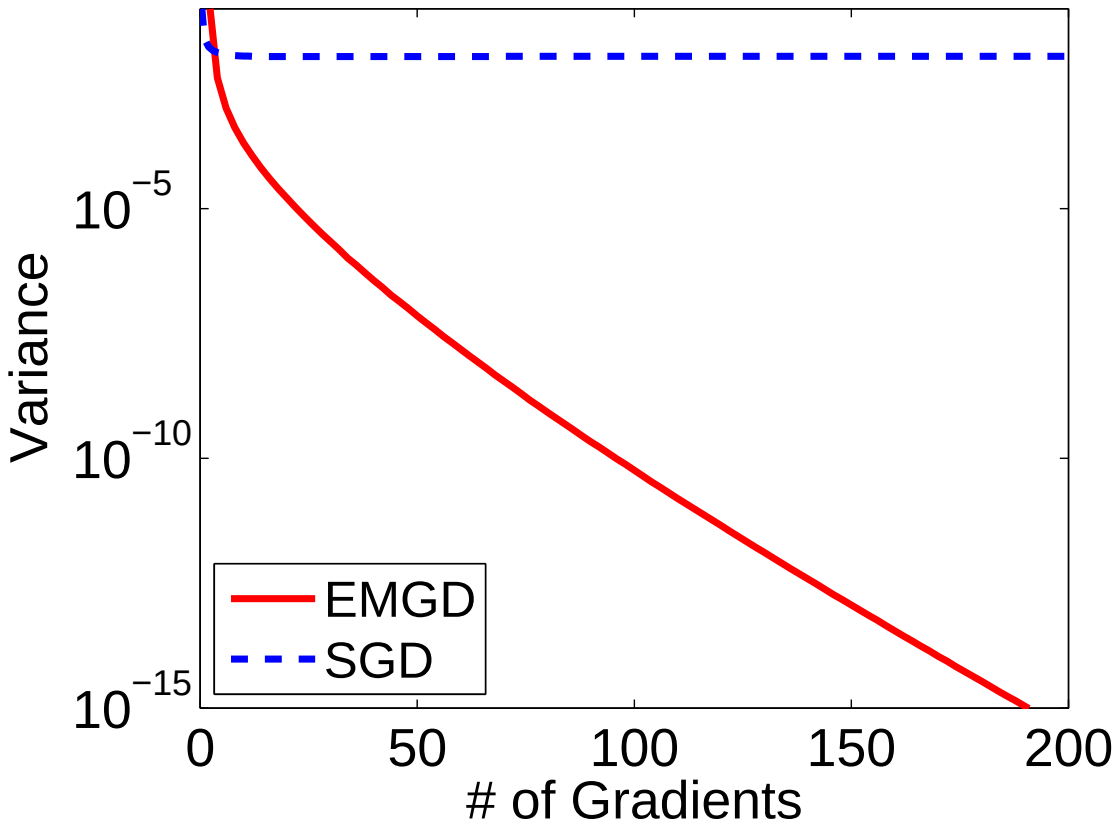
Experimental Results (I)

- Reuters Corpus Volume I (RCV1) data set
- The optimization error



Experimental Results (II)

- Reuters Corpus Volume I (RCV1) data set
- The variance of mixed gradient



Outline

LAMDA
Learning And Mining from DatA

Stochastic Gradient Descent (SGD)

Nuclear Norm Regularization (Composition Optimization)

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

- The optimal solution W_* is low-rank

The Algorithm [Avron et al., 2012]

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: $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
4: $W_{t+1} = \Pi_r(W'_{t+1})$ (truncate small singular values)
5: **end for**

Efficient Implementation

- $W_{t+1} = \Pi_r(W_t - \eta_t \widehat{G}_t)$ 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 Gradient Descent (SGD)

Nuclear Norm Regularization (Composition Optimization)

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

- The optimal solution W_* is low-rank

The Algorithm [Avron et al., 2012]

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: $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
4: $W_{t+1} = \Pi_r(W'_{t+1})$ (truncate small singular values)
5: **end for**

Efficient Implementation

- $W_{t+1} = \Pi_r(W_t - \eta_t \widehat{G}_t)$ 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 Gradient Descent (SGD)

Nuclear Norm Regularization (Composition Optimization)

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

- The optimal solution W_* is low-rank

The Algorithm [Avron et al., 2012]

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: $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
4: $W_{t+1} = \Pi_r(W'_{t+1})$ (truncate small singular values)
5: **end for**

Efficient Implementation

- $W_{t+1} = \Pi_r(W_t - \eta_t \widehat{G}_t)$ 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 Gradient Descent (SGD)

Nuclear Norm Regularization (Composition Optimization)

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

- The optimal solution W_* is low-rank

The Algorithm [Avron et al., 2012]

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: $W'_{t+1} = W_t - \eta_t \widehat{G}_t$
4: $W_{t+1} = \Pi_r(W'_{t+1})$ (truncate small singular values)
5: **end for**

Advantage v.s. Disadvantage

- Space complexity per iteration is low: $O((m+n)r)$
- There is no theoretical guarantee due to truncation

LAMDA Learning And Mining from DatA

Outline

LAMDA
Learning And Mining from DatA

Stochastic Proximal Gradient Descent (SPGD)

Nuclear Norm Regularization

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

The Algorithm [Zhang et al., 2015b]

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 also 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 Regularization

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

The Algorithm [Zhang et al., 2015b]

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 also 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 Regularization

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

The Algorithm [Zhang et al., 2015b]

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 also 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 Regularization

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

The Algorithm [Zhang et al., 2015b]

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}

Advantages

- Space complexity per iteration is still low: $O((m+n)r)$
- It is supported by solid theoretical guarantees

LAMDA Learning And Mining from DatA

Theoretical Guarantees

Theorem 2 (General convex functions [Zhang et al., 2015b])

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 3 (Strongly convex functions [Zhang et al., 2015b])

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

Theoretical Guarantees

Theorem 2 (General convex functions [Zhang et al., 2015b])

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 3 (Strongly convex functions [Zhang et al., 2015b])

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

Application to 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

The Challenge

- Space complexity is $O(n^2)$

The Question

- Is it possible to find top eigensystems of K without constructing K explicitly?
- Yes, by SPGD.

LAMDA Learning And Mining from DatA

Application to 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

The Challenge

- Space complexity is $O(n^2)$

The Question

- Is it possible to find top eigensystems of K without constructing K explicitly?
- Yes, by SPGD.

LAMDA
Learning And Mining from DatA

Application to 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

The Challenge

- Space complexity is $O(n^2)$

The Question

- Is it possible to find top eigensystems of K without constructing K explicitly?
- Yes, by SPGD.

LAMDA
Learning And Mining from DatA

Application to Kernel PCA (I)

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 can be recovered from the optimal solution W_*

Proximal Gradient Descent

1: **for** $t = 1, 2, \ldots, T$ **do**
2: Calculate the gradient $G_t = W_t - K$
3:

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

4: **end for**

LAMDA Learning And Mining from DatA

Application to Kernel PCA (I)

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 can be recovered from the optimal solution W_*

Proximal Gradient Descent

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

2: Calculate the gradient $G_t = W_t - K$

3:

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

4: **end for**

LAMDA Learning And Mining from DatA

Application to Kernel PCA (II)

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., 2015a]

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

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

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**

ξ is low-rank and $\mathbb{E}[\xi] = K$

- Random Fourier features [Rahimi and Recht, 2008]
- Constructing columns/rows of K randomly

LAMDA Learning And Mining from DatA

Application to Kernel PCA (II)

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., 2015a]

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

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

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**

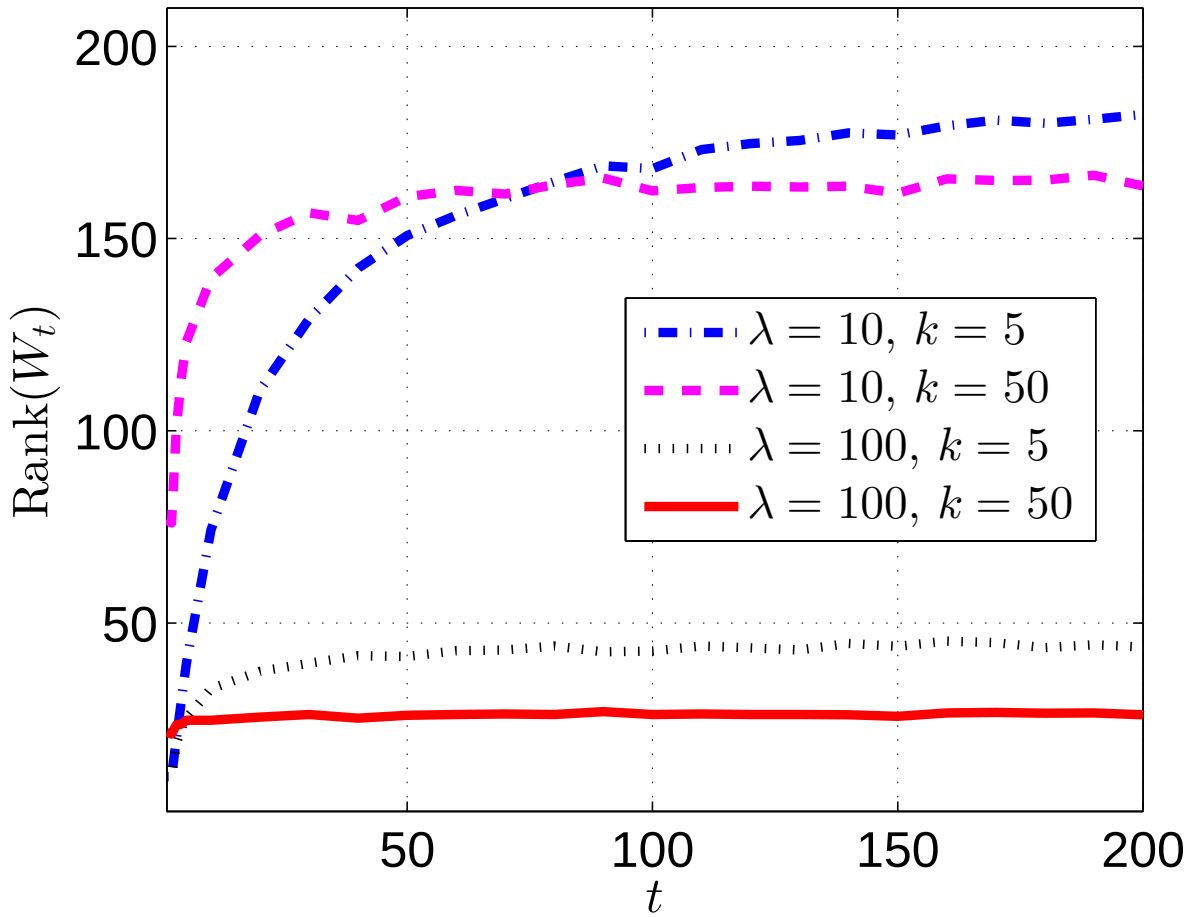
ξ is low-rank and $\mathbb{E}[\xi] = K$

- Random Fourier features [Rahimi and Recht, 2008]
- Constructing columns/rows of K randomly

LAMDA Learning And Mining from DatA

Experimental Results (I)

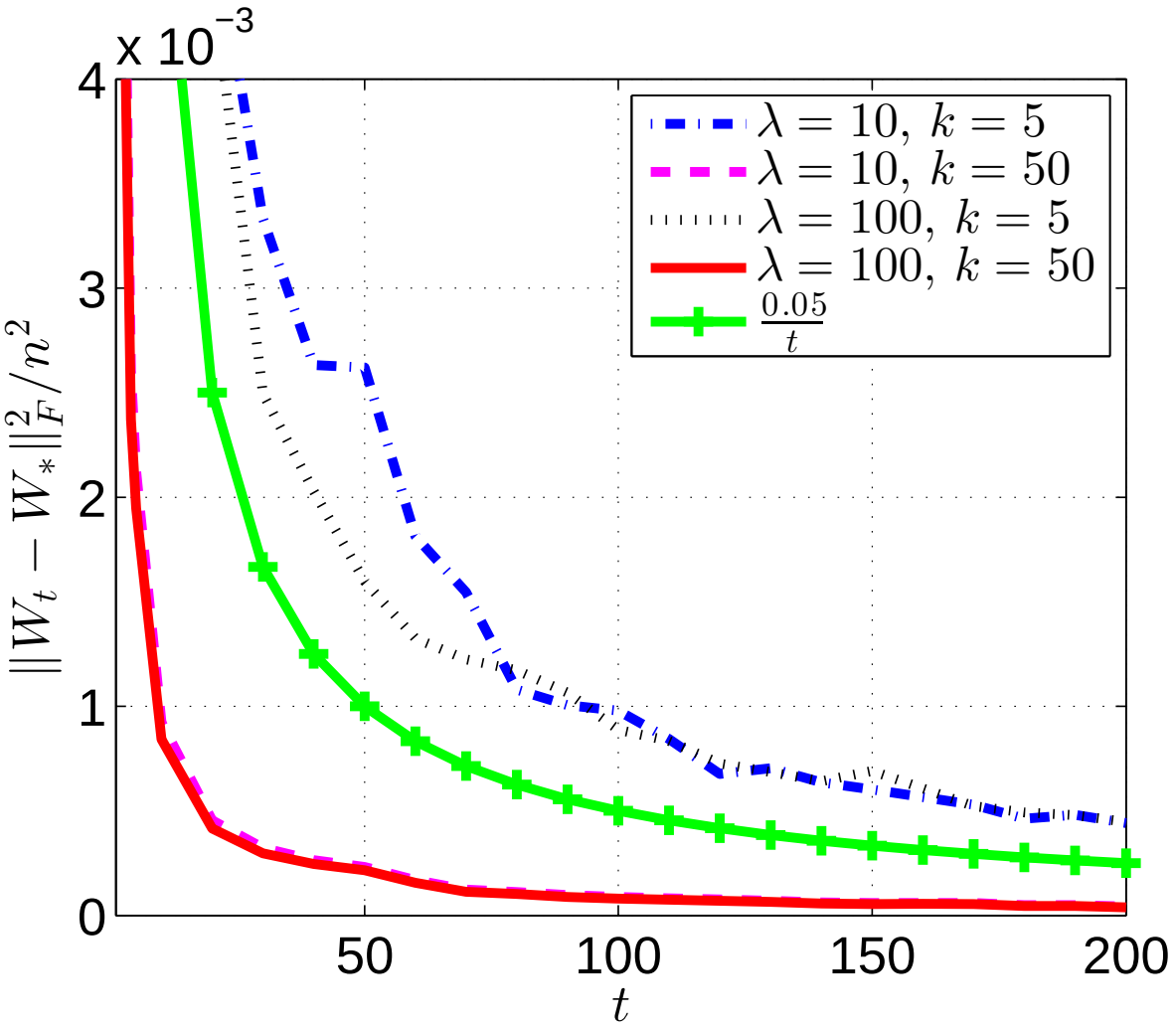
- The Magic data set
- The rank of each iterate



LAMDA Learning And Mining from DatA

Experimental Results (II)

- The Magic data set
- The optimization error



LAMDA Learning And Mining from DatA

Conclusion and Future Work

Time Reduction by Stochastic Optimization

- A novel algorithm named Mixed Gradient Descent (MGD) is proposed to reduce the iteration complexity
- Extend MGD to other scenarios, such as distributed setting, non-convex functions

Space Reduction by Stochastic Optimization

- A simple algorithm based on Stochastic Proximal Gradient Descent (SPGD) is proposed to reduce the space complexity
- Apply to more applications, such as matrix completion, multi-task learning

LAMDA
Learning And Mining from DatA

Special Issue on "Multi-instance Learning in Pattern Recognition and Vision"

URL: `http://www.journals.elsevier.com/pattern-recognition/call-for-papers/special-issue-on-multiple-instance-learning-in-pattern-recog`

Dates:

- First submission date: Mar. 1, 2016
- Final paper notification: Dec. 1, 2016

Guest editors:

- Jianxin Wu, Nanjing University
- Xiang Bai, Huazhong University of Science and Technology
- Marco Loog, Delft University of Technology
- Fabio Roli, University of Cagliari
- Zhi-Hua Zhou, Nanjing University

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.

Hazan, E. and Kale, S. (2011).
Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization.
In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 421–436.

Johnson, R. and Zhang, T. (2013).
Accelerating stochastic gradient descent using predictive variance reduction.
In *Advances in Neural Information Processing Systems 26*, pages 315–323.

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.

LAMDA
Learning And Mining from DatA

Reference II


Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009).
Robust stochastic approximation approach to stochastic programming.
SIAM Journal on Optimization, 19(4):1574–1609.


Rahimi, A. and Recht, B. (2008).
Random features for large-scale kernel machines.
In *Advances in Neural Information Processing Systems 20*, pages 1177–1184.


Zhang, L., Mahdavi, M., and Jin, R. (2013).
Linear convergence with condition number independent access of full gradients.
In *Advance in Neural Information Processing Systems 26*, pages 980–988.


Zhang, L., Yang, T., Jin, R., and Zhou, Z.-H. (2015a).
Stochastic optimization for kernel pca.
Submitted.


Zhang, L., Yang, T., Jin, R., and Zhou, Z.-H. (2015b).
Stochastic proximal gradient descent for nuclear norm regularization.
ArXiv e-prints, arXiv:1511.01664.

LAMDA
Learning And Mining from DatA