#### Randomized Algorithms in Machine Learning

Tianbao Yang

#### Department of Computer Science The University of Iowa

AMCS Seminar

April 24, 2015

イロト イボト イヨト イヨ

#### Outline

- Machine Learning
   Introduction
- 2 Randomized Algorithms (RA)
- Our Recent work on RA for Big Data Optimization
  - Randomized Reduction and Recovery
  - Dual-sparse Randomized Reduction and Recovery
  - Results

#### Take-home Messages

イロト イボト イヨト イヨ

590

< □ > < □ > < □ > < □ > < □ >

#### Machine Learning

## What is Machine Learning?

#### Arthur Samule (1959)

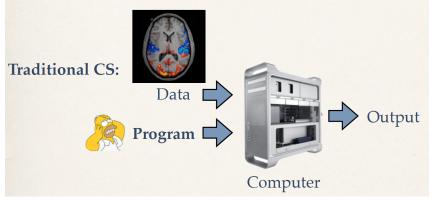
"Field of study that gives computers the ability to learn without being explicitly programmed"

Yang (CS@Uiowa)

< □ > < 同 > < 三 > <

#### Machine Learning

#### Traditional Computer Science



picture by courtesy of Killian Weinberger.

Yang (C	S@Uiowa)
---------	----------

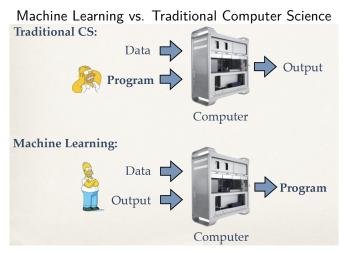
AMCS seminar

April 24, 2015 5 / 68

э

イロト イボト イヨト イヨ

#### Machine Learning



picture by courtesy of Killian Weinberger.

Yang (CS@Uiowa)

AMCS seminar

April 24, 2015 6 / 68

э

#### Machine Learning



#### Let the Data Speak for itself!

picture by courtesy of Killian Weinberger.

Yang (CS@	Jiowa)	a)
-----------	--------	----

AMCS seminar

April 24, 2015 7 / 68

э

イロト イボト イヨト イヨ

#### Applications of Machine Learning

#### Spam Filter

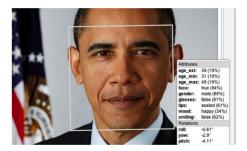
Google	e in:spam -		Q
Gmail -	-	C More -	
COMPOSE			Delete all spam messages now (messages that have been in Spa
COMPOSE		me	New submission from Quick Poll: Facebook Pre-Fill - I would u
Inbox (7)		no1.gr	Προστατέψτε το κινητό σας Ean den mporeite na deite to ne
Starred		PayPal	Your PayPal account has been limited! - Warning Notification De
Important Sent Mail	口☆●	EdFed	"What NOT TO DO During Your Interview" - To ensure prompt di
Drafts (15)		LoopGalaxy	March Madness Sale! 50% Off All Sample Packs - Share Ember
All Mail		LinkShare	Register Now: Social & Mobile Technologies Webinar - Social
3pam (46)		WESTERN UNION MONEY TR	WESTERN UNION - Attn, We are grateful to contact you and anno
Trash		Miss Beauty Musa	Dearest - Dearest I know this mail will come to you as a surprise s
Circles		American Musical Supply	Live Loud on Stage with Pro Gear up to 66% off - Speaker Syst



イロト イボト イヨト イヨ

### Applications of Machine Learning

## Face Recognition



э

<ロト <回ト < 回ト < 回ト

## Applications of Machine Learning

#### Speech Recognition



Yang (CS@Uiowa)

3

イロト イボト イヨト イヨト

## Applications of Machine Learning





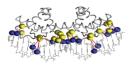








Image: A match a ma

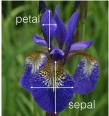
April 24, 2015 11 / 68

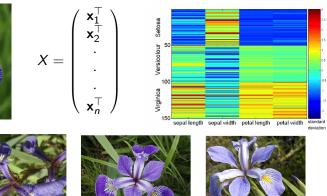
★ ∃ → ∃

DQC

## Data Matrices and Machine Learning

#### The Instance-feature Matrix: $X \in \mathbb{R}^{n \times d}$





Setosa

Versicolour

Virginica

◆□▶ ◆□▶ ◆三▶ ◆三▶

April 24, 2015 12 / 68

€ 990

The output vector: 
$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ \vdots \\ \vdots \\ y_n \end{pmatrix}$$

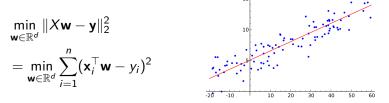
- continuous  $y_i \in \mathbb{R}$ : regression (e.g., house price)
- discrete, e.g.,  $y_i \in \{1, 2, 3\}$ : classification (e.g., species of iris)



イロト イポト イヨト イヨト

Many machine learning tasks are formulated based on the data matrix X and the output vector **y**.

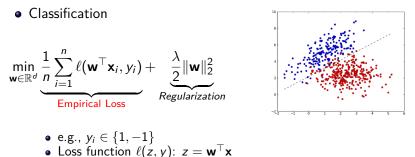
• Regression: (minimize the least-squares error)



- $\mathbf{w} \in \mathbb{R}^d$  refers to the predictive model (or the program as referred at the beginning)
- Prediction on new data:  $\mathbf{x}_{new}^{\top} \mathbf{w}_*$  ( $\mathbf{w}_*$  optimizes the objective function)

## Data Matrices and Machine Learning

Many machine learning tasks are formulated based on the data matrix Xand the output vector **y**.



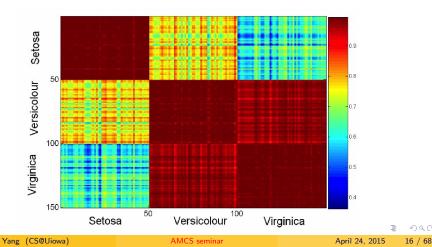
- 1. SVMs: (squared) hinge loss  $\ell(z, y) = \max(0, 1 yz)^p$ , where p = 1, 2
- 2. Logistic Regression:  $\ell(z) = \log(1 + \exp(-yz))$

<ロト <回ト < 回ト < 回ト = ヨト = ヨ

#### Data Matrices and Machine Learning

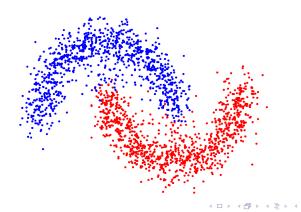
The Instance-Instance Matrix:  $K \in \mathbb{R}^{n \times n}$ 

- Similarity Matrix
- Kernel Matrix



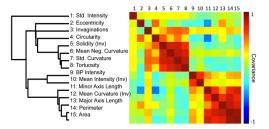
Some machine learning tasks are formulated on the kernel matrix

- Clustering
- Kernel Methods



The Feature-Feature Matrix:  $C \in \mathbb{R}^{d \times d}$ 

- Covariance Matrix
- Distance Metric Matrix



3

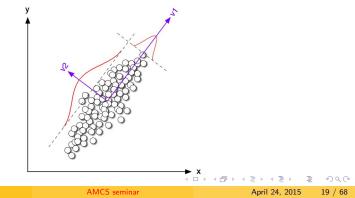
イロト イボト イヨト イヨト

Some machine learning tasks requires the covariance matrix

- Principal Component Analysis
- Dimensionality Reduction

Yang (CS@Uiowa)

• Top-k Singular Value (Eigen-Value) Decomposition of the Covariance Matrix



Huge amount of data generated every day

- Facebook users upload 3 million photos
- Goolge receives 3 billion queries
- Youtube users upload over 1,700 hours video
- Global internet population is 2.1 billion people
- 247 billion emails sent

http://www.visualnews.com/2012/06/19/how-much-data-created-every-minute/



## Do we really need Big Data?

April 24, 2015 20 / 68

Huge amount of data generated every day

- Facebook users upload 3 million photos
- Goolge receives 3 billion queries
- Youtube users upload over 1,700 hours video
- Global internet population is 2.1 billion people
- 247 billion emails sent

http://www.visualnews.com/2012/06/19/how-much-data-created-every-minute/

## Do we really need Big Data?



#### General Visual Recognition Challenge (ImageNet Challenge)



Hundreds of Thousands of Objects

3

イロト イポト イヨト イヨト

#### Fine-grained Image Classification



(a) Siberian husky



(b) Eskimo dog



April 24, 2015 22 / 68

▲□▶ ▲□▶ ▲ 三▶ ▲ 三▶ - 三 - のへで

## **Big Data Challenge**

## Big Data will be the key to achieve success

Example: 1000 Objects Classification

- 14 millions of images indexed
- surpass human-level performance: top-1 accuracy 78% and top-5 accuracy 95%

イロト 不得下 イヨト イヨト

## Big Data will be the key to achieve success

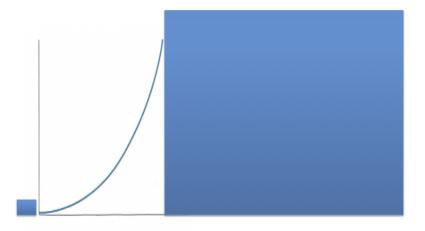
Example: 1000 Objects Classification

- 14 millions of images indexed
- surpass human-level performance: top-1 accuracy 78% and top-5 accuracy 95%

## Why Learning from Big Data is challenging?

イロト イボト イヨト イヨト

#### Why Big Data is challenging



Yang (	CS@Uiowa	l

王

590

イロト イヨト イヨト イヨト

#### Outline

- Machine Learning
   Introduction
- 2 Randomized Algorithms (RA)
- Our Recent work on RA for Big Data Optimization
  - Randomized Reduction and Recovery
  - Dual-sparse Randomized Reduction and Recovery
  - Results

#### Take-home Messages

イロト イボト イヨト イヨト

#### Outline

# Machine Learning Introduction

#### 2 Randomized Algorithms (RA)

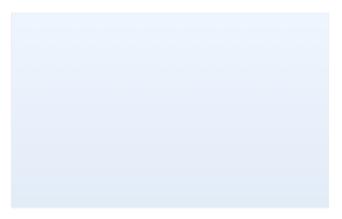
3 Our Recent work on RA for Big Data Optimization

- Randomized Reduction and Recovery
- Dual-sparse Randomized Reduction and Recovery
- Results

#### Take-home Messages

イロト イボト イヨト イヨ

• Use some kind of randomization (sampling) to reduce the cost of computation



3

イロト 不得 トイヨト イヨト

• Use some kind of randomization (sampling) to reduce the cost of computation



イロト イボト イヨト イヨト

э

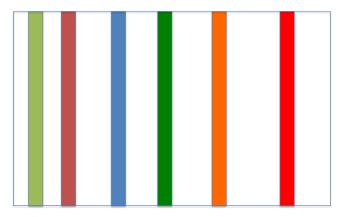
• Use some kind of randomization (sampling) to reduce the cost of computation (e.g., sampling rows or instances)

1	

3

イロト イボト イヨト イヨト

• Use some kind of randomization (sampling) to reduce the cost of computation (e.g., sampling columns or features)



イロト イボト イヨト イヨト

Algorithms:

- Stochastic Optimization (e.g., SGD)
- Randomized Low-rank Matrix Approximation (e.g., randomized SVD)
- Dropout for Deep Learning
- Randomized reduction for regression and classification

Benefits:

- Faster
- More robust (implicit regularization)
- Easy to analyze
- exploit modern computational architectures

3

イロト 不得下 イヨト イヨト

Algorithms:

- Stochastic Optimization (e.g., SGD)
- Randomized Low-rank Matrix Approximation (e.g., randomized SVD)
- Dropout for Deep Learning
- Randomized reduction for regression and classification

Benefits:

- Faster
- More robust (implicit regularization)
- Easy to analyze
- exploit modern computational architectures

イロト イポト イヨト イヨト 二日

Algorithms:

- Stochastic Optimization (e.g., SGD)
- Randomized Low-rank Matrix Approximation (e.g., randomized SVD)
- Dropout for Deep Learning
- Randomized reduction for regression and classification

Benefits:

- Faster
- More robust (implicit regularization)
- Easy to analyze
- exploit modern computational architectures

イロト イポト イヨト イヨト 二日

Randomized Algorithms (RA)

#### Randomized Feature Reduction for Classification

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



- Randomized feature reduction:  $\hat{\mathbf{x}}_i = A\mathbf{x}_i$ , where  $A \in \mathbb{R}^{m \times d}$  with  $m \ll d$
- A: random projection matrix (e.g., Gaussian entries)
- Solve the reduced problem

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

# Why does Randomized Reduction Works?

#### The Johnson-Lindenstrauss Lemma (Johnson & Lindenstrauss (1984)).



projections of the vectors above to random planes (note the planes are translated to the origin)



## Question

#### How can we recover a model in original high-dimensional space?

- Usually features in original feature space have meanings (e.g., genes, words)
- Finding a model in the original feature space can help understand the importance of different features
- Help us design better strategies (e.g., for controlling risk of a disease)

## Question

How can we recover a model in original high-dimensional space?

- Usually features in original feature space have meanings (e.g., genes, words)
- Finding a model in the original feature space can help understand the importance of different features
- Help us design better strategies (e.g., for controlling risk of a disease)

イロト イヨト イヨト

# Outline

- Machine Learning
   Introduction
- 2 Randomized Algorithms (RA)
- Our Recent work on RA for Big Data Optimization
  - Randomized Reduction and Recovery
  - Dual-sparse Randomized Reduction and Recovery
  - Results

### Take-home Messages

# Outline

- Machine Learning
   Introduction
- 2 Randomized Algorithms (RA)
- 3 Our Recent work on RA for Big Data Optimization
  - Randomized Reduction and Recovery
  - Dual-sparse Randomized Reduction and Recovery
  - Results

#### Take-home Messages

## Randomized Feature Reduction for Classification

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

- $\mathbf{x}_i \in \mathbb{R}^d$ , expensive when d is very very large, e.g., millions or billions
- Randomized feature reduction:  $\hat{\mathbf{x}}_i = A\mathbf{x}_i$ , where  $A \in \mathbb{R}^{m \times d}$  with  $m \ll d$
- Solve the reduced problem

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

Question: How to obtain a good model  $\widehat{\boldsymbol{w}}_*$  in the original feature space?

# A Naive Approach

$$\mathbf{u}_{*} = \arg\min_{\mathbf{u}\in\mathbb{R}^{m}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{u}^{\top}\widehat{\mathbf{x}}_{i}, y_{i}) + \frac{\lambda}{2} \|\mathbf{u}\|_{2}^{2}$$
$$\mathbf{u}_{*} = \arg\min_{\mathbf{u}\in\mathbb{R}^{m}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{u}^{\top}A\mathbf{x}_{i}, y_{i}) + \frac{\lambda}{2} \|\mathbf{u}\|_{2}^{2}$$

Naive Recovery:

$$\widehat{\mathbf{w}}_* = A^\top \mathbf{u}_* \in \mathbb{R}^d$$

Problem:  $\widehat{\mathbf{w}}_*$  could be a very bad solution

$$\|\widehat{\mathbf{w}}_* - \mathbf{w}_*\|_2 \geq \Omega\left(\sqrt{rac{d-m}{d}}\|\mathbf{w}_*\|_2
ight)$$

э

# Dual Recovery (COLT'13, IEEE-IT'14)

Yang (CS@Uiowa)

э

< □ > < □ > < □ > < □ > < □ >

## Our Approach: Dual Recovery

The Dual Problem: (using Fenchel conjugate)

$$\ell_i^*(\alpha_i) = \max_{\alpha_i} \alpha_i z - \ell(z, y_i)$$

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

Dual 
$$\alpha_* = \arg \max_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{2\lambda n^2} \alpha^\top X X^\top \alpha$$
  
$$\mathbf{w}_* = -\frac{1}{\lambda n} X^\top \alpha_*$$

э

イロト イボト イヨト イヨト

590

# Our Approach: Dual Recovery

Important Implication from the Dual:  $\mathbf{w}_*$  lies in the row space of the data matrix  $X \in \mathbb{R}^{n imes d}$ 

- the Naive approach:  $\widehat{\mathbf{w}}_* = A^\top \mathbf{u}_*$
- Dual Recovery:  $\widetilde{\mathbf{w}}_* = -\frac{1}{\lambda n} X^\top \widehat{\alpha}_*$ , where

$$\widehat{\alpha}_* = \arg \max_{\alpha \in \mathbb{R}^n} - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{2\lambda n^2} \alpha^\top \widehat{X} \widehat{X}^\top \alpha$$

- $\widehat{X} = XA^{\top} \in \mathbb{R}^{n \times m}$
- Our theorem: under low-rank assumption of the data matrix X (e.g., rank(X) = r), with a high probability  $1 \delta$ ,

$$\|\widetilde{\mathbf{w}}_* - \mathbf{w}_*\|_2 \leq \frac{\epsilon}{1-\epsilon} \|\mathbf{w}_*\|_2, \quad \text{where } \epsilon = \Theta\left(\sqrt{\frac{r\log(r/\delta)}{m}}\right)$$

▲ロト ▲圖 ト ▲ ヨト ▲ ヨト 一 ヨ … のへで

# Our Approach: Dual Recovery

Important Implication from the Dual:  $\mathbf{w}_*$  lies in the row space of the data matrix  $X \in \mathbb{R}^{n imes d}$ 

- the Naive approach:  $\widehat{\mathbf{w}}_* = A^\top \mathbf{u}_*$
- Dual Recovery:  $\widetilde{\mathbf{w}}_* = -\frac{1}{\lambda n} X^\top \widehat{\alpha}_*$ , where

$$\widehat{\alpha}_* = \arg \max_{\alpha \in \mathbb{R}^n} - \frac{1}{n} \sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{2\lambda n^2} \alpha^\top \widehat{X} \widehat{X}^\top \alpha$$

- $\widehat{X} = XA^{\top} \in \mathbb{R}^{n \times m}$
- Our theorem: under low-rank assumption of the data matrix X (e.g., rank(X) = r), with a high probability  $1 \delta$ ,

$$\|\widetilde{\mathbf{w}}_* - \mathbf{w}_*\|_2 \leq \frac{\epsilon}{1-\epsilon} \|\mathbf{w}_*\|_2, \quad \text{ where } \epsilon = \Theta\left(\sqrt{\frac{r\log(r/\delta)}{m}}\right)$$

## Can you remove low-rank assumption?

#### Yes, we can. How?

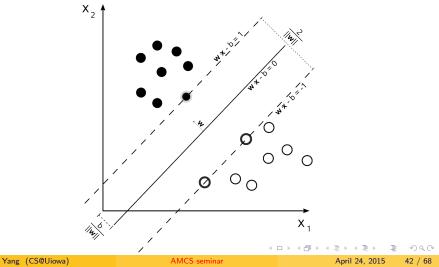
by exploiting the sparsity of the dual variables



## Can you remove low-rank assumption?

Yes, we can. How?

by exploiting the sparsity of the dual variables



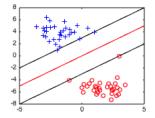
# Dual-sparse Recovery (To appear in ICML'15)

Yang (CS@Uiowa)

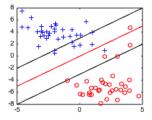
AMCS seminar

April 24, 2015 43 / 68

### Can you remove low-rank assumption?



High-dimensional Space



low-dimensional Space

< □ > < 同 >

590

## Our New Approach: Dual-sparse Recovery

• Dual-sparse Recovery:  $\widetilde{\mathbf{w}}_* = -\frac{1}{\lambda p} X^\top \widehat{\alpha}_*$ , where

$$\widehat{\alpha}_* = \arg \max_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{2\lambda n^2} \alpha^\top \widehat{X} \widehat{X}^\top \alpha - \frac{\tau}{n} \|\alpha\|_1$$

• Our theorem: if  $\alpha_*$  is s-sparse, with a high probability  $1-\delta$ ,

$$\|\widetilde{\mathbf{w}}_* - \mathbf{w}_*\|_2 \le \epsilon \|\mathbf{w}_*\|_2$$
, where  $\epsilon = \Theta\left(\sqrt{\frac{s\log(n/\delta)}{m}}\right)$ 

イロト 不得 トイヨト イヨト

## Our New Approach: Dual-sparse Recovery

• Dual-sparse Recovery:  $\widetilde{\mathbf{w}}_* = -\frac{1}{\lambda n} X^{\top} \widehat{\alpha}_*$ , where

$$\widehat{\alpha}_* = \arg \max_{\alpha \in \mathbb{R}^n} -\frac{1}{n} \sum_{i=1}^n \ell_i^*(\alpha_i) - \frac{1}{2\lambda n^2} \alpha^\top \widehat{X} \widehat{X}^\top \alpha - \frac{\tau}{n} \|\alpha\|_1$$

• Our theorem: if  $\alpha_*$  is s-sparse, with a high probability  $1 - \delta$ ,

$$\|\widetilde{\mathbf{w}}_* - \mathbf{w}_*\|_2 \le \epsilon \|\mathbf{w}_*\|_2, \quad \text{ where } \epsilon = \Theta\left(\sqrt{rac{s\log(n/\delta)}{m}}
ight)$$

Exploit Convex Optimization theory, JL lemma, Compressive Sensing • theory

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 つのべ

# Results

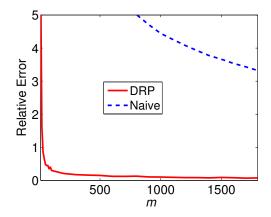
Yang (CS@Uiowa)

April 24, 2015 46 / 68

E

DQC

# Dual Recovery vs Naive Recovery



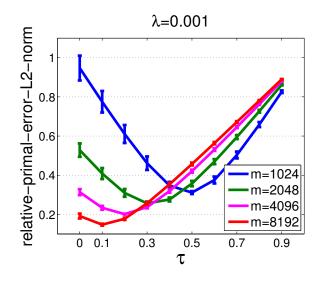
э

э

Image: A matrix and a matrix

DQC

## **Dual-sparse Recovery**



э April 24, 2015 48 / 68

э

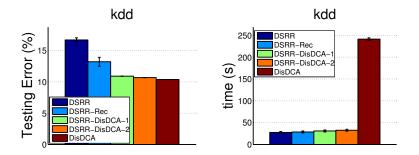
< □ > < 同 > < 回</p>

590

#### Results

# Big Data Experiments

### KDDcup Data: n = 8,407,752, d = 29,890,095, 10 machines



э

## Outline

- Machine Learning
   Introduction
- 2 Randomized Algorithms (RA)
- Our Recent work on RA for Big Data Optimization
  - Randomized Reduction and Recovery
  - Dual-sparse Randomized Reduction and Recovery
  - Results

#### Take-home Messages

## Outline

- Machine Learning
   Introduction
- 2 Randomized Algorithms (RA)
- 3 Our Recent work on RA for Big Data Optimization
  - Randomized Reduction and Recovery
  - Dual-sparse Randomized Reduction and Recovery
  - Results

#### Take-home Messages



- Machine Learning is changing our life
- Machine Learning is not just about Programming
- Big Data brings ground-breaking advances
- Randomized Algorithms are useful for Big Data
- If you are interested in any of these topics, I am happy to discuss with you.

# THANK YOU!

# Randomized Algorithms for Optimization

Yang (CS@Uiowa)

AMCS seminar

April 24, 2015 54 / 68

## Stochastic Gradient Descent in Machine Learning

$$\mathcal{F}(\mathbf{w}) = rac{1}{n} \sum_{i=1}^n \ell(\mathbf{w}^{ op} \mathbf{x}_i, y_i) + rac{\lambda}{2} \|\mathbf{w}\|_2^2$$

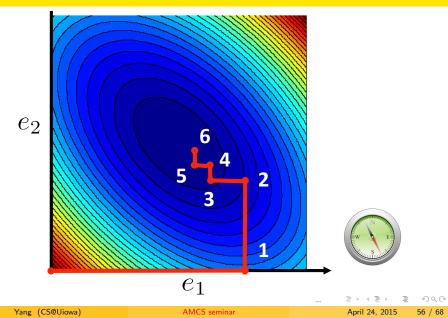
• let  $i_t \in \{1, \dots, n\}$  uniformly randomly sampled

- key equation:  $E_{i_t}[\nabla \ell(\mathbf{w}^\top \mathbf{x}_{i_t}, y_{i_t}) + \lambda \mathbf{w}] = \nabla F(\mathbf{w})$
- computation is cheaper O(d) compared with full gradient O(nd)

$$\mathbf{w}_t = (1 - \gamma_t \lambda) \mathbf{w}_{t-1} - \gamma_t 
abla \ell (\mathbf{w}_{t-1}^{ op} \mathbf{x}_{i_t}, y_{i_t})$$

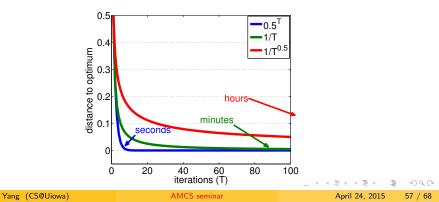
イロト (四) (三) (三) (二) (つ)

## Stochastic Coordinate Descent



# Research on Stochastic Optimization

- Establish Fast Convergence Rate for various learning problems.
- Convex Optimization Theory
- Our Research
  - SGD with only one projection for complex domains (NIPS'12)
  - Distributed Stochastic Dual Coordinate Ascent (NIPS'13)



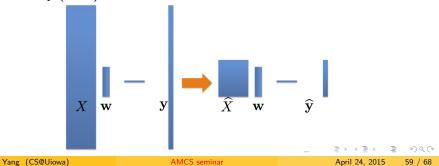
# Randomized Reduction Methods

э

# Over-constrained Least Squares Regression (LSR)

$$\min_{\mathbf{w}\in\mathbb{R}^d}\|X\mathbf{w}-\mathbf{y}\|_2, \quad \textit{where} \quad X\in\mathbb{R}^{n\times d}, n\gg d$$

- Randomized Reduction  $A \in \mathbb{R}^{m \times n} : \mathbb{R}^n \to \mathbb{R}^m$ ,  $m \ll n$
- $\min_{\mathbf{w}\in\mathbb{R}^d} \|(AX)\mathbf{w} (A\mathbf{y})\|_2$
- Time complexity:  $O(nd^2) \rightarrow o(nd^2)$
- Mahoney (2011)



# Research on Randomized Over-constrained LSR

$$egin{aligned} \mathbf{w}_* &= rg\min_{\mathbf{w}\in\mathbb{R}^d} \|X\mathbf{w}-\mathbf{y}\|_2 \ \widehat{\mathbf{w}}_* &= rg\min_{\mathbf{w}\in\mathbb{R}^d} \|(AX)\mathbf{w}-(A\mathbf{y})\|_2 \end{aligned}$$

- What is a appropriate reduction matrix  $A \in \mathbb{R}^{m \times n}$ ?
- $\bullet$  The error bound of  $\|\widehat{\boldsymbol{w}}_* \boldsymbol{w}_*\|_2$
- Convex optimization theory, random matrix theory
- Our Research
  - A New Sampling Distribution for A (to appear in ICML'15)

イロト 人間ト イヨト イヨト

# Randomized Algorithms for Low-rank Matrix Approximation

Yang (CS@Uiowa)

AMCS seminar

April 24, 2015 61 / 68

・ 何 ト ・ ヨ ト ・ ヨ ト

## low-rank matrix approximation

Many machine learning problems require computing the top-k components of the singular value decomposition (SVD)

- Principal Component Analysis
- Latent Semantic Indexing (information retrieval)

Given a  $m \times n$  large matrix, how to efficiently compute its top-k components (SVD)?

イロト イポト イヨト イヨト 二日

# RA for low-rank matrix approximation

Traditional Methods

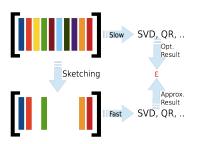
- SVD:  $O(\min(mn^2, m^2n))$
- partial SVD (for top-k components): O(mnk)
- rank revealing QR factorization: O(mnk)

Randomized Algorithms Halko et al. (2011)

- more robust
- can be as fast as  $O(mn \log(k))$

# RA for low-rank matrix approximation

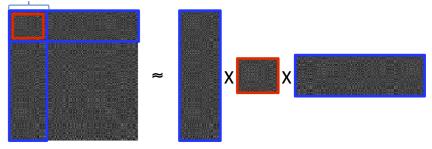
Random Sketching



- Random Projection:  $\Omega \in \mathbb{R}^{n \times \ell}$ ,  $\ell > k$  (random projection or random fourier transform); compute  $B = A\Omega \in \mathbb{R}^{m \times \ell}$ ; compute the top-k components based on B
- Column Subset Selection (CSS): sample a subset of columns
- CUR decomposition: X = CUR, sample columns and rows

# CUR decomposition for Kernel matrix

## the Nyström method



## Research on RA Low-rank Martrix Approximation

The relative error of the approximated low-rank matrix

$$\|X - \hat{X}_k\|_{2,\mathsf{F}} \leq (1+\epsilon)\|X - X_k\|_{2,\mathsf{F}}$$

- Our Research
  - Better Bounds on the Nyström method (NIPS'12, IEEE-IT)
  - Better Sampling Distributions for CSS (to appear in ICML'15).

・ロト ・同ト ・ヨト ・ヨ

## Why low-rank assumption?

$$\alpha_{*} = \arg \max_{\alpha \in \mathbb{R}^{n}} -\frac{1}{n} \sum_{i=1}^{n} \ell_{i}^{*}(\alpha_{i}) - \frac{1}{2\lambda n^{2}} \alpha^{\top} X X^{\top} \alpha$$
$$\widehat{\alpha}_{*} = \arg \max_{\alpha \in \mathbb{R}^{n}} -\frac{1}{n} \sum_{i=1}^{n} \ell_{i}^{*}(\alpha_{i}) - \frac{1}{2\lambda n^{2}} \alpha^{\top} X A^{\top} A X^{\top} \alpha$$

$$U\Sigma \underbrace{V^{\top} A^{\top} A V}_{BB^{\top}} \Sigma U^{\top}, \quad U\Sigma \underbrace{V^{\top} V}_{I_{r}} \Sigma U^{\top}$$

 $B \in \mathbb{R}^{r \times m}$  tail bounds for the eigenvalues of a sum of random matrices

$$\|BB^{\top} - I\|_2 \le O\left(\sqrt{\frac{r}{m}}\right)$$

Yang (CS@Uiowa)

イロト 不得下 イヨト イヨト 二日

## Why low-rank assumption?

$$\alpha_{*} = \arg \max_{\alpha \in \mathbb{R}^{n}} -\frac{1}{n} \sum_{i=1}^{n} \ell_{i}^{*}(\alpha_{i}) - \frac{1}{2\lambda n^{2}} \alpha^{\top} X X^{\top} \alpha$$
$$\widehat{\alpha}_{*} = \arg \max_{\alpha \in \mathbb{R}^{n}} -\frac{1}{n} \sum_{i=1}^{n} \ell_{i}^{*}(\alpha_{i}) - \frac{1}{2\lambda n^{2}} \alpha^{\top} X A^{\top} A X^{\top} \alpha$$
$$Let \ X = U \Sigma V^{\top}: \ V \in \mathbb{R}^{d \times r}$$
$$U \Sigma V^{\top} A^{\top} A V \Sigma U^{\top} \qquad U \Sigma V^{\top} V \Sigma U^{\top}$$

$$U \Sigma \underbrace{V^{+}A^{+}AV}_{BB^{\top}} \Sigma U^{+}, \quad U \Sigma \underbrace{V^{+}V}_{I_{r}} \Sigma U^{+}$$

 $B \in \mathbb{R}^{r \times m}$  tail bounds for the eigenvalues of a sum of random matrices

$$\|BB^{\top} - I\|_2 \leq O\left(\sqrt{\frac{r}{m}}\right)$$

Yang (CS@Uiowa)

April 24, 2015 67 / 68

3

## Why low-rank assumption?

$$\alpha_{*} = \arg \max_{\alpha \in \mathbb{R}^{n}} -\frac{1}{n} \sum_{i=1}^{n} \ell_{i}^{*}(\alpha_{i}) - \frac{1}{2\lambda n^{2}} \alpha^{\top} X X^{\top} \alpha$$
$$\widehat{\alpha}_{*} = \arg \max_{\alpha \in \mathbb{R}^{n}} -\frac{1}{n} \sum_{i=1}^{n} \ell_{i}^{*}(\alpha_{i}) - \frac{1}{2\lambda n^{2}} \alpha^{\top} X A^{\top} A X^{\top} \alpha$$
$$Let \ X = U \Sigma V^{\top}: \ V \in \mathbb{R}^{d \times r}$$
$$U \Sigma \underbrace{V^{\top} A^{\top} A V}_{BB^{\top}} \Sigma U^{\top}, \quad U \Sigma \underbrace{V^{\top} V}_{I_{r}} \Sigma U^{\top}$$

 $B \in \mathbb{R}^{r imes m}$  tail bounds for the eigenvalues of a sum of random matrices

$$\|BB^{\top} - I\|_2 \leq O\left(\sqrt{\frac{r}{m}}\right)$$

996

イロト 不得下 イヨト イヨト 二日

## References I

- Halko, N., Martinsson, P. G., and Tropp, J. A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53:217–288, 2011.
- Johnson, William and Lindenstrauss, Joram. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26, pp. 189–206. 1984.
- Mahoney, Michael W. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3:123–224, 2011.

イロト イポト イヨト イヨト 二日