Randomized Algorithms for Large-scale Convex Optimization

Lijun Zhang

LAMDA group, Nanjing University, China

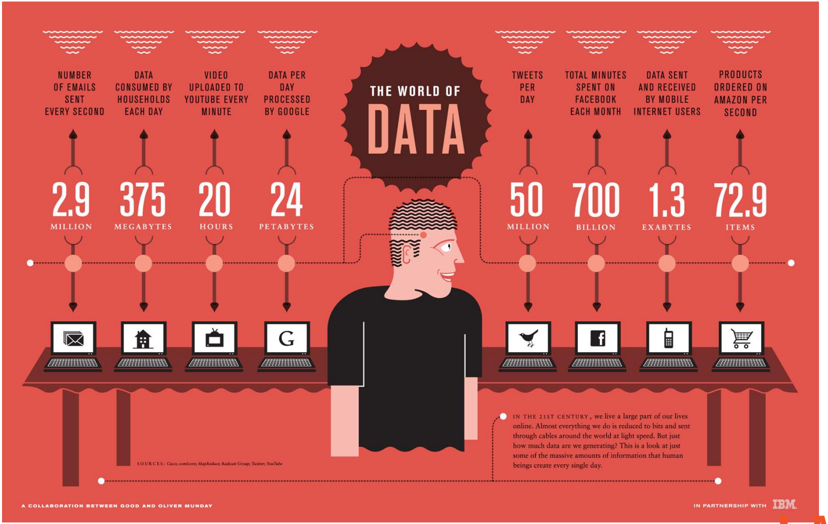
The 2nd Chinese Workshop on Evolutionary Computation and Learning

Outline

LAMDA
Learning And Mining from DatA

Outline

Big Data



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

Supervised Learning by Optimization

Supervised Learning

Input

- A set of training data $\{(\mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R})\}_{i=1}^n$
- A set of hypotheses $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^d$

Output

- A hypothesis $\mathbf{w}_* \in \mathcal{W}$ that minimizes testing error

$$\mathbf{x} \mapsto \mathbf{x}^\top \mathbf{w}_*$$

Empirical Risk Minimization

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

- $\ell(\cdot, \cdot)$ 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

The Challenges

Large-scale Convex Optimization

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

Gradient Descent (GD)

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

Computational Cost

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

LAMDA Learning And Mining from DatA

The Challenges

Large-scale Convex Optimization

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

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: $O(nd)$

The Challenges

Large-scale Convex Optimization

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

Gradient Descent (GD)

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

Computational Cost

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

LAMDA Learning And Mining from DatA

The Challenges

Large-scale Convex Optimization

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

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: $O(nd)$

The Challenges

Large-scale Convex Optimization

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

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: $O(nd)$

The Challenges

Large-scale Convex Optimization

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

Gradient Descent (GD)

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

Computational Cost

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

LAMDA Learning And Mining from DatA

Randomized Algorithms

Random Sampling based Algorithms

- aim to address the large-scale challenge, i.e., large n
- select a subset of training data randomly
- referred to as *Stochastic Optimization*

Random Projection based Algorithms

- aim to address the high-dimensional challenge, i.e., large d
- reduce the dimensionality by random projection
- referred to as *Stochastic Approximation*

Outline

LAMDA Learning And Mining from DatA

Outline

LAMDA Learning And Mining from DatA

Stochastic Gradient Descent (SGD)

The Algorithm

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

Advantages

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

Limitations

- The iteration complexity is much higher than GD

LAMDA Learning And Mining from DatA

Stochastic Gradient Descent (SGD)

The Algorithm

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

Advantages

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

Limitations

- The iteration complexity is much higher than GD

LAMDA Learning And Mining from DatA

Stochastic Gradient Descent (SGD)

The Algorithm

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

Advantages

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

Limitations

- The iteration complexity is much higher than GD

Stochastic Gradient Descent (SGD)

The Algorithm

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

Advantages

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

Limitations

- The iteration complexity is much higher than GD

Stochastic Gradient Descent (SGD)

The Algorithm

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

Advantages

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

Limitations

- The iteration complexity is much higher than GD

Stochastic Gradient Descent (SGD)

The Algorithm

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

Advantages

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

Limitations

- The iteration complexity is much higher than GD

Outline

The Problem

Iteration Complexity

The number of iterations T to ensure

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

Comparisons between GD and SGD

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

Note

$$\frac{1}{\epsilon^2} > \frac{1}{\epsilon} > \frac{1}{\sqrt{\epsilon}} \gg \log\frac{1}{\epsilon}$$

$$10^{12} > 10^6 > 10^3 \gg 6, \ \epsilon = 10^{-6}$$

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

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

Stochastic Gradients introduce a constant error

The key idea

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

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

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

Stochastic Gradients introduce a constant error

The key idea

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

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

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

Stochastic Gradients introduce a constant error

The key idea

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

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

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

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

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

Mixed Gradient Descent II

The Algorithm (NIPS 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: Select a training instance $(\mathbf{x}_i, y_i)$ 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: $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \mathbf{m}(\mathbf{w}_t)$

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

7: **end for**

Learning And Mining from DatA

Theoretical Guarantees

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

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	$O\left(\log \frac{1}{\epsilon}\right)$	$O\left(\kappa^2 \log \frac{1}{\epsilon}\right)$

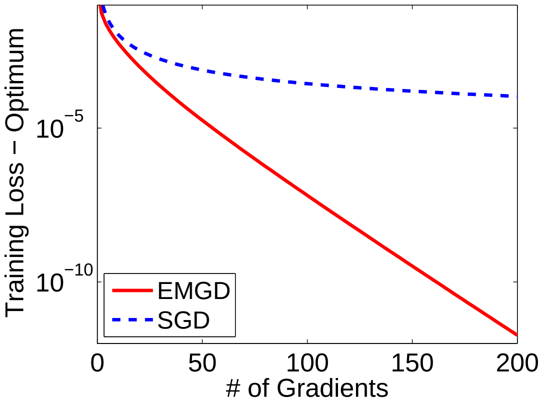
In contrast, SGD needs $O(1/\epsilon)$ stochastic gradients.

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]

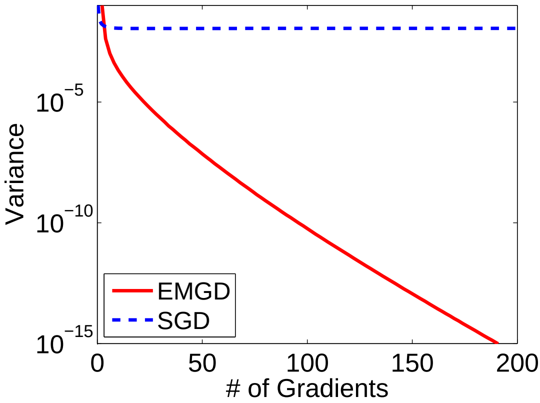
Experimental Results I

- 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

Outline

The Power of Random Projection

Random Projection

A dimensionality reduction method:

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

where $A \in \mathbb{R}^{d \times m}$ and $A_{ij} \sim \mathcal{N}(0, 1/m)$

Theorem 1 (Johnson and Lindenstrauss Lemma [Achlioptas, 2003])

Given $\epsilon > 0$ and an integer n, let m be a positive integer such that $m = \Omega(\epsilon^{-2} \log n)$. For every set P of n points in $\mathbb{R}^d$ there exists $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$ such that for all $\mathbf{x}_i, \mathbf{x}_j \in P$

$$(1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|^2 \leq (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2.$$

Optimization after Random Projection I

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

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

Traditional Approach

1. Reduce the dimensionality $\widehat{\mathbf{x}}_i = A^\top \mathbf{x}_i \in \mathbb{R}^m$
2. Solve the primal problem in $\mathbb{R}^m$

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

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 II

Advantages

- Time complexity is reduced from $O(nd)$ to $O(nm)$
- Space complexity is reduced from $O(nd)$ to $O(nm)$
- It is possible to run gradient descent which converges fast

The Limitation

$\widehat{\mathbf{w}}$ is not a good approximation of

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

Optimization after Random Projection III

Proposition 1 (Distance of a Random Subspace to a Fixed Point [Vershynin, 2009])

Let $E \in G_{d,m}$ be a random subspace (codim $E = d - m$). Let $\mathbf{x}$ be an united length vector, which is arbitrary but fixed. Then

$$\Pr\left(dist(\mathbf{x}, E) \leq \epsilon\sqrt{\frac{d-m}{d}} \right) \leq (c\epsilon)^{d-m} \text{ for any } \epsilon > 0,$$

where c is an universal constant.

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

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

Outline

LAMDA
Learning And Mining from DatA

Motivations I

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

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i,\mathbf{x}_i^\top\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2, \quad \text{(P1)}$$

The Dual Problem

$$\max_{\boldsymbol{\alpha}\in\Omega^n} -\sum_{i=1}^{n}\ell_*(\alpha_i) - \frac{1}{2n\lambda}(\boldsymbol{\alpha}\circ\mathbf{y})^\top X^\top X(\boldsymbol{\alpha}\circ\mathbf{y}), \quad \text{(D1)}$$

Proposition 2

Let $\mathbf{w}_* \in \mathbb{R}^d$ and $\boldsymbol{\alpha}_* \in \mathbb{R}^n$ be solutions to (P1) and (D1).

$$\mathbf{w}_* = -\frac{1}{\lambda n}X(\boldsymbol{\alpha}_*\circ\mathbf{y}),$$

$$[\boldsymbol{\alpha}_*]_i = \ell'\left(y_i,\mathbf{x}_i^\top\mathbf{w}_*\right), \; i = 1,\ldots,n.$$

Motivations II

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

$$\min_{\mathbf{z}\in\mathbb{R}^m} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, \mathbf{z}^\top \widehat{\mathbf{x}}_i) + \frac{\lambda}{2}\|\mathbf{z}\|^2, \quad \text{(P2)}$$

The Dual Problem

$$\max_{\boldsymbol{\alpha}\in\Omega^n} -\sum_{i=1}^{n}\ell_*(\alpha_i) - \frac{1}{2\lambda n}(\boldsymbol{\alpha}\circ\mathbf{y})^\top X^\top A A^\top X(\boldsymbol{\alpha}\circ\mathbf{y}), \quad \text{(D2)}$$

Proposition 3

Let $\mathbf{z}_* \in \mathbb{R}^m$ and $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$ be solutions to (P2) and (D2).

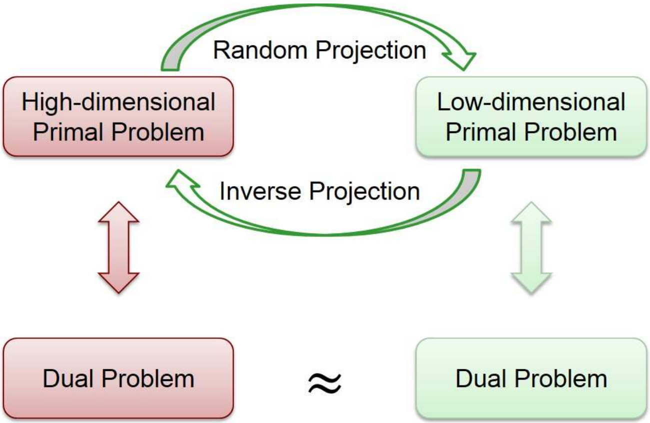
$$\mathbf{z}_* = -\frac{1}{\lambda n}A^\top X(\widehat{\boldsymbol{\alpha}}_* \circ \mathbf{y}),$$

$$[\widehat{\boldsymbol{\alpha}}_*]_i = \ell'\left(y_i, \widehat{\mathbf{x}}_i^\top \mathbf{z}_*\right), \ i = 1, \ldots, n.$$

Motivations III

The Big Picture

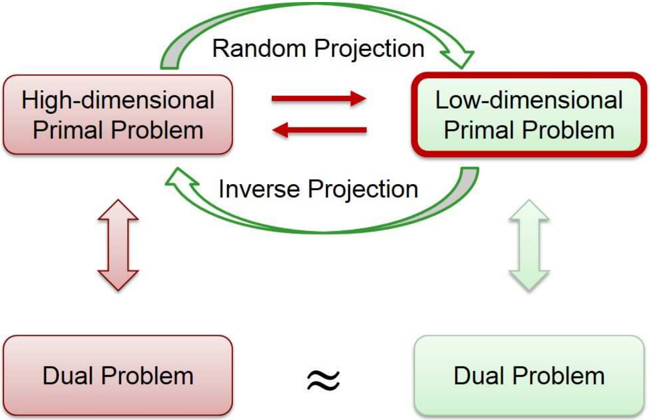
Primal-Primal Primal-Dual Dual-Dual



Motivations IV

Optimization after Random Projection

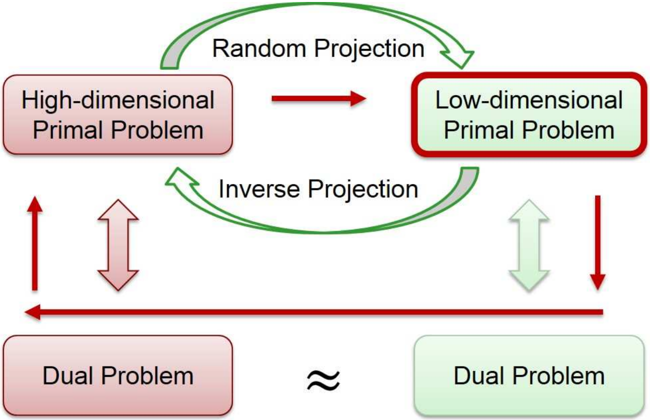
Primal Solution $\rightarrow$ Primal Solution



Dual Random Projection

Use Dual Solutions to Bridge Primal Solutions

Primal Solution $\rightarrow$ Dual Solution $\rightarrow$ Primal Solution



Dual Random Projection

The Algorithm (COLT 2013 & IEEE Trans. Inf. Theory 2014)

1. Reduce the dimensionality $\widehat{\mathbf{x}}_i = A^\top \mathbf{x}_i \in \mathbb{R}^m$
2. Solve the low-dimensional problem

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

3. Construct the dual solution $\widehat{\alpha}_* \in \mathbb{R}^n$ by

$$[\widehat{\alpha}_*]_i = \ell'\left(y_i, \widehat{\mathbf{x}}_i^\top \mathbf{z}_*\right), \ i = 1, \ldots, n$$

4. Compute $\widetilde{\mathbf{w}} \in \mathbb{R}^d$ by

$$\widetilde{\mathbf{w}} = -\frac{1}{\lambda n} X(\widehat{\alpha}_* \circ \mathbf{y})$$

Theoretical Guarantees

Low-rank Assumption

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

Theorem 2 ([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 $\Omega(r \log r)$.

Theoretical Guarantees

Low-rank Assumption

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

Theorem 2 ([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

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

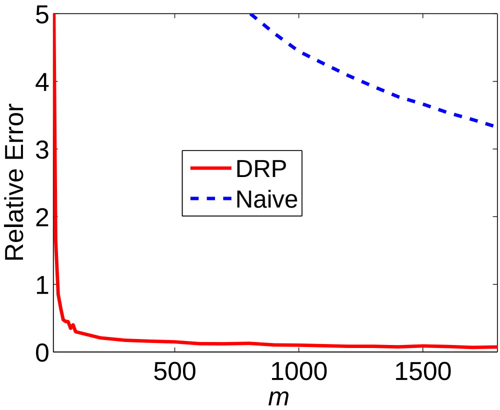
where constant c is at least $1/4$*.*

Implication

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

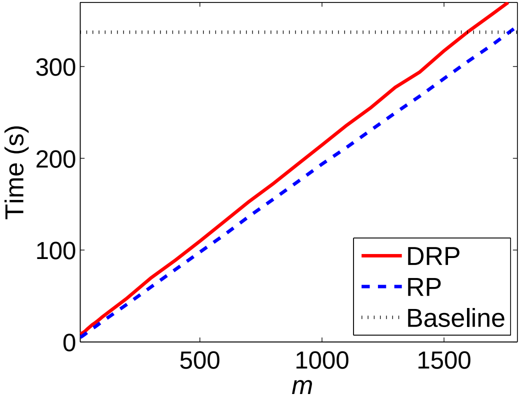
Experimental Results I

- 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

Conclusions and Future Work

Summary

- Based on random sampling, we propose a Mixed Gradient Descent (MGD) algorithm which improves the convergence rate significantly.
- Based on random projection, we propose a Dual Random Projection (DRP) algorithm which can recover the optimal solution accurately.

Future Work

- Extend MGD to distributed environments
- Relax assumptions in Dual Random Projection [Yang et al., 2015]
- Extend DRP to more problems, such as sparse learning

LAMDA Learning And Mining from DatA

Conclusions and Future Work

Summary

- Based on random sampling, we propose a Mixed Gradient Descent (MGD) algorithm which improves the convergence rate significantly.
- Based on random projection, we propose a Dual Random Projection (DRP) algorithm which can recover the optimal solution a[illegible]

Thanks!

Future Work

- Extend MGD to distributed environments
- Relax assumptions in Dual Random Projection [Yang et al., 2015]
- Extend DRP to more problems, such as sparse learning

Reference I

,

Achlioptas, D. (2003).
Database-friendly random projections: Johnson-lindenstrauss with binary coins.
Journal of Computer and System Sciences, 66(4):671 – 687.

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.

Vershynin, R. (2009).
Lectures in geometric functional analysis.
Technical report, University of Michigan.

Yang, T., Zhang, L., Jin, R., and Zhu, S. (2015).
Theory of dual-sparse regularized randomized reduction.
In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*.

Learning And Mining from DatA

Reference II

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 (TIT), 60(11):7300–7316.

LAMDA Learning And Mining from DatA