Random Projections for Classification: A Recovery Approach

Mehrdad Mahdavi

Toyota Technological Institute at University of Chicago

May 2015

Randomized Algorithms in Large-Scale Learning

Large-Scale Learning

■ Two main issues in modern data: size and dimensionality.

■ Large data sizes can cause access and storage problem: parallelization (divide and conquer) or stochastic methods

■ High-dimensional data suffer from statistical issues: make structural assumptions about the data such as *sparsity* or *low-rank*

Randomized Methods

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

Randomized Methods

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

Algorithms:

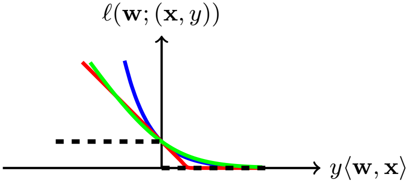
- Stochastic optimization for large-scale learning
- Randomized low-rank approximations for kernelized learning
- Random projections for high-dimensional learning
- Sketching for numerical linear algebra and matrix computation

Stochastic Optimization

- An effective optimization method for learning from **large data sizes**

Stochastic Optimization

- An effective optimization method for learning from **large data sizes**
- Surrogate convex loss functions of non-convex 0-1 loss:

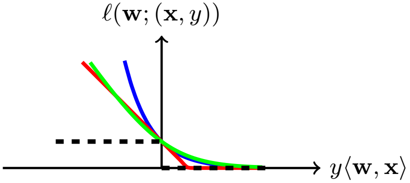


Examples:

- ✓ Hinge loss (Support Vector Machine (SVM)): $\ell(\mathbf{w}; (\mathbf{x}, y)) = \max(0, 1 - y\langle \mathbf{w}, \mathbf{x} \rangle)$.
- ✓ Logistic loss (Logistic Regression): $\ell(\mathbf{w}; (\mathbf{x}, y)) = \log(1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle))$.
- ✓ Exponential loss (Boosting): $\ell(\mathbf{w}; (\mathbf{x}, y)) = \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle)$.

Stochastic Optimization

- An effective optimization method for learning from **large data sizes**
- Surrogate convex loss functions of non-convex 0-1 loss:



Examples:

- ✓ Hinge loss (Support Vector Machine (SVM)): $\ell(\mathbf{w}; (\mathbf{x}, y)) = \max(0, 1 - y\langle \mathbf{w}, \mathbf{x} \rangle)$.
- ✓ Logistic loss (Logistic Regression): $\ell(\mathbf{w}; (\mathbf{x}, y)) = \log(1 + \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle))$.
- ✓ Exponential loss (Boosting): $\ell(\mathbf{w}; (\mathbf{x}, y)) = \exp(-y\langle \mathbf{w}, \mathbf{x} \rangle)$.

- **Convex learning problems:**

$$\min_{\mathbf{w} \in \mathcal{W}} \left[L_{\mathcal{S}}(\mathbf{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}; (\mathbf{x}_i, y_i)) + \lambda \|\mathbf{w}\| \right]$$

Stochastic Optimization

Empirical risk minimization as a convex optimization problem:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}} \left(\mathbf{w}_t - \eta \mathbf{g}_t \right)$$

Empirical risk minimization as a convex optimization problem:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left(\mathbf{w}_t - \eta \mathbf{g}_t\right)$$

Two regimes:

❶ GD: all samples per Iteration [deterministic]

$$\mathbf{g}_t = \nabla L_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell(\mathbf{w}; (\mathbf{x}_i, y_i))\mathbf{x}_i.$$

Stochastic Optimization

Empirical risk minimization as a convex optimization problem:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left(\mathbf{w}_t - \eta \mathbf{g}_t\right)$$

Two regimes:

❶ GD: all samples per Iteration [deterministic]

$$\mathbf{g}_t = \nabla L_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell(\mathbf{w}; (\mathbf{x}_i, y_i))\mathbf{x}_i.$$

❷ SGD: one sample per Iteration [stochastic]

$$\mathbf{g}_t = \nabla \ell(\mathbf{w}; (\mathbf{x}_{i_t}, y_{i_t}))\mathbf{x}_{i_t}$$

where is random sample $(\mathbf{x}_{i_t}, y_{i_t})$ uniformly sampled from $[n]$

Stochastic Optimization

Empirical risk minimization as a convex optimization problem:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left(\mathbf{w}_t - \eta \mathbf{g}_t\right)$$

Two regimes:

❶ GD: all samples per Iteration [deterministic]

$$\mathbf{g}_t = \nabla L_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \nabla \ell(\mathbf{w}; (\mathbf{x}_i, y_i))\mathbf{x}_i.$$

❷ SGD: one sample per Iteration [stochastic]

$$\mathbf{g}_t = \nabla \ell(\mathbf{w}; (\mathbf{x}_{i_t}, y_{i_t}))\mathbf{x}_{i_t}$$

where is random sample $(\mathbf{x}_{i_t}, y_{i_t})$ uniformly sampled from $[n]$

✔ SGD: efficient for large-scale learning (independent of number of training examples n)

Nyström Method

- Low-rank approximation of symmetric positive semi-definite matrices such as Laplacian and kernel matrices

Nyström Method

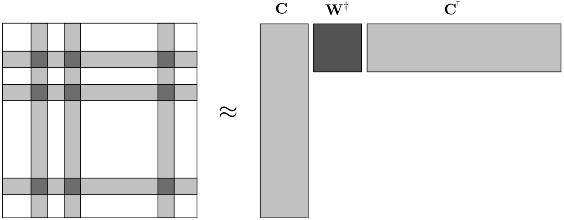
■ Low-rank approximation of symmetric positive semi-definite matrices such as Laplacian and kernel matrices

√ Let $\mathbf{K} \in \mathbb{R}^{n\times n}$ be kernel matrix of n training samples $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^d$, with $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$

■ Low-rank approximation of symmetric positive semi-definite matrices such as Laplacian and kernel matrices

√ Let $\mathbf{K} \in \mathbb{R}^{n \times n}$ be kernel matrix of n training samples $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^d$, with $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$

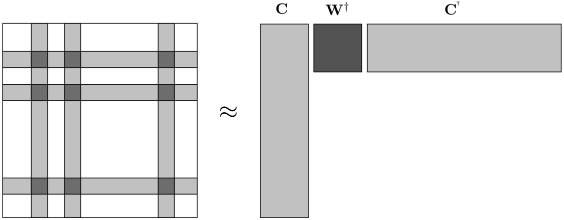
√ Randomly sample $m \ll n$ training examples out of n samples



■ Low-rank approximation of symmetric positive semi-definite matrices such as Laplacian and kernel matrices

√ Let $\mathbf{K} \in \mathbb{R}^{n\times n}$ be kernel matrix of n training samples $\mathcal{S} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}, \mathbf{x}_i \in \mathbb{R}^d$, with $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$

√ Randomly sample $m \ll n$ training examples out of n samples



√ Approximate the kernel matrix by $\widehat{\mathbf{K}} = \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^{\top}$

Nyström Approximation

- Matrix inversion lemma (Woodbury):

$$
\begin{aligned}
&(\lambda \mathbf{I} + \mathbf{K})^{-1} \\
&\approx (\lambda \mathbf{I} + \widehat{\mathbf{K}})^{-1} \\
&= \left(\lambda \mathbf{I} + \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^{\top}\right)^{-1} \\
&= \frac{1}{\lambda}\left(\mathbf{I} - \mathbf{C}\underbrace{\left(\lambda \mathbf{I} + \mathbf{W}^{\dagger}\mathbf{C}^{\top}\mathbf{C}\right)^{-1}}_{m \times m}\mathbf{W}^{\dagger}\mathbf{C}^{\top}\right)
\end{aligned}
$$

Nyström Approximation

- Matrix inversion lemma (Woodbury):

$$
\begin{aligned}
&(\lambda \mathbf{I} + \mathbf{K})^{-1} \\
&\approx (\lambda \mathbf{I} + \widehat{\mathbf{K}})^{-1} \\
&= \left(\lambda \mathbf{I} + \mathbf{C}\mathbf{W}^{\dagger}\mathbf{C}^{\top}\right)^{-1} \\
&= \frac{1}{\lambda}\left(\mathbf{I} - \mathbf{C}\underbrace{\left(\lambda \mathbf{I} + \mathbf{W}^{\dagger}\mathbf{C}^{\top}\mathbf{C}\right)^{-1}}_{m \times m}\mathbf{W}^{\dagger}\mathbf{C}^{\top}\right)
\end{aligned}
$$

- Only requires inversion of a $m \times m$ matrix: $\left(\lambda \mathbf{I} + \mathbf{W}^{\dagger}\mathbf{C}^{\top}\mathbf{C}\right)^{-1}$

✔ $O(n^3)$ versus $O(nmk) + O(m^3)$: efficient large-scale learning!

✔ SVMs, kernel ridge regression, KPCA, spectral clustering, and etc

Random Projections

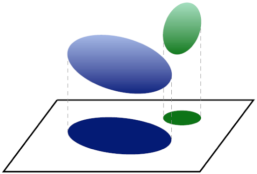
- An effective **dimensionality reduction** method

- An effective **dimensionality reduction** method

$\mathbb{R}^d$

Random Projection

$\mathbb{R}^m, m \ll d$



$\mathbf{w}_* \in \mathbb{R}^d$

$\mathbf{z}_* \in \mathbb{R}^m$

- Classification, clustering, range query (e.g., hashing)

Sketching

- Generate a sketch of data points with applications in regression, graph sparsification, numerical linear algebra

Sketching

- Generate a sketch of data points with applications in regression, graph sparsification, numerical linear algebra
- Regression: often too slow to be of practical value

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

- Generate a sketch of data points with applications in regression, graph sparsification, numerical linear algebra
- Regression: often too slow to be of practical value

$$\min_{\mathbf{w} \in \mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

- A sketching technique

Sketching

- Generate a sketch of data points with applications in regression, graph sparsification, numerical linear algebra
- Regression: often too slow to be of practical value

$$\min_{\mathbf{w} \in \mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

- A sketching technique
  - √ Sample a **random** matrix $\mathbf{R} \in \mathbb{R}^{r \times n}$

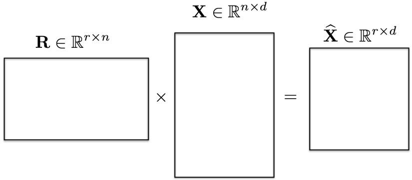
Sketching

- Generate a sketch of data points with applications in regression, graph sparsification, numerical linear algebra

- Regression: often too slow to be of practical value

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

- A sketching technique
  - √ Sample a **random** matrix $\mathbf{R} \in \mathbb{R}^{r\times n}$
  - √ Compute a sketch of the data matrix $\mathbf{X}$

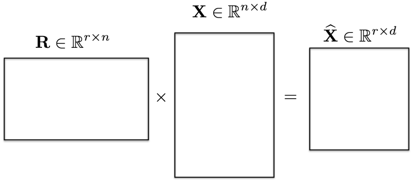


Sketching

- Generate a sketch of data points with applications in regression, graph sparsification, numerical linear algebra
- Regression: often too slow to be of practical value

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$

- A sketching technique
  - √ Sample a **random** matrix $\mathbf{R} \in \mathbb{R}^{r\times n}$
  - √ Compute a sketch of the data matrix $\mathbf{X}$



  - √ Solve the sketched problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{R}\mathbf{X}\mathbf{w} - \mathbf{R}\mathbf{y}\|_2^2$$

Random Projections
for
High-dimensional Classification

The Classification Problem

- Input: a set of training samples from $\mathcal{X} \subseteq \mathbb{R}^d \times \{-1, +1\}$

$$\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n))$$

The Classification Problem

- Input: a set of training samples from $\mathcal{X} \subseteq \mathbb{R}^d \times \{-1, +1\}$

$$\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n))$$

- A loss function: a differentiable convex loss function with Lipschitz gradient, e.g.,

$$\ell(z) = \ln\left(\log(1 + \exp(-z))\right)$$

The Classification Problem

- Input: a set of training samples from $\mathcal{X} \subseteq \mathbb{R}^d \times \{-1, +1\}$

$$\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n))$$

- A loss function: a differentiable convex loss function with Lipschitz gradient, e.g.,

$$\ell(z) = \ln\left(\log(1 + \exp(-z))\right)$$

- The goal: learn a classifier $f : \mathbb{R}^d \to \{\pm 1\}$ from training set $\mathcal{S}$.

The Classification Problem

- Input: a set of training samples from $\mathcal{X} \subseteq \mathbb{R}^d \times \{-1, +1\}$

$$\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n))$$

- A loss function: a differentiable convex loss function with Lipschitz gradient, e.g.,

$$\ell(z) = \ln\left(\log(1 + \exp(-z))\right)$$

- The goal: learn a classifier $f : \mathbb{R}^d \to \{\pm 1\}$ from training set $\mathcal{S}$.

Method: Regularized ERM

1. Solve the high-dimensional problem

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

2. Classify using the function

$$f(\mathbf{x}) = \operatorname{sign}\left(\mathbf{x}^\top \mathbf{w}_*\right)$$

where $\mathbf{w}_*$ is the optimal solution of (P1).

Random Projection

- A dimensionality reduction method

$$\mathbf{x} \in \mathbb{R}^d \to \frac{1}{\sqrt{m}}\mathbf{R}^\top \mathbf{x} \in \mathbb{R}^m$$

where $\mathbf{R} \in \mathbb{R}^{d\times m}$ is a (Gaussian) random matrix, i.e., $\mathbf{R}_{i,j} \sim \mathcal{N}(0,1)$.

Random Projection

- A dimensionality reduction method

$$\mathbf{x} \in \mathbb{R}^d \to \frac{1}{\sqrt{m}} \mathbf{R}^\top \mathbf{x} \in \mathbb{R}^m$$

where $\mathbf{R} \in \mathbb{R}^{d \times m}$ is a (Gaussian) random matrix, i.e., $\mathbf{R}_{i,j} \sim \mathcal{N}(0, 1)$.

- Simple yet powerful (satisfying JL lemma)

Theorem 1 (Johnson and Lindenstrauss)

Given $\epsilon > 0$ and an integer n, let $m = \Omega(\epsilon^{-2} \log n)$. For every set $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of n points in $\mathbb{R}^d$, $\exists$ a mapping $\mathfrak{M} : \mathbb{R}^d \to \mathbb{R}^m$ such that for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}$

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

Random Projection

- A dimensionality reduction method

$$\mathbf{x} \in \mathbb{R}^d \rightarrow \frac{1}{\sqrt{m}} \mathbf{R}^\top \mathbf{x} \in \mathbb{R}^m$$

where $\mathbf{R} \in \mathbb{R}^{d \times m}$ is a (Gaussian) random matrix, i.e., $\mathbf{R}_{i,j} \sim \mathcal{N}(0, 1)$.

- Simple yet powerful (satisfying JL lemma)

Theorem 1 (Johnson and Lindenstrauss)

Given $\epsilon > 0$ and an integer n, let $m = \Omega(\epsilon^{-2} \log n)$. For every set $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ of n points in $\mathbb{R}^d$, $\exists$ a mapping $\mathfrak{M} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ such that for all $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}$

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

✔ The gist of proof: the squared length of a vector is sharply concentrated around its mean when projected onto a random m-dimensional subspace

✔ Classification, clustering, regression, manifold learning, hashing

[Johnson and Lindenstrauss 1984, Achlioptas 2003]

Random Projection for Classification

RP for ERM

1. Apply random projection to reduce the dimensionality

$$\widehat{\mathbf{x}}_i = \frac{1}{\sqrt{m}}\mathbf{R}^\top \mathbf{x}_i$$

Random Projection for Classification

RP for ERM

❶ Apply random projection to reduce the dimensionality

$$\widehat{\mathbf{x}}_i = \frac{1}{\sqrt{m}}\mathbf{R}^\top \mathbf{x}_i$$

❷ Solve the low-dimensional problem

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

Random Projection for Classification

RP for ERM

❶ Apply random projection to reduce the dimensionality

$$\widehat{\mathbf{x}}_i = \frac{1}{\sqrt{m}} \mathbf{R}^\top \mathbf{x}_i$$

❷ Solve the low-dimensional problem

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

❸ Classify using the function

$$\hat{f}(\mathbf{x}) = \text{sign}\left(\frac{1}{\sqrt{\text{m}}} \mathbf{x}^\top \mathbf{R} \mathbf{z}_* \right) = \text{sign}\left(\mathbf{x}^\top \widehat{\mathbf{w}} \right)$$

where $\mathbf{z}_*$ is the optimal solution of (P2) and $\widehat{\mathbf{w}} = \frac{\mathbf{R}\mathbf{z}_*}{\sqrt{m}}$

Random Projection for Classification

RP for ERM

❶ Apply random projection to reduce the dimensionality

$$\widehat{\mathbf{x}}_i = \frac{1}{\sqrt{m}}\mathbf{R}^\top \mathbf{x}_i$$

❷ Solve the low-dimensional problem

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

❸ Classify using the function

$$\hat{f}(\mathbf{x}) = \text{sign}\left(\frac{1}{\sqrt{\mathrm{m}}}\mathbf{x}^\top \mathbf{R}\mathbf{z}_*\right) = \text{sign}\left(\mathbf{x}^\top \widehat{\mathbf{w}}\right)$$

where $\mathbf{z}_*$ is the optimal solution of (P2) and $\widehat{\mathbf{w}} = \frac{\mathbf{R}\mathbf{z}_*}{\sqrt{m}}$

- Let's call $\widehat{\mathbf{w}} \in \mathbb{R}^d$ the naive solution to original learning problem

Random Projections and Recovery Problem

Question I

√ Is naive solution $\widehat{\mathbf{w}} = \frac{1}{\sqrt{m}}\mathbf{R}\mathbf{z}_*$ a good classifier?

Question I

√ Is naive solution $\widehat{\mathbf{w}} = \frac{1}{\sqrt{m}}\mathbf{R}\mathbf{z}_*$ a good classifier?
Yes.

Theorem 2

If the data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ *is linearly separable by normalized margin* $\gamma \in (0, 1)$, *then for any* $\delta, \epsilon \in (0, 1)$ *and any*

$$m \geq \frac{12}{3\epsilon^2 - 2\epsilon^3} \ln \frac{6n}{\delta},$$

w.p at least $1 - \delta$, *the data set* $\{(\mathbf{R}^\top \mathbf{x}_i, y_i)\}_{i=1}^n$ *is linearly separable by margin*

$$\gamma - \frac{2\epsilon}{1 - \epsilon}.$$

- We note this holds for normalized margin defined as $\gamma = y_i \frac{\mathbf{u}^\top \mathbf{x}_i}{\|\mathbf{u}\| \|\mathbf{x}_i\|}$

- The argument can be generalized to error allowed margin

[Balcan et. al., COLT'04, Shi et. al., ICML'12]

Question II

- $\surd$ Is naive solution $\widehat{\mathbf{w}} = \frac{1}{\sqrt{m}}\mathbf{R}\mathbf{z}_*$ a good approximation of $\mathbf{w}_*$?

Question II

√ Is naive solution $\widehat{\mathbf{w}} = \frac{1}{\sqrt{m}}\mathbf{R}\mathbf{z}_*$ a good approximation of $\mathbf{w}_*$?
No.

■ $\widehat{\mathbf{w}}$ lies in a random subspace spanned by the column vectors in $\mathbf{R}$!

√ Is naive solution $\widehat{\mathbf{w}} = \frac{1}{\sqrt{m}}\mathbf{R}\mathbf{z}_*$ a good approximation of $\mathbf{w}_*$?
No.

■ $\widehat{\mathbf{w}}$ lies in a random subspace spanned by the column vectors in $\mathbf{R}$!

Theorem 3 (Distance of a Random Subspace to a Fixed Point)

For any $0 < \varepsilon \leq 1/3$*, with a probability at least* $1 - \exp(-(d-r)/32) - \exp(-m/32) - \delta$*, we have*

$$\|\widehat{\mathbf{w}} - \mathbf{w}_*\|_2 \geq \frac{1}{2}\sqrt{\frac{d-r}{m}}\left(1 - \frac{\varepsilon\sqrt{2(1+\varepsilon)}}{1-\varepsilon}\right)\|\mathbf{w}_*\|_2,$$

provided

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

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

[Lectures in Geometric Functional Analysis, Roman Vershynin]

A Natural Question

■ From low-dimensional solution to original high-dimensional optimal solution

The Recovery Problem

Is it possible to accurately recover $\mathbf{w}_*$, the optimal solution of

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

from $\mathbf{z}_*$, the optimal solution of

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

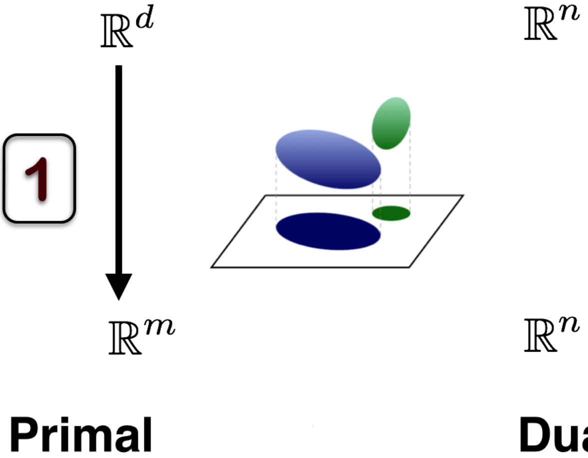
■ Is it possible to accurately recover the optimal solution $\mathbf{w}_* \in \mathbb{R}^d$ based on $\mathbf{z}_* \in \mathbb{R}^m$, the optimal solution to low-dimensional optimization problem?

Possible applications: feature selection [?]

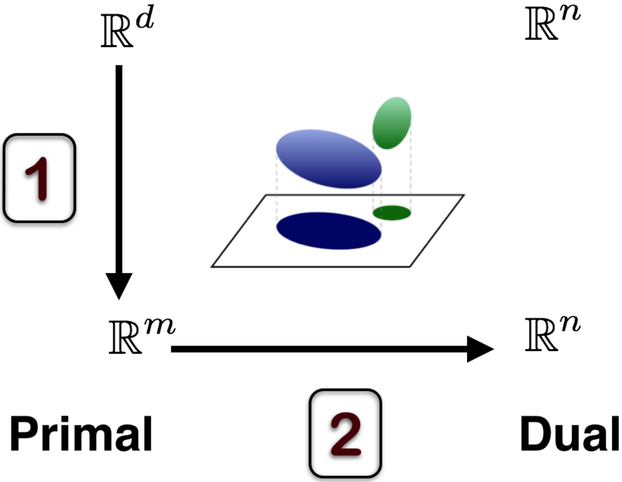
Outline

The Main Idea

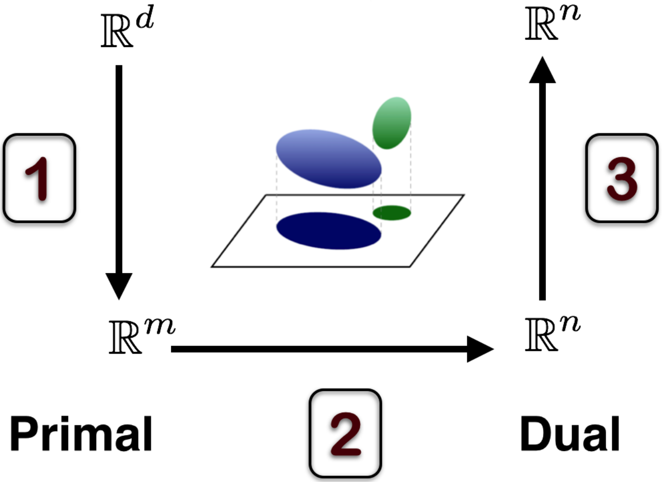
❶ Project the data and compute $\mathbf{z}_* \in \mathbb{R}^m$



❷ Compute $\widehat{\alpha}_* \in \mathbb{R}^m$ from $\mathbf{z}_* \in \mathbb{R}^m$

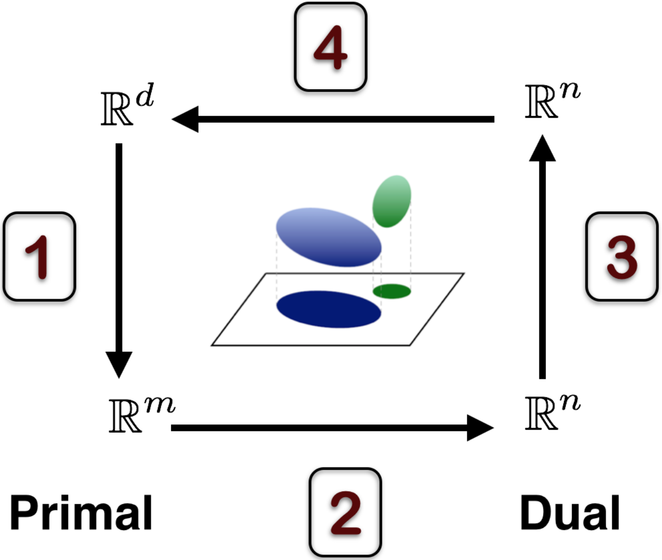


❸ Compute $\boldsymbol{\alpha}_* \in \mathbb{R}^n$ from $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$



The Main Idea

❹ Compute $\mathbf{w}_* \in \mathbb{R}^d$ from $\boldsymbol{\alpha}_* \in \mathbb{R}^n$



Outline

Primal and Dual Problems

√ The primal problem

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

Primal and Dual Problems

√ The primal problem

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

√ $\ell(z)$ can be written as

$$\ell(z) = \max_{\alpha\in\Omega} \alpha z - \ell_*(\alpha),$$

where $\ell_*(\alpha)$ is the convex conjugate of $\ell(z)$.

Primal and Dual Problems

√ The primal problem

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

√ $\ell(z)$ can be written as

$$\ell(z) = \max_{\alpha\in\Omega} \alpha z - \ell_*(\alpha),$$

where $\ell_*(\alpha)$ is the convex conjugate of $\ell(z)$.

√ The dual problem

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

where $\mathbf{G} = \mathrm{diag}(\mathbf{y})\mathbf{X}^\top\mathbf{X}\mathrm{diag}(\mathbf{y})$.

From Primal to Dual

Proposition 1

Let $\mathbf{w}_* \in \mathbb{R}^d$ be the optimal primal solution to (P1), and $\boldsymbol{\alpha}_* \in \mathbb{R}^n$ be the optimal dual solution to (D1). We have

$$\mathbf{w}_* = -\frac{1}{\lambda}\mathbf{X}\mathrm{diag}(\mathbf{y})\boldsymbol{\alpha}_*,$$
$$[\boldsymbol{\alpha}_*]_i = \nabla\ell\left(y_i\mathbf{x}_i^\top\mathbf{w}_*\right), \; i = 1, \ldots, n.$$

Observation 1

We can construct $\mathbf{w}_* \in \mathbb{R}^d$ from $\boldsymbol{\alpha}_* \in \mathbb{R}^n$, and vice versa.

Primal and Dual Problems after Random Projection

√ The primal problem

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

where $\widehat{\mathbf{x}}_i = \frac{1}{\sqrt{m}}\mathbf{R}^\top \mathbf{x}_i$.

Primal and Dual Problems after Random Projection

- The primal problem

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

where $\widehat{\mathbf{x}}_i = \frac{1}{\sqrt{m}}\mathbf{R}^\top \mathbf{x}_i$.

- The dual problem

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

where $\widehat{\mathbf{G}} = \mathrm{diag}(\mathbf{y})\mathbf{X}^\top \left(\frac{\mathbf{R}\mathbf{R}^\top}{\mathrm{m}}\right)\mathbf{X}\mathrm{diag}(\mathbf{y})$.

Relation between Primal and Dual Solutions

Proposition 2

Let $\mathbf{z}_* \in \mathbb{R}^m$ be the optimal primal solution to (P2), and $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$ be the optimal dual solution to (D2). We have

$$\mathbf{z}_* = -\frac{1}{\lambda}\frac{1}{\sqrt{m}}\mathbf{R}^\top \mathbf{X}\mathrm{diag}(\mathbf{y})\widehat{\boldsymbol{\alpha}}_*,$$

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

Observation 2

We can construct $\mathbf{z}_* \in \mathbb{R}^m$ from $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$, and vice versa.

Relations between Dual Solutions

√ The first dual problem

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

where $\mathbf{G} = \mathrm{diag}(\mathbf{y})\ \mathbf{X}^\top \mathbf{X}\ \mathrm{diag}(\mathbf{y})$.

Relations between Dual Solutions

√ The first dual problem

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

where $\mathbf{G} = \mathrm{diag}(\mathbf{y})\ \mathbf{X}^\top\mathbf{X}\ \mathrm{diag}(\mathbf{y})$.

√ The second dual problem

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

where $\widehat{\mathbf{G}} = \mathrm{diag}(\mathbf{y})\ \mathbf{X}^\top\left(\frac{\mathbf{R}\mathbf{R}^\top}{\mathrm{m}}\right)\mathbf{X}\ \mathrm{diag}(\mathbf{y})$.

Relations between Dual Solutions

√ The first dual problem

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

where $\mathbf{G} = \mathrm{diag}(\mathbf{y})\ \mathbf{X}^\top\mathbf{X}\ \mathrm{diag}(\mathbf{y})$.

√ The second dual problem

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

where $\widehat{\mathbf{G}} = \mathrm{diag}(\mathbf{y})\ \mathbf{X}^\top\left(\frac{\mathbf{R}\mathbf{R}^\top}{\mathrm{m}}\right)\mathbf{X}\ \mathrm{diag}(\mathbf{y})$.

Observation 3

We can approximate $\boldsymbol{\alpha}_*$ by $\widehat{\boldsymbol{\alpha}}_*$, when m is large enough.

Relations between Dual Solutions

√ The first dual problem

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

where $\mathbf{G} = \mathrm{diag}(\mathbf{y})\ \mathbf{X}^\top\mathbf{X}\ \mathrm{diag}(\mathbf{y})$.

√ The second dual problem

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

where $\widehat{\mathbf{G}} = \mathrm{diag}(\mathbf{y})\ \mathbf{X}^\top \left(\frac{\mathbf{R}\mathbf{R}^\top}{\mathrm{m}}\right) \mathbf{X}\ \mathrm{diag}(\mathbf{y})$.

Observation 3

We can approximate $\boldsymbol{\alpha}_*$ by $\widehat{\boldsymbol{\alpha}}_*$, when m is large enough.

The expectation

$$\mathbb{E}\left[\widehat{\mathbf{G}}\right] = \mathrm{diag}(\mathbf{y})\mathrm{X}^\top\mathbb{E}\left[\frac{\mathbf{R}\mathbf{R}^\top}{\mathrm{m}}\right]\mathbf{X}\mathrm{diag}(\mathbf{y}) = \mathrm{diag}(\mathbf{y})\mathbf{X}^\top\ \mathbf{I}\ \mathbf{X}\mathrm{diag}(\mathbf{y}) = \mathrm{G}$$

Putting Everything Together

Observations

We can construct $\mathbf{w}_* \in \mathbb{R}^d$ from $\boldsymbol{\alpha}_* \in \mathbb{R}^n$, and vice versa.
We can construct $\mathbf{z}_* \in \mathbb{R}^m$ from $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$, and vice versa.
We can approximate $\boldsymbol{\alpha}_*$ by $\widehat{\boldsymbol{\alpha}}_*$, when m is large enough.

Putting Everything Together

Observations

We can construct $\mathbf{w}_* \in \mathbb{R}^d$ from $\boldsymbol{\alpha}_* \in \mathbb{R}^n$, and vice versa.
We can construct $\mathbf{z}_* \in \mathbb{R}^m$ from $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$, and vice versa.
We can approximate $\boldsymbol{\alpha}_*$ by $\widehat{\boldsymbol{\alpha}}_*$, when m is large enough.

The main idea

1. Construct $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$ from $\mathbf{z}_* \in \mathbb{R}^m$
2. Use $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$ to approximate $\boldsymbol{\alpha}_* \in \mathbb{R}^n$
3. Construct $\mathbf{w}_* \in \mathbb{R}^d$ from $\boldsymbol{\alpha}_* \in \mathbb{R}^n$

The Proposed Algorithm

1: **Input:** input patterns $\mathbf{X} \in \mathbb{R}^{d \times n}$, binary class assignment $\mathbf{y} \in \{-1, +1\}^n$, and sample size m

2: Sample a Gaussian random matrix $\mathbf{R} \in \mathbb{R}^{d \times m}$

3: Compute $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n] = \mathbf{R}^\top \mathbf{X}/\sqrt{m}$

The Proposed Algorithm

1: **Input:** input patterns $\mathbf{X} \in \mathbb{R}^{d \times n}$, binary class assignment $\mathbf{y} \in \{-1, +1\}^n$, and sample size m

2: Sample a Gaussian random matrix $\mathbf{R} \in \mathbb{R}^{d \times m}$

3: Compute $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n] = \mathbf{R}^\top \mathbf{X} / \sqrt{m}$

4: Obtain the primal solution $\mathbf{z}_* \in \mathbb{R}^m$ by solving

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

The Proposed Algorithm

1: **Input:** input patterns $\mathbf{X} \in \mathbb{R}^{d\times n}$, binary class assignment $\mathbf{y} \in \{-1,+1\}^n$, and sample size m
2: Sample a Gaussian random matrix $\mathbf{R} \in \mathbb{R}^{d\times m}$
3: Compute $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1,\ldots,\widehat{\mathbf{x}}_n] = \mathbf{R}^\top\mathbf{X}/\sqrt{m}$
4: Obtain the primal solution $\mathbf{z}_* \in \mathbb{R}^m$ by solving

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

5: Construct the dual solution $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$ by

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

The Proposed Algorithm

1: **Input:** input patterns $\mathbf{X} \in \mathbb{R}^{d \times n}$, binary class assignment $\mathbf{y} \in \{-1, +1\}^n$, and sample size m
2: Sample a Gaussian random matrix $\mathbf{R} \in \mathbb{R}^{d \times m}$
3: Compute $\widehat{\mathbf{X}} = [\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n] = \mathbf{R}^\top \mathbf{X}/\sqrt{m}$
4: Obtain the primal solution $\mathbf{z}_* \in \mathbb{R}^m$ by solving

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

5: Construct the dual solution $\widehat{\boldsymbol{\alpha}}_* \in \mathbb{R}^n$ by

$$[\widehat{\boldsymbol{\alpha}}_*]_i = \nabla \ell\left(\frac{y_i}{\sqrt{m}} \mathbf{x}_i^\top \mathbf{R} \mathbf{z}_*\right), \; i = 1, \ldots, n$$

6: Compute $\widetilde{\mathbf{w}} \in \mathbb{R}^d$ by

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

7: **Output:** the recovered solution $\widetilde{\mathbf{w}}$

The Key Difference

- The naive solution

$$\widehat{\mathbf{w}} \propto \mathbf{R}\mathbf{z}_*$$

- The recovered solution by DRP

$$\widetilde{\mathbf{w}} \propto \mathbf{X}(\widehat{\boldsymbol{\alpha}}_* \circ \mathbf{y})$$

Example: Square Loss

- The square loss

$$\ell(z) = \frac{1}{2}(1 - z)^2$$

- The low-dimensional optimization problem

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

- The optimal solution

$$\mathbf{z}_* = \left(\lambda \mathbf{I} + \widehat{\mathbf{X}}\widehat{\mathbf{X}}^\top\right)^{-1} \widehat{\mathbf{X}}\mathbf{y}$$

- The dual solution

$$\widehat{\boldsymbol{\alpha}}_* = \mathrm{diag}(\mathbf{y})\widehat{\mathbf{X}}^\top \mathbf{z}_* - \mathbf{1}$$

- The recovered solution

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

Example: Square Loss

- The original optimization problem

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

- The optimal solution

$$\mathbf{w}_* = \left(\lambda \mathbf{I} + \mathbf{X}\mathbf{X}^\top\right)^{-1}\mathbf{X}\mathbf{y} = \mathbf{X}\left(\lambda \mathbf{I} + \mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{y}$$

- The recovered solution by DRP

$$\widetilde{\mathbf{w}} = \mathbf{X}\left(\lambda \mathbf{I} + \mathbf{X}^\top \frac{\mathbf{R}\mathbf{R}^\top}{m}\mathbf{X}\right)^{-1}\mathbf{y}$$

$m = \Omega(r \log r)$ is required to ensure $\mathbf{X}^\top \frac{\mathbf{R}\mathbf{R}^\top}{m}\mathbf{X} \approx \mathbf{X}^\top\mathbf{X}$

Example: Square Loss

√ The original optimization problem

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

√ The optimal solution

$$\mathbf{w}_* = \left(\lambda \mathbf{I} + \mathbf{X}\mathbf{X}^\top\right)^{-1}\mathbf{X}\mathbf{y} = \mathbf{X}\left(\lambda\mathbf{I} + \mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{y}$$

√ The recovered solution by DRP

$$\widetilde{\mathbf{w}} = \mathbf{X}\left(\lambda\mathbf{I} + \mathbf{X}^\top\frac{\mathbf{R}\mathbf{R}^\top}{m}\mathbf{X}\right)^{-1}\mathbf{y}$$

$m = \Omega(r \log r)$ is required to ensure $\mathbf{X}^\top\frac{\mathbf{R}\mathbf{R}^\top}{m}\mathbf{X} \approx \mathbf{X}^\top\mathbf{X}$

√ The naive solution

$$\widehat{\mathbf{w}} = \frac{\mathbf{R}\mathbf{R}^\top}{m}\mathbf{X}\left(\lambda\mathbf{I} + \mathbf{X}^\top\frac{\mathbf{R}\mathbf{R}^\top}{m}\mathbf{X}\right)^{-1}\mathbf{y}$$

$m = \Omega(d \log d)$ is required to ensure $\frac{\mathbf{R}\mathbf{R}^\top}{m} \approx \mathbf{I}$

Outline

Motivation

- Suspose the reconstruction error satisfying

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

  with a small $\epsilon \leq 1$, i.e., multiplicative NOT additive!

- Applying dual random projection to recover

$$\Delta\mathbf{w} = \mathbf{w}_* - \widetilde{\mathbf{w}},$$

  the reconstruction error will be

$$\epsilon\|\Delta\mathbf{w}\|_2 \leq \epsilon^2\|\mathbf{w}_*\|_2$$

Motivation

√ Suppose the reconstruction error satisfying

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

with a small $\epsilon \leq 1$, i.e., multiplicative NOT additive!

√ Applying dual random projection to recover

$$\Delta \mathbf{w} = \mathbf{w}_* - \widetilde{\mathbf{w}},$$

the reconstruction error will be

$$\epsilon \|\Delta \mathbf{w}\|_2 \leq \epsilon^2 \|\mathbf{w}_*\|_2$$

Implication

We can reduce the reconstruction error *exponentially* by running Dual Random Projection *iteratively*.

An Iterative Extension

1: Sample a Gaussian random matrix $\mathbf{R} \in \mathbb{R}^{d \times m}$

2: Compute $\widehat{\mathbf{X}} = \mathbf{R}^\top \mathbf{X}/\sqrt{m}$

3: Initialize $\widetilde{\mathbf{w}}^0 = \mathbf{0}$

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

5: Obtain $\mathbf{z}_*^t \in \mathbb{R}^m$ by solving the following optimization problem

$$\min_{\mathbf{z} \in \mathbb{R}^m} \frac{\lambda}{2} \left\| \mathbf{z} + \frac{1}{\sqrt{m}} \mathbf{R}^\top \widetilde{\mathbf{w}}^{t-1} \right\|_2^2 + \sum_{i=1}^n \ell \left(y_i \mathbf{z}^\top \widehat{\mathbf{x}}_i + y_i [\widetilde{\mathbf{w}}^{t-1}]^\top \mathbf{x}_i \right)$$

6: Construct the dual solution $\widehat{\boldsymbol{\alpha}}_*^t \in \mathbb{R}^n$ using

$$[\widehat{\boldsymbol{\alpha}}_*^t]_i = \nabla \ell \left(y_i \widehat{\mathbf{x}}_i^\top \mathbf{z}_*^t + y_i [\widetilde{\mathbf{w}}^{t-1}]^\top \mathbf{x}_i \right), \; i = 1, \ldots, n$$

7: Update the solution by $\widetilde{\mathbf{w}}^t = -\mathbf{X} \mathrm{diag}(\mathbf{y}) \widehat{\boldsymbol{\alpha}}_*^{\mathrm{t}} / \lambda$

8: **end for**

9: **Output** the recovered solution $\widetilde{\mathbf{w}}^T$

- Note that the random projections is applied only once!

Outline

The Reconstruction Error

√ We denote by r the rank of matrix $\mathbf{X}$, and assume $r \ll \min(d, n)$.

Theorem 4

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

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

provided

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

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

The Reconstruction Error

√ We denote by r the rank of matrix $\mathbf{X}$, and assume $r \ll \min(d,n)$.

Theorem 4

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

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

provided

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

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

Implication

To accurately recover $\mathbf{w}_*$, when number of required random projections is $\Omega(r \log r)$, we have:

$$\|\widetilde{\mathbf{w}} - \mathbf{w}_*\|_2 \leq O\left(\sqrt{\frac{r}{m}}\right)\|\mathbf{w}_*\|_2$$

The Sketch of the Proof

√ Show

$$\widehat{\mathbf{G}} = \mathrm{diag}(\mathbf{y})\mathbf{X}^\top \left(\frac{\mathbf{R}\mathbf{R}^\top}{\mathrm{m}}\mathbf{X}\right)\mathrm{diag}(\mathbf{y}) \approx \mathrm{diag}(\mathbf{y})\mathbf{X}^\top\mathbf{X}\mathrm{diag}(\mathbf{y}) = \mathbf{G}$$

A concentration inequality

Let $\mathbf{A} \in \mathbb{R}^{r\times m}$ be a standard Gaussian random matrix. For any $0 < \varepsilon \leq 1/2$, with a probability at least $1-\delta$, we have

$$\left\|\frac{1}{m}\mathbf{A}\mathbf{A}^\top - \mathbf{I}\right\|_2 \leq \varepsilon,$$

provided

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

where c is a constant whose value is at least $1/4$.

which implies $\widehat{\boldsymbol{\alpha}}_* \approx \boldsymbol{\alpha}_*$

√ Show

$$\widetilde{\mathbf{w}} = -\frac{1}{\lambda}\mathbf{X}\mathrm{diag}(\mathbf{y})\widehat{\boldsymbol{\alpha}}_* \approx -\frac{1}{\lambda}\mathbf{X}\mathrm{diag}(\mathbf{y})\boldsymbol{\alpha}_* = \mathbf{w}_*$$

The Reconstruction Error of the Iterative Algorithm

Theorem 5

Let $\widetilde{\mathbf{w}}^T$ be the solution recovered after T iterations. For any $0 < \varepsilon < 1/2$, with a probability at least $1 - \delta$, we have

$$\|\widetilde{\mathbf{w}}^T - \mathbf{w}_*\|_2 \leq \left(\frac{\varepsilon}{1-\varepsilon}\right)^T \|\mathbf{w}_*\|_2,$$

provided

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

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

The Reconstruction Error of the Iterative Algorithm

Theorem 5

Let $\widetilde{\mathbf{w}}^T$ be the solution recovered after T iterations. For any $0 < \varepsilon < 1/2$, with a probability at least $1 - \delta$, we have

$$\|\widetilde{\mathbf{w}}^T - \mathbf{w}_*\|_2 \leq \left(\frac{\varepsilon}{1-\varepsilon}\right)^T \|\mathbf{w}_*\|_2,$$

provided

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

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

Implication

We can recover the optimal solution with a relative error ϵ, i.e.,

$$\|\mathbf{w}_* - \widetilde{\mathbf{w}}^T\|_2 \leq \epsilon\|\mathbf{w}_*\|_2$$

by using $\log_{(1-\varepsilon)/\varepsilon} 1/\epsilon$ iterations.

Outline

The Reconstruction Error

Theorem 6

Assume $\mathbf{w}_*$ *lies in the subspace spanned by the first* k *left singular vectors of* $\mathbf{X}$*, and the loss* $\ell(\cdot)$ *is* γ*-smooth. For any* $0 < \varepsilon \leq 1$*, with a probability at least* $1 - \delta$*, we have*

$$\|\widetilde{\mathbf{w}} - \mathbf{w}_*\|_2 \leq \frac{\varepsilon}{1-\varepsilon}\left(1 + \frac{\sqrt{\lambda}}{\sqrt{\gamma}\sigma_k}\right)\|\mathbf{w}_*\|_2,$$

provided

$$m \geq \frac{\bar{r}\sigma_1^2}{c\varepsilon^2(\lambda/\gamma + \sigma_1^2)}\log\frac{2d}{\delta},$$

where σ_i *is the* i*-th singular value of* $\mathbf{X}$*,* $\bar{r} = \sum_{i=1}^{d}\frac{\sigma_i^2}{\lambda/\gamma+\sigma_i^2}$*, and the constant* c *is at least* $1/32$*.*

Discussions

- √ Similar to the low-rank case, the number of required random projections is $\Omega(\bar{r}\log d)$

$$\bar{r} = \sum_{i=1}^{d} \frac{\sigma_i^2}{\lambda/\gamma + \sigma_i^2}$$

- √ The number $\bar{r}$ is closely related to the numerical $\sqrt{\frac{\lambda}{\gamma}}$-rank of $\mathbf{X}$ [?].

  - $\mathbf{X}$ has numerical ν-rank r_ν if

$$\sigma_{r_\nu} > \nu \geq \sigma_{r_\nu+1}.$$

  - Using the notation of numerical rank, we have

$$\bar{r} \leq r_{\sqrt{\lambda/\gamma}} + \sum_{i=r_{\sqrt{\lambda/\gamma}}+1}^{d} \frac{\sigma_i^2}{\lambda/\gamma + \sigma_i^2} = O(r_{\sqrt{\lambda/\gamma}})$$

  if $\sigma_i \ll \sqrt{\lambda/\gamma}$ for $i > r_{\sqrt{\lambda/\gamma}}$.

Outline

Reconstruction Error for Sparse Solutions

- Recovery error when the optimal solution is known to be sparse i.e., $\|\mathbf{w}_*\|_0 \leq s$

Theorem 7

When the optimal solution $\mathbf{w}_*$ is s-sparse, we can bound the error by

$$\|\widetilde{\mathbf{w}} - \mathbf{w}_*\|_2 \leq O\left(\sqrt{\frac{r}{m}}\right)\|\mathbf{w}_*\|_2$$

provided

$$m \geq O(s \log d).$$

- It can be generalized to cases where $\mathbf{w}_*$ is approximately sparse

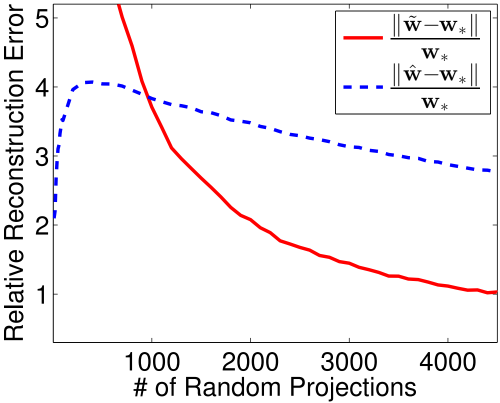
Outline

A Preliminary Empirical Study

√ rcv1.binary from Libsvm

- 20,242 samples and 47,236 features
- Half for training, and half for testing

√ The reconstruction error

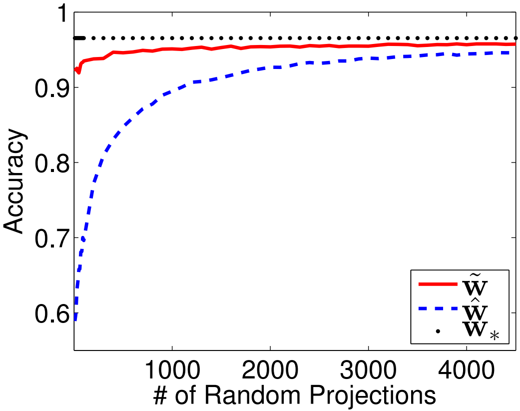


A Preliminary Empirical Study

√ rcv1.binary from Libsvm

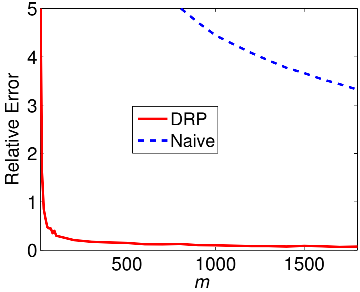
- 20,242 samples and 47,236 features
- Half for training, and half for testing

√ Accuracy



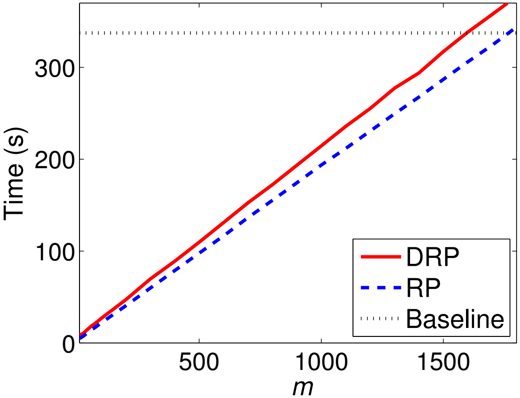
Relative Error Recovery

- Synthetic data set
  - √ $d = 20,000$
  - √ $n = 50,000$
  - √ $r = 10$
  - √ $\ell(x) = \ln(1 + \exp(-x))\,,\ \lambda = 1/n$



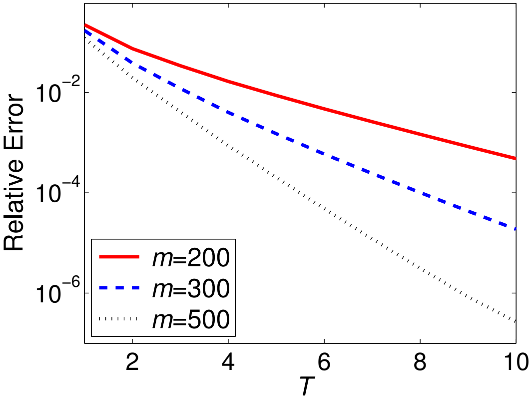
- Relative errore: $\frac{\|\tilde{\mathbf{w}}_* - \mathbf{w}_*\|}{\|\mathbf{w}_*\|}$ and $\frac{\|\hat{\mathbf{w}}_* - \mathbf{w}_*\|}{\|\mathbf{w}_*\|}$

Running Time

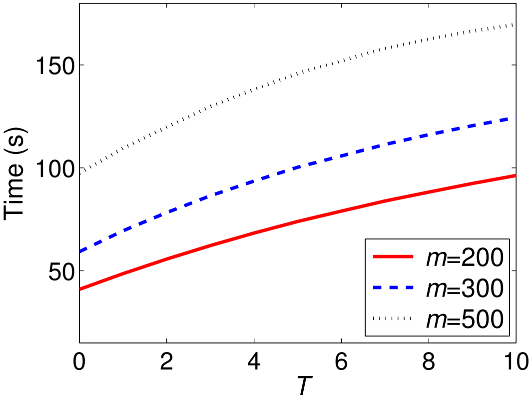


- Majority of running time of DRP is spent on random projection!

Relative Error of Iterative Extension



Running Time of Iterative Extension



Conclusion

√ We considered a novel problem for classification

- Recovering the optimal solution to the high-dimensional optimization problem based on solution obtained from random projection

Conclusion

√ We considered a novel problem for classification

- Recovering the optimal solution to the high-dimensional optimization problem based on solution obtained from random projection

√ We proposed a simple algorithm, named **Dual Random Projection**

- Using the dual solution of the low-dimensional optimization problem to recover the optimal solution

Conclusion

√ We considered a novel problem for classification

- Recovering the optimal solution to the high-dimensional optimization problem based on solution obtained from random projection

√ We proposed a simple algorithm, named **Dual Random Projection**

- Using the dual solution of the low-dimensional optimization problem to recover the optimal solution

√ Theoretical analysis shows that

- When $\mathbf{X}$ is of low rank, we can recover the optimal solution by using $\Omega(r \log r)$ projections
- A similar result also holds when the data matrix can be well approximated by a low rank matrix.
- Also we show the recovery error when $\mathbf{w}_*$ is sparse

Conclusion

√ We considered a novel problem for classification

- Recovering the optimal solution to the high-dimensional optimization problem based on solution obtained from random projection

√ We proposed a simple algorithm, named **Dual Random Projection**

- Using the dual solution of the low-dimensional optimization problem to recover the optimal solution

√ Theoretical analysis shows that

- When $\mathbf{X}$ is of low rank, we can recover the optimal solution by using $\Omega(r \log r)$ projections
- A similar result also holds when the data matrix can be well approximated by a low rank matrix.
- Also we show the recovery error when $\mathbf{w}_*$ is sparse

√ Empirical results demonstrate the merits of proposed algorithm


Thank you