Simple Optimization, Bigger Models, and Faster Learning

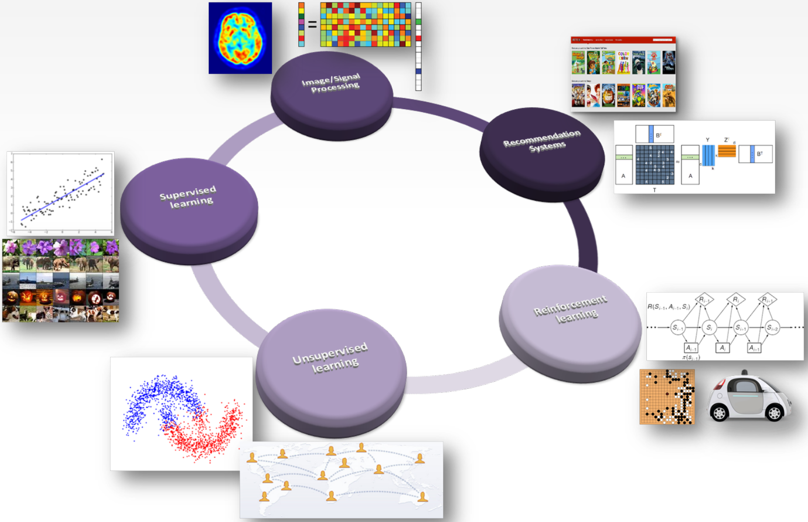
Niao He

Industrial and Enterprise Systems Engineering

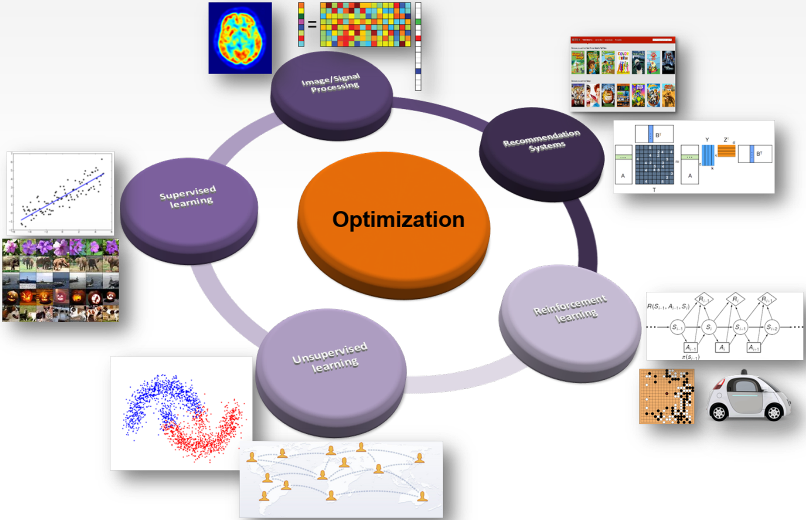
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Big Data Symposium, UIUC, 2016

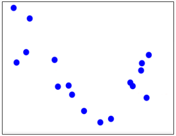
Big Data, Big Picture



Big Data, Big Picture

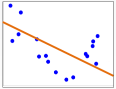


Big Data, Big Picture

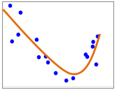


$(x, y), y \sim f(x)$

- Linear model: $f(x) = w^T x$



- Nonlinear model: $f(x) = w^T \phi(x)$



- Multi-layer network model: $f(x) = W_3^T g_2(W_2^T g_1(W_1^T x))$

$$\min_{f \in \mathcal{F}} L(f)$$

$$e.g., L(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

Data

Model

Optimization

Big Data, Big Opportunity

- Most problems are convex!
  - Local optimum is also global optimum; fundamentally tractable.
- Some problems are non-convex, but admits nice structure.
  - Local optimum "behaves like" global optimum.
- We have a dedicated library of efficient first-order optimization algorithms.
  - Gradient Descent, its acceleration and cousins
  - Conditional Gradient (a.k.a. Frank Wolfe algorithm)
  - Coordinate Descent, its randomization variations
  - Primal-dual algorithms, ADMM
  - Quasi-Newton Methods

Big Data, Big Challenges

- Too many data points (n large): simpler algorithms are needed
  - Stochastic gradient descent (SGD, a.k.a. stochastic approximation) type of algorithms become the only method of choice
  - Cheap iteration cost and (at least) sublinear convergence guarantee

- Too many features (d large): bigger models are needed
  However,
  - Kernel methods are usually not scalable
  - Neural network models break convexity

Revisit: Stochastic Optimization and SGD

SGD – Overview

(Stochastic) convex optimization problem

$$\min_{\theta \in \Theta} \phi(\theta) = \mathbb{E}_{\xi}[F(\theta,\xi)] \quad \text{or} \quad \frac{1}{n}\sum_{i=1}^{n} F(\theta,\xi_i)$$

- Stochastic Gradient Descent [Robbins-Monro, 1951]

$$\theta_{t+1} = \Pi_{\Theta}\left(\theta_t - \gamma_t \nabla F(\theta_t,\xi_t)\right)$$

where $\Pi_{\Theta}(\eta) = \mathrm{Argmin}_{\theta\in\Theta}\left\{\frac{1}{2}\|\theta-\eta\|_2^2\right\}$.

- Stochastic Mirror Descent [Nemirovski, 1979]

$$\theta_{t+1} = P_{\theta_t}\left(\gamma_t \nabla F(\theta_t,\xi_t)\right)$$

where $P_{\theta_t}(\eta) = \mathrm{Argmin}_{\theta\in\Theta}\left\{D_{\omega}(\theta,\theta_t) + \langle \eta,\theta\rangle\right\}$.

- Inexact Stochastic Mirror Descent

$$\theta_{t+1} \in P_{\theta_t}^{\epsilon_t}\left(\gamma_t \nabla F(\theta_t,\xi_t)\right)$$

where $P_{\theta_t}^{\epsilon_t}(\eta) = \mathrm{Argmin}_{\theta\in\Theta}^{\epsilon_t}\left\{D_{\omega}(\theta,\theta_t) + \langle \eta,\theta\rangle\right\}$.

SGD – Typical Results

$$\min_{\theta \in \Theta} \phi(\theta) = \mathbb{E}_\xi[F(\theta, \xi)]$$

The inexact Stochastic Mirror Descent algorithm guarantees that

$$\mathbb{E}\left[\phi\left(\frac{\sum_{\tau=1}^t \gamma_\tau \theta_\tau}{\sum_{\tau=1}^t \gamma_\tau}\right) - \phi(\theta_*)\right] \leq \frac{M^2 \sum_{\tau=1}^t \gamma_\tau^2 + D_\omega(\theta_*, \theta_1) + \sum_{\tau=1}^t \epsilon_\tau}{\sum_{\tau=1}^t \gamma_\tau}$$

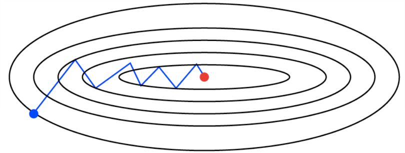
where $M^2 = \max_{\theta \in \Theta} \mathbb{E}[\|\nabla F(\theta, \xi)\|_*^2]$.

unbiased gradient + bounded variance
+ proper stepsize + well-controlled error
+ good average scheme

- $O(1/t)$ convergence rate for strongly convex case
- $O(1/\sqrt{t})$ convergence rate for general convex case

SGD – Practical Performance

- **Full Gradient Descent:**
  converges faster but with expensive iteration cost



- **Stochastic Gradient Descent:**
  converges slowly but with cheaper iteration cost

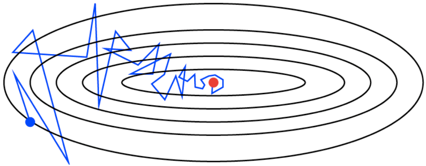


Figure from [Bach,2013]

SGD – Beyond

- Lots of recent algorithmic development for supervised learning:
  - SGD for convex-concave saddle point problems
  - SGD with adaptive learning rates / preconditioning (AdaGrad, etc.)
  - SGD with importance / stratified sampling (Iprox-SMD, etc.)
  - SGD with second order information (SQN, stochastic BFGS, etc.)
  - Variance reduced algorithms (SAG, SAGA, SVRG, PRDG, etc.)
  - Parallel and asynchronous SGD (Hogwild!, Downpour SGD, etc.)

- However, still for several fundamental machine learning tasks, SGD or any of the above adaptation is not enough.

Three Variants of Stochastic Gradient Descent

- Supervised Learning:
  - Doubly Stochastic Gradient Descent (Doubly SGD)
    [with Dai, Xie, Liang, Balcan, Song, NIPS'14]
- Bayesian Inference:
  - Particle Mirror Descent (PMD)
    [with Dai^2 and Song, AISTATS'15]
- Reinforcement Learning:
  - Embedding Stochastic Gradient Descent (Embedding-SGD)
    [with Dai, Pan, and Song, 2016]

Doubly SGD: scaling up big nonlinear models

Learning in Hilbert Space

$$\min_{f \in \mathcal{H}} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i) + \nu \|f\|_{\mathcal{H}}^2$$

or

$$\min_{f \in \mathcal{H}} L(f) := \underbrace{\mathbb{E}_{(x,y)\sim\mathbb{P}(x,y)}[\ell(f(x), y)]}_{\textbf{expected loss}} + \underbrace{\frac{\nu}{2}\|f\|_{\mathcal{H}}^2}_{\textbf{regularizer}}$$

with domain $\mathcal{H}$ as the *reproducing kernel Hilbert space:*

- generators: $k(x, \cdot), \forall x \in \mathcal{X}$
- reproducing property: $\langle f, k(x, \cdot)\rangle_{\mathcal{H}} = f(x)$.

Previous Work

- Dual approach: e.g. for square loss

$$\min_{\alpha \in \mathbf{R}^n} \alpha^T K \alpha + \lambda \alpha^T \alpha - 2\alpha^T y$$

  – inpractical to store/compute kernel matrix $K = (k(x_i, x_j))$.

- Stochastic Gradient Descent/Dual Coordinate Ascent

  [Kivinen et.al.,2004; Shalev-Shwartz & Zhang, 2013]

  – at step t, $f_t(x) = \sum_{i=1}^{t} \alpha_i k(x_i, \cdot)$

  – require high memory to retrieve support vectors

- Low-rank approximation/Random feature approximation

  [Williams & Seeger, 2001; Rahimi & Rechet, 2008]

  – low memory, but does not generalize well

Duality Between Kernels and Random Processes

Theorem (Bochner)

A continuous kernel $k(x, x') = k(x - x')$ on $\mathbf{R}^d$ is PD if and only if $k(x - x')$ is the Fourier transform of a non-negative measure $\mathbb{P}(\omega)$.

$$k(x, x') = \int_{\mathbf{R}^d} e^{i\omega^\top (x - x')} \, d\mathbb{P}(\omega) = \mathbb{E}_\omega[\phi_\omega(x)\phi_\omega(x')].$$

Examples

Kernel	$k(x, x')$	$p(\omega)$
Gaussian	$\exp(-\frac{\|x-x'\|_2^2}{2})$	$2\pi^{-\frac{d}{2}} \exp(-\frac{\|\omega\|_2^2}{2})$
Laplacian	$\exp(-\|x - x'\|_1)$	$\prod_{i=1}^d \frac{1}{\pi(1+\omega_i^2)}$
Cauchy	$\prod_{i=1}^d \frac{2}{1+(x_i-x_i')^2}$	$\exp(-\|\omega\|_1)$

many other kernels (dot product, polynomial, Hellinger's, χ^2, Arc-cosine).

Doubly SGD: Basic Idea

$$\min_{f\in\mathcal{H}} \quad \mathbb{E}_{(x,y)\in\mathbb{P}(x,y)}[\ell(f(x),y)] + \nu\|f\|^2_{\mathcal{H}}$$

- First randomly sample $(x,y) \sim \mathbb{P}(x,y) \Rightarrow$ stochastic gradient

$$g(\cdot) = \ell'(f(x),y)k(x,\cdot) + \nu f(\cdot)$$

- Then randomly sample $\omega \sim \mathbb{P}(\omega) \Rightarrow$ doubly stochastic gradient

$$\widehat{g}(\cdot) = \ell'(f(x),y)\phi_\omega(x)\phi_\omega(\cdot) + \nu f(\cdot)$$

  - double sources of randomness:
  - unbiased: $\mathbb{E}_{x,y,\omega}[\widehat{g}(\cdot)] = \nabla R(f)$

- Observation: $f_t(\cdot) = \sum_{i=1}^t \beta_i k(x_i,\cdot) \Longrightarrow f_t(\cdot) = \sum_{i=1}^t \alpha_i \phi_{\omega_i}(\cdot)$
  Memory significantly reduced from $O(td)$ to $O(t)$

- Caveat: no longer in $\mathcal{H}$

Doubly SGD: Theoretical Complexity

Assumption: Loss function is smooth and kernel is bounded;

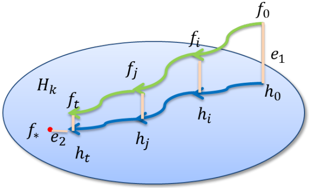
Theorem

When $\gamma_t = \frac{\theta}{t}$ with $\theta > 0$ such that $\theta\nu \in \mathbb{Z}_+$, $\forall x \in \mathcal{X}$,

$$|f_{t+1}(x) - f_*(x)|^2 \le \widetilde{O}\left(\frac{1}{t}\right), \quad \textit{with high probability}$$

High-level proof idea. Decompose the error into two terms

$$|f_{t+1}(x) - f_*(x)|^2 \leqslant 2 \underbrace{|f_{t+1}(x) - h_{t+1}(x)|^2}_{\text{error due to random features}} + 2\kappa \underbrace{\|h_{t+1} - f_*\|_{\mathcal{H}}^2}_{\text{error due to random data}}$$



Doubly SGD: Key Features

Doubly SGD: $\{\alpha_i\}_{i=1}^t$

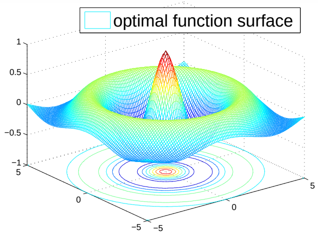
Input: $\mathbb{P}(\omega)$, $\phi_\omega(x)$, $\ell(f(x), y)$, ν.

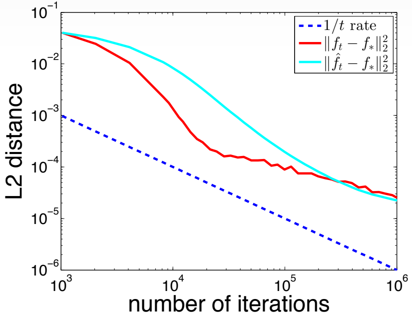
for $i = 1, \ldots, t$ **do**
  Sample $(x_i, y_i) \sim \mathbb{P}(x, y)$.
  Sample $\omega_i \sim \mathbb{P}(\omega)$ with seed i.
  $f(x_i) = \textbf{Predict}(x_i, \{\alpha_j\}_{j=1}^{i-1})$.
  $\alpha_i = -\gamma_i \ell'(f(x_i), y_i)\phi_{\omega_i}(x_i)$.
  $\alpha_j = (1 - \gamma_i \nu)\alpha_j$, $j = 1, \ldots, i-1$.
end for

- simple algorithm
- **flexible, nonparametric**
- **low memory cost**
  - $O(t)$ for doubly SGD
  - $O(n^2)$ for kernel matrix
  - $O(td)$ for vanilla SGD
- **cheap computation cost**
  – $O(td)$ at each iteration
- **theoretically grounded**
  – $O(1/t)$ w.h.p.
- **strong empirical results**
  – competes with neural nets

Toy Example

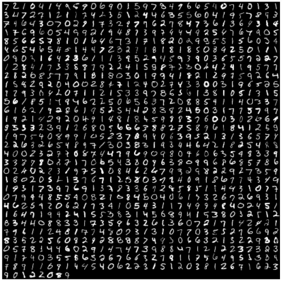
- Model: Kernelized Ridge Regression
- Dataset: 2D Synthetic dataset



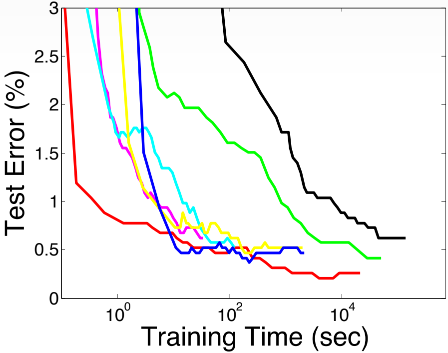


Handwritten Digit Recognition

- Model: Support Vector Machines
- Dataset: 1.6 million images for digit 6 and digit 8
- Input Dimension: each data point is of size 784

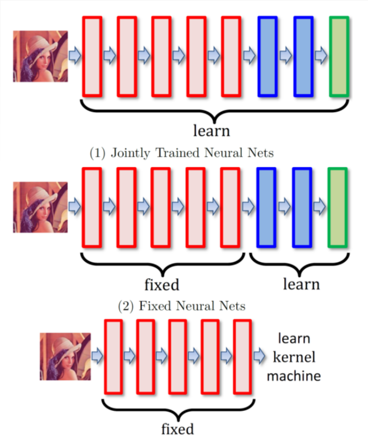


MNIST handwritten digits



doubly SGD/ SGD / SDCA / n-SDCA...

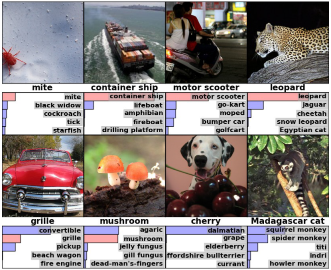
ImageNet Classification



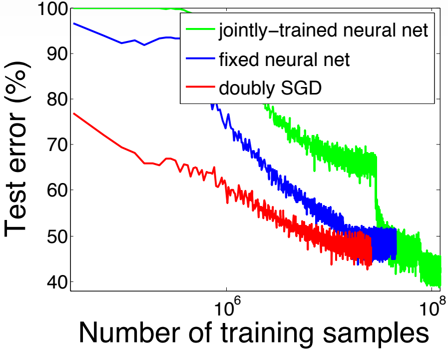
- Red layers are convolutions with max pooling layers.
- Blue layers are fully connected layers.
- Green layer is the output layer – multiclass logistic regression model.

ImageNet Classification

- Model: Logistic regression
- Dataset: 1.3 million color images and 1000 classes
- Input Dimension: each data point is of size 9216



ImageNet



Platform: AMD 16 2.4GHz Opteron CPUs and 200G memory

Extending to Min-Max Saddle Point Problems

Optimizing saddle point problems over RKHS:

$$\min_{f \in \mathcal{H}_k} \max_{g \in \mathcal{G}_{\widetilde{k}}} \quad \mathbb{E}_{x,y}[f(x)g(x) - \ell(g(x), y)] + \frac{\nu_1}{2}\|f\|_{\mathcal{H}}^2 - \frac{\nu_2}{2}\|g\|_{\mathcal{G}}^2$$

- The doubly SGD trick still applies.
- Under mild conditions (smooth loss and kernels), we have

$$\mathbb{E}[|f(x_t) - f_*(x)|^2 + |g(x_t) - g_*(x)|^2] \leq \widetilde{O}\left(\frac{1}{t}\right)$$

- Recently been applied to solve reinforcement learning problem.

Further Implication

Solving two-level stochastic optimization problems

$$\min_{\theta \in \Theta} \mathbb{E}_\xi[F_\xi(\mathbb{E}_\eta[G_\eta(\theta, \xi)])]$$

- The algorithm and analysis can be easily extended to address general stochastic problems involving two levels of expectations.

for $i = 1, \dots, t$ **do**
  Sample $(\xi_i, \eta_i) \sim \mathbb{P}(\xi, \eta)$.
  $\widehat{g} = \frac{1}{i} \sum_{j=1}^{i} G_{\eta_i}(\theta_i, \xi_j)$
  $\theta_{i+1} = P_{\theta_i}\left(\gamma_i \nabla G_{\eta_i}(\theta_i, \xi_i)^T \nabla F_{\xi_i}(\widehat{g})\right)$
end for

- When $\xi \perp\!\!\!\perp \eta$ and f Lipschitz smooth, g Lipschitz continuous, the overall function is strongly convex, then we obtain the "optimal" $O(1/t)$ rate of convergence.

Summary

- Optimization lies at the heart of Big Data analytics.
- Stochastic gradient descent is powerful, but has limitations.
- Simple optimization techniques allow us to learn bigger and faster.