Signal Processing for Big Data

G. B. Giannakis, K. Slavakis, and G. Mateos

Acknowledgments: NSF Grants EARS-1343248, EAGER-1343860
MURI Grant No. AFOSR FA9550-10-1-0567

Lisbon, Portugal
September 1, 2014
Big Data: A growing torrent

$600 to buy a disk drive that can store all of the world’s music

5 billion mobile phones in use in 2010

30 billion pieces of content shared on Facebook every month

40% projected growth in global data generated per year vs. 5% growth in global IT spending

Big Data: Capturing its value

$300 billion
potential annual value to US health care—more than double the total annual health care spending in Spain

€250 billion
potential annual value to Europe’s public sector administration—more than GDP of Greece

$600 billion
potential annual consumer surplus from using personal location data globally

60% potential increase in retailers’ operating margins possible with big data

Big Data and NetSci analytics

- Online social media
- Internet
- Clean energy and grid analytics
- Robot and sensor networks
- Biological networks
- Square kilometer array telescope

Desiderata: process, analyze, and learn from large pools of network data
Challenges

- Sheer **volume** of data
  - Distributed and parallel processing
  - Security and privacy concerns

- Modern massive datasets involve many **attributes**
  - Parsimonious models to ease interpretability
  - Enhanced predictive performance

- **Real-time** streaming data
  - Online processing
  - Quick-rough answer vs. slow-accurate answer?

- **Outliers** and **misses**
  - **Robust** imputation algorithms

- **Good news**: Ample research opportunities arise!
Opportunities

Theoretical and Statistical Foundations of Big Data Analytics

- Big tensor data models and factorizations
- High-dimensional statistical SP
- Network data visualization
- Resource tradeoffs
- Analysis of multi-relational data
- Common principles across networks

Algorithms and Implementation Platforms to Learn from Massive Datasets

- Scalable online, decentralized optimization
- Randomized algorithms
- Convergence and performance guarantees
- Information processing over graphs
- Novel architectures for large-scale data analytics
- Graph SP
- Robustness to outliers and missing data
SP-relevant Big Data themes

Roadmap

- Context and motivation
- Critical Big Data tasks
  - Encompassing and parsimonious data modeling
  - Dimensionality reduction, data visualization
  - Data cleansing, anomaly detection, and inference
- Optimization algorithms for Big Data
- Randomized learning
- Scalable computing platforms for Big Data
- Conclusions and future research directions
Encompassing model

\[
    Y = L + DS + V
\]

- **Observed data** \(\in \mathbb{R}^{D \times T}\)
- **Background (low rank)**
- **Patterns, innovations, (co-)clusters, outliers**
- **Noise**
- **Dictionary** \(\in \mathbb{R}^{D \times Q}\)
- **Sparse matrix** \(\in \mathbb{R}^{Q \times T}\)

- Subset \(\Omega \subset \{1, \ldots, D\} \times \{1, \ldots, T\}\) of observations and projection operator

\[
    [\mathcal{P}_\Omega(Y)]_{ij} = \begin{cases} 
    [Y]_{ij}, & \text{if } (i, j) \in \Omega \\
    0, & \text{o.w.}
    \end{cases}
\]

- Allow for misses

- Large-scale data \(D \gg \) and/or \(T \gg \)
- Any of \(\{L, D, S\}\) unknown
Subsumed paradigms

- **Structure leveraging criterion**

\[
\min_{\{L, D, S\}} \frac{1}{2} \|P_\Omega(Y - L - DS)\|_F^2 + \lambda_* \|L\|_* + \lambda_1 \|S\|_1
\]

- **Nuclear norm:** \(\|L\|_* := \sum_{j=1}^{\text{rank}(L)} \sigma_j(L)\)

- **\(\ell_1\)-norm:** \(\|S\|_1 := \sum_{q,t} |s_{q,t}|\)

(With or without misses)

- **\(L = 0, D\) known** \(\Rightarrow\) Compressive sampling (CS) [Candes-Tao ‘05]
- **\(L = 0\)** \(\Rightarrow\) Dictionary learning (DL) [Olshausen-Field ‘97]
- **\(L = 0, [D]_{ij} \geq 0, [S]_{ij} \geq 0\)** \(\Rightarrow\) Non-negative matrix factorization (NMF) [Lee-Seung ‘99]
- **\(D = I_D\)** \(\Rightarrow\) Principal component pursuit (PCP) [Candes et al ‘11]
- **\(S = 0, \text{rank}(L) \leq \rho\)** \(\Rightarrow\) Principal component analysis (PCA) [Pearson 1901]
PCA formulations

- Training data \( \{y_t \in \mathbb{R}^D\}_{t=1}^T \) \( \hat{C}_{yy} := (1/T) \sum_{t=1}^T y_t y_t^\top \)

- Minimum reconstruction error
  - Compression \( G \in \mathbb{R}^{d \times D} \) \( d \ll D \)
  - Reconstruction \( U \in \mathbb{R}^{D \times d} \)

\[
\min_{U,G} \sum_{t=1}^T \|y_t - UGy_t\|_2^2, \quad \text{s.to.} \quad U^\top U = I_d
\]

- Maximum variance

\[
\max_U \text{trace} [U^\top \hat{C}_{yy} U], \quad \text{s.to.} \quad U^\top U = I_d
\]

- Component analysis model \( y_t = U \psi_t + \varepsilon_t \)

\[
\min_{U,\psi_t} \sum_{t=1}^T \|y_t - U\psi_t\|_2^2, \quad \text{s.to.} \quad U^\top U = I_d
\]

Solution: \( \hat{U}_d = \text{d-evecs}(\hat{C}_{yy}) \), \( \hat{G} = \hat{U}_d^\top \), \( \hat{\psi}_t = \hat{U}_d^\top y_t \)
Dual and kernel PCA

- **SVD:**
  \[ Y_{D \times T} = U \Sigma V^\top \]
  \[ Y^\top Y = V \Sigma^2 V^\top \in \mathbb{R}^{T \times T} \quad O(DT^2) \]
  \[ YY^\top = U \Sigma^2 U^\top \in \mathbb{R}^{D \times D} \quad O(TD^2) \]
  
  Gram matrix

- **Mapping to “stretches” data geometry**
  \[ \hat{U}_d = Y \hat{V}_d \hat{\Sigma}_d^{-1} \]

- **Kernel (K)PCA; e.g., [Scholkopf-Smola’01]**
  - Mapping \( y_t \) to \( \varphi(y_t) \) “stretches” data geometry
    \[ e^y = [1, 1, 1/2!, \ldots][1, y, y^2, \ldots]^\top = c^\top \varphi(y) \]
  - Kernel trick \( \kappa(y_{t_i}, y_{t_j}) = \langle \varphi(y_{t_i}), \varphi(y_{t_j}) \rangle \)
    e.g., \( \kappa(y_{t_i}, y_{t_j}) = \exp[-\|y_{t_i} - y_{t_j}\|^2] \)

- **Kernel trick**
  \[ \hat{V}_d = d\text{-evecs}(K) \]
  \[ [K]_{ij} := \kappa(y_{t_i}, y_{t_j}) \]
  \[ [\kappa_t]_i := \kappa(y_{t_i}, y_t) \]

- **Inner products**
  \[ \hat{U}_d \varphi(y_t) = \hat{\Sigma}_d^{-1} \hat{V}_d \kappa_t \]

- **Kernel (K)PCA; e.g., [Scholkopf-Smola’01]**

Multi-dimensional scaling

“Given dissimilarities or “distances” \( \{\delta_{tt'}\}_{t,t'=1}^T \subset \mathbb{R}_{\geq 0} \), identify low-dim vectors \( \{\psi_t\}_{t=1}^T \subset \mathbb{R}^d \) that preserve \( \delta_{tt'} \)”

**LS or Kruskal-Shephard MDS:**

\[
\min_{\Psi=[\psi_1, \ldots, \psi_T] \in \mathbb{R}^{d \times T}} \sum_{t, t'=1}^T [\delta_{tt'} - \|\psi_t - \psi_{t'}\|]^2
\]

If \( \exists \{y_t\}_{t=1}^T \subset \mathbb{R}^D \) s.t. \( \delta_{tt'} = \|y_t - y_{t'}\| \)

**Classical MDS:**

\[
\min_{\Psi \in \mathbb{R}^{d \times T}} \|Y^\top Y - \Psi^\top \Psi\|_F^2
\]

**Solution** Dual PCA:

\[
\hat{V}_d = d\text{-evecs}(Y^\top Y)
\]

\[
\begin{cases}
Y^\top Y = V\Lambda V^\top \\
\Psi_{\text{opt}} = \Lambda_d^{\frac{1}{2}} \hat{V}_d^\top
\end{cases}
\]

- Distance-preserving downscaling of data dimension

Local linear embedding

- For each $y_t$ find neighborhood $\{y_{t'}\}_{t' \in \mathcal{N}_t}$, e.g., k-nearest neighbors

- Weight matrix captures local affine relations

$$\min_{W := [w_1, \ldots, w_T] \in \mathbb{R}^{T \times T}} \sum_{t=1}^{T} \left( y_t - \sum_{t' \in \mathcal{N}_t} w_{t't} y_{t'} \right)^2$$

Sparse $\mathcal{N}_t$ and $W$ [Elhamifar-Vidal’11]

- Identify low-dimensional vectors preserving local geometry [Saul-Roweis’03]

$$\min_{\Psi := [\psi_1, \ldots, \psi_T] \in \mathbb{R}^{d \times T}} \left\{ \sum_{t=1}^{T} \left( \psi_t - \sum_{t'=1}^{T} w_{t't} \psi_{t'} \right)^2 = \text{trace} \left[ \Psi (I_T - W) (I_T - W)^\top \Psi^\top \right] \right\}$$

Solution: The rows of $\Psi$ are the $d$ minor, excluding $1_T$, evecs$[(I_T - W)(I_T - W)^\top]$}

Application to graph visualization

- Undirected graph $G(\mathcal{V}, \mathcal{E})$: 
  \[
  \begin{cases}
    \mathcal{V} & : \text{Nodes or vertices} \\
    \mathcal{E} & : \text{Edges or communicating pairs of nodes} \\
    \mathcal{N}_\nu & : \text{All nodes that communicate with node } \nu \in \mathcal{V} \\
    \text{(neighborhood)}
  \end{cases}
  \]

- Given centrality metrics $\{c_\nu\}_{\nu \in \mathcal{V}} \subset \mathbb{R}_{\geq 0}$, centrality-constrained (CC) LLE

- Identify weights or geometry
  \[
  \forall \nu \in \mathcal{V}, \quad \min_{w_\nu \in \mathbb{R}^{|\mathcal{N}_\nu|}} \left\| y_\nu - \sum_{\nu' \in \mathcal{N}_\nu} w_{\nu'\nu} y_{\nu'} \right\|^2
  \]
  \[
  \text{s.to } \left\| \sum_{\nu' \in \mathcal{N}_\nu} w_{\nu'\nu} y_{\nu'} \right\|^2 = \exp(-c_\nu)
  \]

- Visualize ($d = 2, 3$):
  \[
  \min_{\{\psi_\nu\}_{\nu \in \mathcal{V}} \subset \mathbb{R}^d} \sum_{\nu \in \mathcal{V}} \left\| \psi_\nu - \sum_{\nu' \in \mathcal{N}_\nu} w_{\nu'\nu} \psi_{\nu'} \right\|^2
  \]
  \[
  \text{s.to } \|\psi_\nu\|^2 = \exp(-c_\nu), \quad \forall \nu \in \mathcal{V}
  \]

- Centrality constraints, e.g., node degree

- Only corr. $\{y_\nu^\top y_{\nu'}\}_{\nu, \nu' \in \mathcal{V}}$ or dissimilarities $\delta_{\nu\nu'} := 1 - \frac{|y_\nu^\top y_{\nu'}|}{\|y_\nu\| \|y_{\nu'}\|}$ need be known

Visualizing the Gnutella network

Gnutella: Peer-to-peer file-sharing network
\[ |\mathcal{V}| = 26,518 \]
\[ |\mathcal{E}| = 65,369 \]

CC-LLE: centrality captured by node degree

Dictionary learning

- Solve for dictionary $D$ and sparse $S$: 
  \[
  \min_{D \in \mathcal{D}, S \in \mathbb{R}^{Q \times T}} \frac{1}{2} \| \mathcal{P}_{\Omega}(Y - DS) \|_F^2 + \lambda_1 \sum_{t=1}^{T} \| s_t \|_1 \]
  \[
  \mathcal{D} := \{ D = [d_1, \ldots, d_Q] : \| d_q \| \leq 1, \forall q \} \quad Q \geq D
  \]

\[
\begin{cases}
  S_{k+1} \in \arg \min_{S \in \mathbb{R}^{Q \times T}} \frac{1}{2} \| Y - D_k S \|_F^2 + \lambda_1 \| S \|_1 \\
  D_{k+1} \in \arg \min_{D \in \mathcal{D}} \| Y - DS_{k+1} \|_F^2 = \arg \min_{D \in \mathcal{D}} \mathcal{L}(D, S_{k+1})
\end{cases}
\]

- Alternating minimization; both $\mathcal{L}(D_k, \cdot)$ and $\mathcal{L}(\cdot, S_{k+1})$ are convex

- Special case of block coordinate descent methods (BCDMs) [Tseng ‘01]

- Under certain conditions, $(D_k, S_k)_{k=0}^{\infty}$ converges to a stationary point of $\mathcal{L}$
Joint DL-LLE paradigm

- Image inpainting by local-affine-geometry-preserving DL
- 50% missing values; PSNR $\simeq 33$ dB

Online dictionary learning

- Data $\{y_t\}_{t=1}^{\infty}$ arrive sequentially

$$
\begin{aligned}
  s_{t+1} &\in \arg \min_{s \in \mathbb{R}^Q} \frac{1}{2} \|y_t - D_t s\|^2 + \lambda_1 \|s\|_1 \\
  D_{t+1} &\in \arg \min_{D \in \mathcal{D}} \sum_{\tau=1}^{t+1} \|y_\tau - D s_\tau\|^2
\end{aligned}
$$

- Inpainting of 12-Mpixel damaged image

  **S1.** Learn dictionary from “clean” patches

  **S2.** Remove text by sparse coding from “clean” $D$

---

Union of subspaces and subspace clustering

- Parsimonious model for clustered data

\[ Y = [y_1, \ldots, y_T] \]

\[ y_t \in \bigcup_{k=1}^K S_k + v_t, \text{ where } \begin{cases} S_k : \text{lin. subspace} \\ v_t \in \mathbb{R}^D : \text{noise} \end{cases} \]

**Subspace clustering (SC)** Identify

- Bases for \( \{S_k\}_{k=1}^K \)
- Data-cluster associations

- SC as DL with block-sparse \( S \)

\[
\min_{D \in \mathbb{R}^{D \times Q}, S \in \mathbb{R}^{Q \times T}} \frac{1}{2} \| Y - DS \|_F^2 + \lambda_1 \sum_{t=1}^T \| s_t \|_2 \quad \text{with } Q \ll T
\]

- If \( K = 1 \) then SC \( \Rightarrow \) PCA

---

Modeling outliers

- Outlier variables \( \{o_t\}_{t=1}^T \) s.t. \( o_t \begin{cases} \neq 0_D, & y_t \text{ outlier} \\ = 0_D, & \text{otherwise} \end{cases} \)

\[
y_t = U\psi_t + o_t + v_t, \quad t = 1, \ldots, T
\]

- Nominal data obey \( y_t = U\psi_t + v_t \); outliers something else
- Linear regression [Fuchs’99], [Wright-Ma’10], [Giannakis et al’11]
- Both \( \{U, \psi_t\} \) and \( O := [o_1 \ldots o_T] \) unknown, under-determined

- \( O \) typically sparse!
Robustifying PCA

Natural (sparsity-leveraging) estimator

\[
\min_{U, \Psi, O} \|Y - U\Psi - O\|_F^2 + \lambda_1 \|O\|_{2,c}, \quad \text{s. to } U^\top U = I_d.
\]

- Tuning parameter \(\lambda_1\) controls sparsity in \(\hat{O} \equiv \text{number of outliers}\)

Q: Does (P1) yield robust estimates \(\{\hat{U}, \hat{\psi}_t\}\)?
A: Yap! Huber estimator is a special case

\[
\min_{\{U, \psi_t\}} \sum_{t=1}^{T} \rho(y_t - U\psi_t)
\]

- Formally justifies the outlier-aware model and its estimator
- Ties sparse regression with robust statistics

Prior art

- Robust covariance matrix estimators [Campbell’80], [Huber’81]
  - M-type estimators in computer vision [Xu-Yuille’95], [De la Torre-Black’03]

- Rank minimization with the nuclear norm, e.g., [Recht-Fazel-Parrilo’10]
  - Matrix decomposition [Candes et al’10], [Chandrasekaran et al’11]

\[
Y = L_0 + S_0
\]

- Singing voice separation [Huang et al’12]
- Face recognition [Wright-Ma’10]

Principal Component Pursuit

\[
\min_{L, S} \|L\|_* + \lambda \|S\|_1 \\
\text{s. t. } Y = L + S.
\]
Video surveillance

- Background modeling from video feeds [De la Torre-Black ‘01]

Data: http://www.cs.cmu.edu/~ftorre/
Big Five personality factors

- Measure five broad dimensions of personality traits [Costa-McRae’ 92]

- Big Five Inventory (BFI)
  - Short-questionnaire (44 items)
  - Rate 1-5, e.g.,
    - ‘I see myself as someone who…
    - …is talkative’
    - …is full of energy’

Eugene-Springfield BFI sample (WEIRD)

\[ T = 437, D = 44, d = 5 \]

Data: courtesy of Prof. L. Goldberg, provided by Prof. N. Waller

Robust unveiling of communities

- Robust kernel PCA for identification of cohesive subgroups

- Networks: NCAA football teams (vertices), Fall '00 games (edges)

- Partitioned graph with outliers
  - Row/column-permutated adjacency matrix

- Identified exactly: Big 10, Big 12, MWC, SEC,...; Outliers: Independent teams

Data: http://www-personal.umich.edu/~mejn/netdata/
Load curve data cleansing and imputation

- **Load curve**: electric power consumption recorded periodically
  - Reliable data: key to realize smart grid vision [Hauser’09]
  - Missing data: Faulty meters, communication errors, few PMUs
  - Outliers: Unscheduled maintenance, strikes, sport events [Chen et al’10]

- **Approach**: load cleansing and imputation via distributed R-PCA

NorthWrite data

- Power consumption of schools, government building, grocery store (‘05–’10)

Cleansing

Imputation

- Outliers: “Building operational transition shoulder periods”
- Prediction error: 6% for 30% missing data (8% for 50%)

Data: courtesy of NorthWrite Energy Group.
Anomalies in social networks

- **Approach**: graph data, decompose the egonet feature matrix $\mathbf{Y}$ using PCP

- **Payoff**: unveil anomalous nodes and features

- **Outlook**: change detection, macro analysis to identify “outlying” graphs
Modeling Internet traffic anomalies

- **Anomalies**: changes in origin-destination (OD) flows [Lakhina et al’04]
  - Failures, congestions, DoS attacks, intrusions, flooding

- Graph $G(N, L)$ with $N$ nodes, $L$ links, and $F$ flows ($F >> L$); OD flow $z_{f,t}$

- Packet counts per link $l$ and time slot $t$

  \[
  y_{l,t} = \sum_{f=1}^{F} r_{l,f}(z_{f,t} + a_{f,t}) + v_{l,t}
  \]

  $\in \{0, 1\}$

- Matrix model across $T$ time slots: $Y = R(Z + A) + V$

Low-rank plus sparse matrices

- **Z (and X:=RZ)** low rank, e.g., [Zhang et al.’05]; **A** is sparse across time and flows

\[
\{\hat{X}, \hat{A}\} = \arg \min_{\{X,A\}} \frac{1}{2} \|Y - X - RA\|_F^2 + \lambda_1 \|A\|_1 + \lambda_\star \|X\|_\star
\] (P1)

Data: http://math.bu.edu/people/kolaczyk/datasets.html
Internet2 data

- Real network data, Dec. 8-28, 2003

- Improved performance by leveraging sparsity and low rank
- Succinct depiction of the network health state across flows and time

Data: http://www.cs.bu.edu/~crovella/links.html
Online estimator

- Construct an estimated map of anomalies in real time
  - Streaming data model: \( y_t = x_t + R_t a_t + v_t, \ t = 1, 2, \ldots \)

- Approach: regularized exponentially-weighted LS formulation

Roadmap

- Context and motivation
- Critical Big Data tasks

- Optimization algorithms for Big Data
  - Decentralized (in-network) operation
  - Parallel processing
  - Streaming analytics

- Randomized learning

- Scalable computing platforms for Big Data

- Conclusions and future research directions
Decentralized processing paradigms

**Goal:** Learning over networks. Why? Decentralized data, privacy

- **Limitations of FC-based architectures**
  - Lack of robustness (isolated point of failure, non-ideal links)
  - High Tx power and routing overhead (as geographical area grows)
  - Less suitable for real-time applications

- **Limitations of incremental processing**
  - Non-robust to node failures
  - (Re-) routing? Hamiltonian routes NP-hard to establish
In-network decentralized processing

- **Network anomaly detection**: spatially-distributed link count data

Centralized: \[ \text{data} = \text{local} + \text{remote} \]

Decentralized: \[ \text{data} = \sum_{i=1}^{N} \text{local}_i + \text{remote} \]

**In-network processing model:**

- **Local processing and single-hop communications**

- **Given local link counts per agent, unveil anomalies in a decentralized fashion**
  - **Challenge**: \( \| \cdot \|_* \) not separable across rows (links/agents)
Separable rank regularization

- Neat identity [Srebro’05]

\[ \|X\|_* := \min_{\{U, \Psi\}} \frac{1}{2} \left[ \|U\|_F^2 + \|\Psi\|_F^2 \right], \quad \text{s.to } X = U\Psi \]

- Nonconvex, separable formulation equivalent to (P1)

\[ \min_{\{U, \Psi, A\}} \frac{1}{2} \|Y - U\Psi - RA\|_F^2 + \lambda_1 \|A\|_1 + \frac{\lambda_*}{2} \left[ \|U\|_F^2 + \|\Psi\|_F^2 \right] \quad (P2) \]

**Proposition:** If \( \{\tilde{U}, \tilde{\Psi}, \tilde{A}\} \) stat. pt. of (P2) and \( \|Y - \tilde{U}\tilde{\Psi} - R\tilde{A}\| \leq \lambda_* \), then \( \{\hat{X} := \tilde{U}\tilde{\Psi}, \hat{A} := \tilde{A}\} \) is a **global optimum** of (P1).

- Key for parallel [Recht-Re’12], decentralized and online rank min. [Mardani et al’12]
Decentralized algorithm

- Alternating-direction method of multipliers (ADMM) solver for (P2)
  - Method [Glowinski-Marrocco’75], [Gabay-Mercier’76]
  - Learning over networks [Schizas-Ribeiro-Giannakis’07]

Consensus-based optimization

Attains centralized performance

Potential for scalable computing

Alternating direction method of multipliers

- Canonical problem

\[
\min_{\{x,y\}} f(x) + g(y), \quad \text{s.to} \quad Ax + By = c
\]

- Two sets of variables, separable cost, affine constraints

**Augmented Lagrangian**

\[
\mathcal{L}_\rho(x, y, \mu) = f(x) + g(y) + \mu^\top (Ax + By - c) + (\rho/2)\|Ax + By - c\|_2^2
\]

**ADMM**

\[
\begin{align*}
x[k+1] &= \arg \min_x \mathcal{L}_\rho(x, y[k], \mu[k]) \\
y[k+1] &= \arg \min_y \mathcal{L}_\rho(x[k+1], y, \mu[k]) \\
\mu[k+1] &= \mu[k] + \rho(Ax[k+1] + By[k+1] - c)
\end{align*}
\]

- One Gauss-Seidel pass over primal variables + dual ascent

Scaled form and variants

- Complete the squares in $\mathcal{L}_\rho$, define $m[k] := (1/\rho)\mu[k]$

ADMM (scaled)

\[
x[k + 1] = \arg\min_x f(x) + \frac{\rho}{2}\|Ax + By[k] - c + m[k]\|^2_2
\]

\[
y[k + 1] = \arg\min_y g(y) + \frac{\rho}{2}\|Ax[k + 1] + By - c + m[k]\|^2_2
\]

\[
m[k + 1] = m[k] + (Ax[k + 1] + By[k + 1] - c)
\]

- Proximal-splitting: $A = I \Rightarrow x[k + 1] = \text{prox}_{f/\rho}(-By[k] + c - m[k])$

Ex: $f(x) = \|x\|_1 \Rightarrow x[k + 1] = S(By[k] - c + m[k], 1/\rho)$

- Variants
  - More than two sets of variables
  - Multiple Gauss-Seidel passes, or, inexact primal updates
  - Strictly convex $g(y)$: $y[k + 1] = \arg\min_y \mathcal{L}(x[k + 1], y, m[k])$

Convergence

**Theorem:** If $f$ and $g$ have closed and convex epigraphs, and $\mathcal{L}$ has a saddle point, then as $k \to \infty$

- Feasibility \quad $Ax[k] + By[k] - c \to 0$
- Objective convergence \quad $f(x[k]) + g(y[k]) \to p^*$
- Dual variable convergence \quad $\mu[k] \to \mu^*$

- Under additional assumptions
  - Primal variable convergence $x[k] \to x^*$, $y[k] \to y^*$
  - Linear convergence

- No results for nonconvex objectives
  - Good empirical performance for e.g., bi-convex problems

Decentralized consensus optimization

- **Generic learning problem**
  \[
  \min_x \sum_{i=1}^{I} f_i(x)
  \]

  - Local costs \(f_i(x)\) per agent \(i\), graph \(G(\mathcal{V}, \mathcal{E})\)

- **Neat trick**: local copies of primal variables + consensus constraints

  \[
  \min_{\{x_i\}} \sum_{i=1}^{I} f_i(x_i) \quad \implies \quad \min_{\{x_i, y_{ij}\}} \sum_{i=1}^{I} f_i(x_i)
  \]

  s.t. \(x_i = x_j, \ \forall (i, j) \in \mathcal{E}\)

  \[
  \min_{\{x_i, y_{ij}\}} \sum_{i=1}^{I} f_i(x_i)
  \]

  s.t. \(x_i = y_{ij}, \ \forall (i, j) \in \mathcal{E}\)

  - Equivalent problems for connected graph
  - Amenable to decentralized implementation with ADMM

---

ADMM for in-network optimization

ADMM (in-network optimization at node $i$)

$$x_i[k+1] = \arg \min_{x_i} f_i(x_i) + \left(\frac{\rho}{2}\right) \sum_{j \in \mathcal{N}_i} \|x_i - \frac{x_i[k] + x_j[k]}{2} + m_{ij}[k]\|_2^2$$

$$m_{ij}[k+1] = m_{ij}[k] + \frac{x_i[k+1] - x_j[k+1]}{2}$$

- Auxiliary variables $y_{ij}$ eliminated!
- Communication of primary variables $x_i$ within neighborhoods $\mathcal{N}_i$ only

- Attractive features
  - Fully decentralized, devoid of coordination
  - Robust to non-ideal links; additive noise and intermittent edges [Zhu et al’10]
  - Provably convergent, attains centralized performance

Example 1: Decentralized SVM

- ADMM-based D-SVM attains centralized performance; outperforms local SVM
- Nonlinear discriminant functions effected via kernels

Example 2: RF cartography

Idea: collaborate to form a spatial map of the spectrum

Goal: find $\Phi(x, f)$ s.t. $\Phi(f) = \Phi(x_0, f)$ is the spectrum at position $x_0$

Approach: Basis expansion for $\Phi(x, f)$, decentralized, nonparametric basis pursuit

Identify idle bands across space and frequency

Parallel algorithms for BD optimization

- Computer clusters offer ample opportunities for parallel processing (PP)
- Recent software platforms promote PP

\[
\min_{x \in \mathcal{X}} \{ f(x) + g(x) \} \quad \begin{cases} 
    f : \text{Smooth loss} \\
    g : \text{Non-smooth regularizer} \\
    \mathcal{X} \subset \mathbb{R}^D, \ D \gg 0
\end{cases}
\]

- **Main idea:** Divide into $\mathcal{B}$ blocks and conquer; e.g., [Kim-GG’11]

\[
x = \begin{bmatrix}
    x_1 \\
x_2 \\
    \vdots \\
x_B
\end{bmatrix} \quad \Rightarrow \quad \text{Parallelization} \quad \Rightarrow \quad \begin{cases} 
    \min_{x_1 \in \mathcal{X}_1} \{ f(x_1 | x_{-1}) + g(x_1 | x_{-1}) \} \\
    \min_{x_2 \in \mathcal{X}_2} \{ f(x_2 | x_{-2}) + g(x_2 | x_{-2}) \} \\
    \vdots \\
    \min_{x_B \in \mathcal{X}_B} \{ f(x_B | x_{-B}) + g(x_B | x_{-B}) \}
\end{cases}
\]

\[x_{-b} : \text{All blocks other than } x_b\]

A challenging parallel optimization paradigm

\((\text{As1})\) \begin{align*}
f & : \text{Smooth + non-convex} \\
g & : \text{Non-smooth + convex + separable } \quad g(x) := \sum_{b=1}^{B} g_b(x_b)
\end{align*}

- Having available \(x^k\), find \(x_b^{k+1} \in \arg \min_{x_b \in \mathcal{X}_b} \left\{ f(x_b \mid x_{-b}^k) + g_b(x_b) \right\} \quad \text{Non-convex in general!}

- Approximate \(f(x_b \mid x_{-b}^k)\) locally by \(\tilde{f}_b(x_b; x^k)\) s.t. \((\text{As2})\) \begin{align*}
\tilde{f}_b(x_b; x^k) : \text{Convex in } x_b \\
\nabla \tilde{f}_b(x_b; x) = \nabla_{x_b} f(x)
\end{align*}

\[\text{Ex.} \quad \tilde{f}_b(x_b; x^k) = f(x^k) + \left[ \nabla_{x_b} f(x^k) \right]^\top (x_b - x_b^k) \]

- Find \(x_b^{k+1} \in \arg \min_{x_b \in \mathcal{X}_b} \left\{ \tilde{h}_b(x_b; x^k) := \tilde{f}_b(x_b; x^k) + \frac{\lambda_b}{2} \|x_b - x_b^k\|^2 + g_b(x_b) \right\} \quad \text{Quadratic proximal term renders task strongly convex!}
In addition to (As1) and (As2), if $\mu_k, \{\epsilon_b^k\}_{b=1}^B$ satisfy $\mu_k \in (0, 1]$, $\sum_k \mu_k = +\infty$, $\sum_k \mu_k^2 < +\infty$, and $\epsilon_b^k \leq \mu_k \alpha_1 \min\{\alpha_2, 1/\|\nabla_{x_b} f(x^k)\|\}$ for some $\alpha_1, \alpha_2 \geq 0$, then every cluster point of $\{x^k\}_{k=1}^\infty$ converges to a stationary solution of $\min_{x \in \mathcal{X}} \{f(x) + g(x)\}$.
Flexa on Lasso

Regression matrix of dimensions $5,000 \times 100,000$

Sparse vector with only $1\%$ non-zero entries
Streaming BD analytics

Data arrive **sequentially**, timely response needed

400M tweets/day  500TB data/day

Limits of **storage** and **computational capability**

Process one new datum at a time

Applications:

**Goal:** Develop “simple” online algorithms with performance guarantees

Roadmap of stochastic approximation

Newton-Raphson iteration
\[ F(w) = 0 \]

Law of large numbers
\[ \frac{1}{t} \sum_{\tau=1}^{t} \Phi(w; x_{\tau}) \to \mathbb{E}\{\Phi(w)\} \]

Solve
\[ F(w) := \mathbb{E}\{\Phi(w; x)\} = 0 \]

Stochastic optimization
\[ \min_{w \in \mathcal{W}} [f(w) = \mathbb{E}\{\phi(w; x)\}] \]

Det. variable
Rand. variable

Adaptive filtering
LMS and RLS as special cases

Applications
Estimation of pdfs, Gaussian mixtures
maximum likelihood estimation, etc.
Numerical analysis basics

- **Problem 1:** Find root $w_0$ of $F$; i.e., $F(w_0) = 0$
  - Newton-Raphson iteration: select $w_1$ and run
    \[ w_{t+1} = w_t - \mu_t F(w_t) \]
  - Popular choice of step-size
    \[ \mu_t = \left[ \frac{dF}{dw}(w_t) \right]^{-1} \]

- **Problem 2:** Find min or max of $f$ with gradient method ($F = \frac{df}{dw}$)
  \[ w_{t+1} = w_t - \mu_t \frac{df}{dw}(w_t) \]

- Stochastic counterparts of these deterministic iterations?
Robbins-Monro algorithm

- Robbins-Monro (R-M) iteration

\[ \hat{w}_{t+1} := \hat{w}_t - \mu_t \hat{\Phi}(\hat{w}_t; x_t) \]

- \( \mathbb{E}\{\Phi\} \) estimated on-the-fly by sample averaging

Find a root of \( F = \mathbb{E}\{\Phi\} \) using online estimates \( \hat{w}_t \) and \( \hat{\Phi}(\hat{w}_t; x_t) \)

Q: How general can the class of such problems be?

A: As general as adaptive algorithms are in:

- SP, communications, control, machine learning, pattern recognition

- Function \( F \) and data pdfs unknown!

Convergence analysis

**Theorem** If \( \mathbb{E}\{\hat{w}_1^2\} < +\infty, \{x_t\}_{t=1}^\infty : \text{i.i.d. r.vs.} \)

\[
\sum_{t=1}^\infty \mu_t = +\infty, \quad \sum_{t=1}^\infty \mu_t^2 < +\infty
\]

\((\text{As1})\) \( s_1 (w - w_0)^2 \leq F(w)(w - w_0) \leq s_2 (w - w_0)^2, \quad 0 < s_1 \leq s_2 < +\infty \)

\((\text{As2})\) \( \int [\Phi(w; x) - F(w)]^2 p(x; w) dx < \infty \)

then R-M converges in the m.s.s., i.e., \( \lim_{t \to \infty} \mathbb{E} \left\{ (\hat{w}_t - w_0)^2 \right\} = 0 \)

\((\text{As1})\) ensures unique \( w_0 \) (two lines with +tive slope cross at \( w_0 \))

\((\text{As2})\) is a finite variance requirement

- Diminishing step-size (e.g., \( \mu_t = 1/t \))
  - Limit does not oscillate around \( w_0 \) (as in LMS)
  - But convergence (\( \mu_t \to 0 \)) should not be too fast

- i.i.d. and (As1) assumptions can be relaxed!

Online learning and SA

Given basis functions $b(x) := [b_1(x), \ldots, b_M(x)]^\top$, and training data $x_t, g(x_t)$, “learn” the nonlinear function $g$ by

$$w_0 := \arg \min_w \mathbb{E} \left\{ [g(x) - w^\top b(x)]^2 \right\}$$

Solve $F(w) := \nabla_w f(w) = \mathbb{E} \left\{ b(x)b^\top(x)w - g(x)b(x) \right\} = 0$

\[=: \Phi(w; x)\]

Batch LMMSE solution: $w_0 = \left[ \mathbb{E} \left\{ b(x)b^\top(x) \right\} \right]^{-1} \mathbb{E} \left\{ g(x)b(x) \right\}

=: \mathbb{R}_{bb}^{-1} \mathbb{r}_{gb}$

R-M iteration:

$$\hat{w}_{t+1} = \hat{w}_t - \mu_t \Phi(\hat{w}_t; x_t)$$

$$= \hat{w}_t - \mu_t b(x_t) \left[ b^\top(x_t)\hat{w}_t - g(x_t) \right]$$

[least mean-squares (LMS)]
**RLS as SA**

\[
\hat{w}_{t+1} = \hat{w}_t - \mu_t b(x_t) \left[ b^\top(x_t) \hat{w}_t - g(x_t) \right]
\]

\[\downarrow\]

Scalar step size
1\textsuperscript{st} order iteration

\[
\hat{w}_{t+1} = \hat{w}_t - M_t b(x_t) \left[ b^\top(x_t) \hat{w}_t - g(x_t) \right]
\]

Matrix step size
2\textsuperscript{nd} order iteration

\[
M_t := \frac{1}{t} \hat{R}_t^{-1} = \frac{1}{t} \left[ \mathbb{E} \{ b(x)b^\top(x) \} \right]^{-1}
\]

Usually unavailable!

\[
\hat{R}_t := \frac{1}{t} \sum_{\tau=1}^{t} b(x_\tau)b^\top(x_\tau) \quad \text{: Sample correlation}
\]

\[
\hat{w}_{t+1} = \hat{w}_t - \frac{1}{t+1} \hat{R}_t^{-1} b(x_t) \left[ b^\top(x_t) \hat{w}_t - g(x_t) \right]
\]

\[
\hat{R}_t^{-1} = \frac{1}{t+1} \left[ \hat{R}_t^{-1} - \frac{\hat{R}_t^{-1} b(x_{t+1}) b^\top(x_{t+1}) \hat{R}_t^{-1}}{t + b^\top(x_{t+1}) \hat{R}_t^{-1} b(x_{t+1})} \right] \quad \text{: RLS}
\]
SA vis-a-vis stochastic optimization

\[ \text{Find } \mathbf{w}_0 = \arg \min_{\mathbf{w} \in \mathcal{W}} [f(\mathbf{w}) = \mathbb{E}\{\phi(\mathbf{w}; \mathbf{x})\}] \]

- Online convex optimization utilizes projection onto the convex \( \mathcal{W} \)

R-M iteration: \( \hat{\mathbf{w}}_{t+1} = P_{\mathcal{W}} \left( \hat{\mathbf{w}}_t - \mu_t \nabla_w \phi(\hat{\mathbf{w}}_t; \mathbf{x}_t) \right) \)

\( P_{\mathcal{W}}(\mathbf{w}) := \arg \min_{\mathbf{w}' \in \mathcal{W}} \| \mathbf{w}' - \mathbf{w} \| \)

\( \text{(As1) } f(\mathbf{w}') \geq f(\mathbf{w}) + (\mathbf{w}' - \mathbf{w})^\top \nabla_w f(\mathbf{w}) + \frac{c}{2} \| \mathbf{w} - \mathbf{w}' \|^2 : \text{ Strong convexity} \)

\( \text{(As2) } \mathbb{E} \left\{ \| \nabla_w \phi(\mathbf{w}; \mathbf{x}) \|^2 \right\} \leq \alpha^2 < \infty \)

**Theorem** Under (As1), (As2) with i.i.d. \( \{x_t\}_{t=1}^\infty \) and \( \mu_t = \mu / t \) with \( \mu > 1/(2c) \)

\[ \mathbb{E} \left\{ \| \hat{\mathbf{w}}_t - \mathbf{w}_0 \|^2 \right\} \leq \frac{1}{t} \max \left\{ \mu^2 \alpha^2 (2c\mu - 1)^{-1}, \| \hat{\mathbf{w}}_1 - \mathbf{w}_0 \|^2 \right\} =: B(\mu) \]

**Corollary** If \( \nabla_w f \) is \( L \)-Lipschitz, then \( \mathbb{E} \left\{ f(\hat{\mathbf{w}}_t) - f(\mathbf{w}_0) \right\} \leq LB(\mu)/t \)

Online convex optimization

- **Motivating example:** Website advertising company
  - Spends $\hat{w}_t \in \mathbb{R}^D$ advertising on $D$ different sites
  - Obtain clicks $-f_t(\hat{w}_t)$ at the end of week $t$

- **Online learning:** A repeated game between learner and nature (adversary)

**OCO**

For $t = 1, 2, \ldots$

- Learner updates $\hat{w}_t \in \mathcal{W} \subset \mathbb{R}^D$
- Nature provides a convex loss
  \[ f_t(w) := f_t(w; x_t) : \mathcal{W} \rightarrow \mathbb{R} \]
- Learner suffers loss $f_t(\hat{w}_t)$
Regret as performance metric

Def. 1 \( \text{Regret}_t(w) := \sum_{\tau=1}^{t} f_{\tau}(\hat{w}_{\tau}) - \sum_{\tau=1}^{t} f_{\tau}(w), \quad \forall w \in \mathcal{W} \)

Def. 2 (Worst case regret) \( \text{Regret}_t(\mathcal{W}) := \sum_{\tau=1}^{t} f_{\tau}(\hat{w}_{\tau}) - \min_{w \in \mathcal{W}} \sum_{\tau=1}^{t} f_{\tau}(w) \)

Goal: Select \( (\hat{w}_t)_{t=1}^{\infty} \) s.t.
\[
\lim_{t \to \infty} \frac{\text{Regret}_t(\mathcal{W})}{t} = 0
\]

- How to update \( \hat{w}_t \)?

Sublinear!
Online gradient descent

**OGD** Let $\hat{w}_1 = 0$ and for $t = 1, 2, \ldots$

- Choose a step-size $\mu_t$ and a (sub)gradient $z_t \in \partial f_t(\hat{w}_t)$
- Update $\hat{w}_{t+1} = \hat{w}_t - \mu_t z_t$

**Theorem** For $f_t$ convex, it holds that

$$\text{Regret}_t(w) \leq \frac{1}{2\mu_t} \|w\|^2_2 + \mu_t \sum_{\tau=1}^t \|z_t\|^2_2, \quad \forall w \in \mathcal{W}$$

If $f_t$ is $L_t$-Lipschitz cont. w.r.t. $\|\cdot\|_2$, and $\exists L$ s.t. $(1/t) \sum_{\tau=1}^t L^2_t \leq L^2$, then for $\mathcal{W} = \{w : \|w\|_2 \leq B\}$,

$$\text{Regret}_t(\mathcal{W}) \leq \frac{1}{2\mu_t} B^2 + \mu_t tL^2$$

and with $\mu_t \propto 1/\sqrt{t}$, regret is sublinear $\lim_{t \to \infty} (1/t)\text{Regret}_t(\mathcal{W}) = 0$

- Unlike SA, no stochastic assumptions; but also no primal var. convergence

Online mirror descent

Given a function \( R : \mathcal{H} \to \mathbb{R} \), \( \hat{w}_1 = 0 \), then for \( t = 1, 2, \ldots \)

- **Update via**
  \[
  \begin{align*}
    \hat{w}_t &:= \text{arg} \max_{\mathbf{w}} \langle \mathbf{w}, \theta_t \rangle - R(\mathbf{w}) \\
    \theta_{t+1} &:= \theta_t - z_t, \quad z_t \in \partial f_t(\hat{w}_t)
  \end{align*}
  \]

**Ex. 1**

\[
R(\mathbf{w}) := \frac{1}{2\mu} \| \mathbf{w} \|_2^2 \Rightarrow \begin{cases} 
  R^*(\theta) = \frac{\mu}{2} \| \theta \|_2^2 \\
  \hat{w}_t = \mu \theta_t \\
  \Rightarrow \hat{w}_{t+1} = \hat{w}_t - \mu z_t
\end{cases}
\]

**Online gradient descent!**

**Ex. 2**

\[
R(\mathbf{w}) := \begin{cases} 
  \frac{1}{\mu} \sum_i w_i \log w_i, & w_i \geq 0, \sum_i w_i = 1 \\
  +\infty, & \text{otherwise}
\end{cases}
\]

\[
\Rightarrow \begin{cases} 
  R^*(\theta) = \frac{1}{\mu} \log \sum_i e^{\mu \theta_i} \\
  [\hat{w}_t]_i = \frac{e^{\mu \theta_t}_i}{\sum_i e^{\mu \theta_t}_i} \\
  \Rightarrow [\hat{w}_{t+1}]_i = \frac{[\hat{w}_t]_i e^{-\mu [z_t]_i}}{\sum_i [\hat{w}_t]_i e^{-\mu [z_t]_i}}
\end{cases}
\]

**Exponentiated gradient!**

OMD regret

**Def.** \( R \) is called \( \nu \)-strongly convex w.r.t. \( \| \cdot \| \), if \( \exists \nu > 0 \) s.t.

\[
R(u) \geq R(v) + z^T (u - v) + \frac{\nu}{2} \| u - v \|^2, \quad \forall z \in \partial R(v), \forall (u, v)
\]

**Theorem** If \( R \) is \((1/\mu_t)\)-strongly convex w.r.t. a norm \( \| \cdot \| \), \( f_t \) is \( L_t \)-Lipschitz cont. w.r.t. the same norm, and \( \exists L \) s.t. \( (1/t) \sum_{\tau=1}^{t} L_{\tau}^2 \leq L^2 \), then

\[
\text{Regret}_t(\mathcal{W}) \leq \max_{w \in \mathcal{W}} R(w) - R(\hat{w}_1) + \frac{\mu_t}{2} t L^2
\]

If in addition \( \max_{w \in \mathcal{W}} R(w) < +\infty \) and \( \mu_t \propto 1/\sqrt{t} \), then

\[
\lim_{t \to \infty} \frac{\text{Regret}_t(\mathcal{W})}{t} = 0
\]

Gradient-free OCO

- **Setting:** Form of \( f \) is unknown; only samples can be obtained
- **Main idea:** Obtain unbiased gradient estimate on-the-fly

**Def.** Given \( \delta > 0 \), define a smoothed version of \( f \) as

\[
\hat{f}(w) := \mathbb{E}_{v \sim \mathcal{U}_{B_1}} \{ f(w + \delta v) \}
\]

\( \mathcal{U}_{B_1} \): uniform distribution over \( B_1 := \{ v : \|v\|_2 \leq 1 \} \)

**Theorem** It holds that

- If \( f \) is \( L \)-Lipschitz continuous, then \(|\hat{f}(w) - f(w)| \leq L\delta\)
- If \( \hat{f} \) is differentiable, then

\[
\nabla \hat{f}(w) = \mathbb{E}_{v \sim \mathcal{U}_{S_1}} \{ \frac{D}{\delta} f(w + \delta v)v \}
\]

When OCO meets SA

For $t = 1, 2, \ldots, \zeta_1 = 0$

S1. Find $\tilde{w}_t := \arg \min_{w \in W} \langle w, \zeta_t \rangle + \frac{1}{2\mu_t} \|w\|_2^2$

S2. Pick $v_t \sim \mathcal{U}_{S_1}$, and form $\hat{w}_t := \tilde{w}_t + \delta v_t$

S3. Let $z_t = \frac{D}{\delta} f_t(\hat{w}_t)v_t$

S4. $\zeta_{t+1} = \zeta_t - z_t$

Theorem

If $\{f_\tau\}_{\tau=1}^t$ are convex, $L$-Lipschitz continuous, and $W$ is convex, with $B = \max_{w \in W} \|w\|_2 < \infty$ and $F = \max_{w \in W, \tau \in \{1, \ldots, t\}} f_\tau(w) < \infty$, then

$$\mathbb{E} \left\{ \sum_{\tau=1}^t [f_\tau(\hat{w}_\tau + \delta v_\tau) - f_\tau(u)] \right\} \leq 3L\delta t + \frac{B^2}{2\mu_t} + \mu_t t D^2 \left( \frac{F}{\delta} + L \right)^2$$

In particular, if $\mu_t \propto 1/\sqrt{t}$ and $\delta_t \propto 1/\sqrt[4]{t}$, then regret is bounded by $O(t^{3/4})$

➢ Compare $O(t^{3/4})$ in the agnostic vs $O(t^{1/2})$ in the full information case

Roadmap

- Context and motivation
- Critical Big Data tasks
- Optimization algorithms for Big Data
- Randomized learning
  - Randomized linear regression, leverage scores
  - Johnson-Lindenstrauss lemma
  - Randomized classification and clustering
- Scalable computing platforms for Big Data
- Conclusions and future research directions
Randomized linear algebra

- **Basic tools**: Random sampling and random projections

- **Attractive features**
  - Reduced dimensionality to lower complexity with Big Data
  - Rigorous error analysis at reduced dimension

**Ordinary least-squares (LS)**

Given \( \mathbf{y} \in \mathbb{R}^D, \mathbf{X} \in \mathbb{R}^{D \times p} \)

\[
\mathbf{\theta}_{\text{LS}} := \arg \min_{\mathbf{\theta} \in \mathbb{R}^p} \| \mathbf{y} - \mathbf{X} \mathbf{\theta} \|_2^2
\]

If \( \text{rank}(\mathbf{X}) = p \) \( \Rightarrow \) \( \mathbf{\theta}_{\text{LS}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y} \)

**SVD** \( \mathbf{X} = \mathbf{U} \Sigma \mathbf{V}^\top \) \( \Rightarrow \) \( \mathbf{P}_R(\mathbf{X}) = \mathbf{U} \mathbf{U}^\top = \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \)

**LS-optimal prediction** \( \Rightarrow \) \( \hat{\mathbf{y}} := \mathbf{P}_R(\mathbf{X})(\mathbf{y}) = \mathbf{U} \mathbf{U}^\top \mathbf{y} \)

- **SVD incurs complexity** \( \mathcal{O}(Dp^2) \). **Q**: What if \( D \gg p \)?
Randomized linear regression

**Key idea:** Sample and rescale (down to $d$) “important” observations

$$p \leq d \ll D$$

$$\Gamma_d \in \mathbb{R}^{d \times d} : \quad \text{Diagonal rescaling matrix}$$

$$S_d \in \{0, 1\}^{d \times D} : \quad \text{Random sampling matrix}$$

$$\hat{\theta}_{LS} := \arg \min_{\theta \in \mathbb{R}^p} \| \Gamma_d S_d (y - X\theta) \|_2^2$$

**Random sampling:** For $i = 1, \ldots, d$

- Pick $j_i$th row (entry) of $X$ ($y$) w.p. $\pi_{j_i}$

$$[S_d]_{ij_i} := 1, \quad [\Gamma_d]_{ii} = 1/\sqrt{d\pi_{j_i}}$$

**Q:** How to determine $\{\pi_{j_i}\}_{j_i=1}^D$?
Statistical leverage scores

- Prediction error \( \tilde{y} := y - \hat{y} \), where \( y = X\theta + v \) and \( \hat{y} := P_R(x)(y) = UU^T y \)

\[
\text{Var}(\tilde{y}_i) = \sigma^2 (1 - [P_R(x)]_{ii}), \quad |\text{Cov}(\tilde{y}_i, \tilde{y}_j)| \leq \sigma^2 \sqrt{[D - \text{rank}(X)](1 - [P_R(x)]_{ii})}
\]

Large \([P_R(x)]_{ii}\) \(\Rightarrow\) Small contribution of \(\tilde{y}_i\) to prediction error!

**Definition:** Statistical leverage scores (SLS)

\[
\pi_i := \frac{[P_R(x)]_{ii}}{\text{trace}(P_R(x))}
\]

- Sample rows (entries) of \(X (y)\) according to \(\{\pi_i\}_{i=1}^D\)

**Theorem**

For any \(\epsilon > 0\), if \(d = \mathcal{O}(p \log p/\epsilon^2)\), then w.h.p.

\[
\|y - X\tilde{\theta}_{LS}\|_2 \leq (1 + \epsilon)\|y - X\theta_{LS}\|_2 \\
\|\theta_{LS} - \tilde{\theta}_{LS}\|_2 \leq \sqrt{\epsilon} \kappa(X) \sqrt{\gamma^{-2} - 1} \|\theta_{LS}\|_2 \\
\kappa(X) \text{ condition number of } X; \text{ and } \gamma = \|\hat{y}\|_2/\|y\|_2
\]

- SLS complexity \(\mathcal{O}(Dp^2)\)

Johnson-Lindenstrauss lemma

- The “workhorse” for proofs involving random projections

**JL lemma:** If $0 < \epsilon < 1$, integer $T$, and reduced dimension satisfies

$$d \geq 4\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} \ln T$$

then for any $Y = [y_1, \ldots, y_T] \in \mathbb{R}^{D \times T}$ there exists a mapping $f : \mathbb{R}^D \rightarrow \mathbb{R}^d$ s.t.

$$ (1 - \epsilon)\|y_{t_1} - y_{t_2}\|^2 \leq \|f(y_{t_1}) - f(y_{t_2})\|^2 \leq (1 + \epsilon)\|y_{t_1} - y_{t_2}\|^2 \quad (\star) $$

Almost preserves pairwise distances!

- If $f(y) := d^{-1/2}Ry$ with i.i.d. $\mathcal{N}(0, 1)$ entries of $R$ and reduced dimension

$$d \geq 4\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} \ln T + O(\log \log T),$$

then $(\star)$ holds w.h.p. [Indyk-Motwani'98]

- If $f(y) := d^{-1/2}Ry$ with i.i.d. uniform over $\{+1,-1\}$ entries of $R$ and reduced dimension as in JL lemma, then $(\star)$ holds w.h.p. [Achlioptas’01]

---

Bypassing cumbersome SLS

- Pre-conditioned LS estimate
  \[ \hat{\theta}_{LS} = \arg \min_{\theta \in \mathbb{R}^p} \| \Gamma_d S_d H_D \Delta_D (y - X\theta) \|_2^2 \]

- Hadamard matrix
  \[ H_D = \frac{1}{\sqrt{D}} \begin{bmatrix} H_{D/2} & H_{D/2} \\ H_{D/2} & -H_{D/2} \end{bmatrix}, \quad H_2 := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \]

- Random diagonal matrix \( \Delta_D \) with \( [\Delta_D]_{ii} \in \{1, -1\} \sim \text{Ber}(1/2) \)

- Select reduced dimension \( d = \mathcal{O}(p \log p \cdot \log D + \epsilon^{-1} D \log p) \)

- Sample/rescale via \( S_d, \Gamma_d \) with uniform pdf \( \pi_i := 1/D, \forall i \)

- Complexity \( o(Dp^2) \) lower than \( \mathcal{O}(Dp^2) \) incurred by SLS computation

Testing randomized linear regression

- Uniform sampling vs Hadamard pre-conditioning
  - $D = 1,024$, $p = 10$, $C_v = 9I_p$
  - Performance depends on $X$

Big data classification

- Support vector machines (SVM): The workhorse for linear discriminant analysis
  - **Data:** \( \{x_t\}_{t=1}^T \subset \mathbb{R}^{D \times T} \) with labels \( \{y_t\}_{t=1}^T \subset \{\pm 1\} \)
  - **Goal:** Separating hyperplane \((w, b)\) to max “between-class” margin

- Binary classification
  - **Primal problem**
    \[
    \min_{(w,b)\in \mathbb{R}^{D} \times \mathbb{R}} \quad \frac{1}{2}\|w\|_2^2 + C \sum_{t=1}^{T} \xi_t \\
    \text{s.t.} \quad \begin{cases}
    \forall t \in \{1, \ldots, T\} \\
    y_t \left[ w^T x_t + b \right] \geq 1 - \xi_t \\
    \xi_t \geq 0
    \end{cases}
    \]
  - Margin: \( \gamma := \frac{1}{\|w\|_2} \)
Randomized SVM classifier

- Dual problem

\[
\min_{\alpha \in \mathbb{R}^T} \frac{1}{2} \alpha^\top Y^\top X^\top X Y \alpha - 1_T^\top \alpha
\]

s.t.
\[
\alpha_t \in [0, C], \quad \forall t \in \{1, \ldots, T\}
\]
\[
1_T^\top Y \alpha = 0
\]

- Random projections approach

- Given \( R \in \mathbb{R}^{d \times D} (d \ll D) \), let \( \tilde{X} = RX \in \mathbb{R}^{d \times T} \)

- Solve the transformed dual problem

\[
\min_{\tilde{\alpha} \in \mathbb{R}^T} \frac{1}{2} \tilde{\alpha}^\top \tilde{Y}^\top \tilde{X}^\top \tilde{X} Y \tilde{\alpha} - 1_T^\top \tilde{\alpha}
\]

s.t.
\[
\tilde{\alpha}_t \in [0, C], \quad \forall t \in \{1, \ldots, T\}
\]
\[
1_T^\top \tilde{Y} \tilde{\alpha} = 0
\]

Performance analysis

**Theorem** Given $\rho := \text{rank}(X)$, accuracy $\epsilon \in (0, 1/2]$, $\zeta \in (0, 1]$, and

$$d = \mathcal{O} \left[ \rho \epsilon^{-2} \log(\rho D \zeta^{-1}) \log(\rho \epsilon^{-2} \zeta^{-1} \log(\rho D \zeta^{-1})) \right]$$

there exists $R \in \mathbb{R}^{d \times D}$ such that if $\gamma_*$ and $\tilde{\gamma}_*$ are SVM-optimal margins corresponding to $X$ and $\tilde{X} := RX$, respectively, then w.p. $\geq 1 - \zeta$

$$\tilde{\gamma}_*^2 \geq (1 - \epsilon) \gamma_*^2$$

**Lemma** Vapnik-Chernovenkis dimension almost unchanged!

Out-of-sample error kept almost unchanged in the reduced-dimension solution

---

Testing randomized SVM

- 295 datasets from Open Directory Project (ODP)
- Each dataset includes
  - $T = 150$ to 280 documents
  - $D = 10,000$ to 40,000 words (features)
  - Binary classification
- Performance in terms of
  - Out-of-sample error ($\epsilon_{\text{out}}$)
  - Margin ($\gamma$)
  - Projection and run time ($t_{\text{rp}}$ and $t_{\text{run}}$)
- In-sample error matches for full/reduced dimensions
- $\epsilon_{\text{out}}$ and $\gamma$ improves as $d$ increases

<table>
<thead>
<tr>
<th>$\epsilon_{\text{out}}$</th>
<th>Projected Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>full</td>
</tr>
<tr>
<td>CW $(\mu)$</td>
<td>24.63</td>
</tr>
<tr>
<td></td>
<td>10.57</td>
</tr>
<tr>
<td>RS $(\mu)$</td>
<td>24.58</td>
</tr>
<tr>
<td></td>
<td>10.57</td>
</tr>
<tr>
<td>FHT $(\mu)$</td>
<td>24.63</td>
</tr>
<tr>
<td></td>
<td>10.66</td>
</tr>
<tr>
<td>RG $(\mu)$</td>
<td>24.59</td>
</tr>
<tr>
<td></td>
<td>10.54</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>Projected Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>full</td>
</tr>
<tr>
<td>CW $(\mu)$</td>
<td>1.66</td>
</tr>
<tr>
<td></td>
<td>3.68</td>
</tr>
<tr>
<td>RS $(\mu)$</td>
<td>1.66</td>
</tr>
<tr>
<td></td>
<td>3.65</td>
</tr>
<tr>
<td>FHT $(\mu)$</td>
<td>1.66</td>
</tr>
<tr>
<td></td>
<td>3.65</td>
</tr>
<tr>
<td>RG $(\mu)$</td>
<td>1.66</td>
</tr>
<tr>
<td></td>
<td>3.70</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$t_{\text{rp}}$</th>
<th>Projected Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>full</td>
</tr>
<tr>
<td>CW $(\mu)$</td>
<td>0.0046</td>
</tr>
<tr>
<td></td>
<td>0.0019</td>
</tr>
<tr>
<td>RS $(\mu)$</td>
<td>0.0429</td>
</tr>
<tr>
<td></td>
<td>0.0178</td>
</tr>
<tr>
<td>FHT $(\mu)$</td>
<td>0.0443</td>
</tr>
<tr>
<td></td>
<td>0.0206</td>
</tr>
<tr>
<td>RG $(\mu)$</td>
<td>0.039</td>
</tr>
<tr>
<td></td>
<td>0.0159</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$t_{\text{run}}$</th>
<th>Projected Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>full</td>
</tr>
<tr>
<td>CW $(\mu)$</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td>0.87</td>
</tr>
<tr>
<td>RS $(\mu)$</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>0.97</td>
</tr>
<tr>
<td>FHT $(\mu)$</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>0.96</td>
</tr>
<tr>
<td>RG $(\mu)$</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>0.83</td>
</tr>
</tbody>
</table>

Big data clustering

- Given \( \{y_t\}_{t=1}^{T} \) with \( \text{dim}(y_t) = D \) assign them to clusters
- **Key idea:** Reduce dimensionality via random projections
- **Desiderata:** Preserve the pairwise data distances in lower dimensions

\[
Y := [y_1, \ldots, y_T]
\]

**Feature extraction**

- Construct \( d \ll D \) “combined” features (e.g., via \( RY \))
- Apply K-means to \( d \)-space

**Feature selection**

- Select \( d \ll D \) of input features (rows of \( Y \))
- Apply K-means to \( d \)-space
Randomized $K$-means

- Cluster $C_k$ with centroid $c_k = \frac{1}{|C_k|} \sum_{y_t \in C_k} y_t = Y [0, 1, \ldots, 1, 0]^\top \frac{1}{|C_k|}$

- Data-cluster association matrix $A \in \mathbb{R}^{K \times T}$

$$[A]_{kt} := \begin{cases} \frac{1}{\sqrt{|C_k|}}, & \text{if } y_t \in C_k \\ 0, & \text{o.w.} \end{cases}$$

- **$K$-means criterion:** $F(Y, A) := \|Y - YA^\top A\|_F^2$

$$A_K := \{A \in \mathbb{R}^{K \times T} : A \text{ is data-cluster assoc. matrix} \}$$

Def. $(1 + \delta)$-approximation algorithm: Any algorithm yielding $A_{(1+\delta)}$ s.t. w.h.p.

$$F(Y, A_{(1+\delta)}) \leq (1 + \delta)F(Y, A_{\text{opt}}), \quad \delta \in (0, 1)$$

- Given a “projection” matrix $R \in \mathbb{R}^{d \times D} (d \ll D)$ and $\tilde{Y} := RY \in \mathbb{R}^{d \times T}$

$$\tilde{A}_{\text{opt}} := \arg \min_{A \in A_K} F(\tilde{Y}, A)$$

Feature extraction via random projections

**Theorem** Given number of clusters $K$, accuracy $\epsilon \in (0, 1/3)$, and $d := \mathcal{O}(\frac{K}{\epsilon^2})$, construct i.i.d. Bernoulli $\mathbf{R} \in \mathbb{R}^{d \times D}$ with $[\mathbf{R}]_{i,j} = \pm 1/\sqrt{d}$. Then, any $(1 + \delta)$-approximation $K$-means on $\tilde{\mathbf{Y}} = \mathbf{R}\mathbf{Y}$, yielding $\tilde{\mathbf{A}}_{(1+\delta)}$, satisfies w.h.p.

$$F(\mathbf{Y}, \tilde{\mathbf{A}}_{(1+\delta)}) \leq [1 + (1 + \epsilon)(1 + \delta)]F(\mathbf{Y}, \mathbf{A}_{opt})$$

- **Complexity:** $\mathcal{O}\left( TD \left\lfloor \frac{K}{\epsilon^2 \log(T)} \right\rfloor \right)$

- **Related results** [Boutsidis et al’14]
  - For $d = K$, $\mathbf{R} := \mathbf{U}_K^\top$ ($\mathbf{U}_K$: $K$-principal comp. of $\mathbf{Y}\mathbf{Y}^\top$); compl. $\mathcal{O}(TD \min\{T, D\})$

  - Approx. $\mathbf{U}_K$ by rand. alg. $\mathbf{Z}_K := \text{FastSVD}(\mathbf{Y}, K)$; complexity $\mathcal{O}(T DK/\epsilon)$

Feature selection via random projections

Given $\mathbf{Y} \in \mathbb{R}^{D \times T}$ select features by constructing $\tilde{\mathbf{Y}} \in \mathbb{R}^{d \times T}$

**S1.** $\mathbf{Z}_K := \text{FastSVD}(\mathbf{Y}, K) \in \mathbb{R}^{D \times K}$

**S2.** $d := \mathcal{O}\left(4K \ln(200K)/\epsilon^2\right)$

**S3.** $\Psi := \text{RandomSampling}(\mathbf{Z}_K, d), \quad \Psi \in \mathbb{R}^{d \times D}$

**S3a.** $\pi_j := \frac{[\mathbf{Z}_K \mathbf{Z}_K^T]_{jj}}{\text{trace}(\mathbf{Z}_K \mathbf{Z}_K^T)}, \quad \forall j \in \{1, \ldots, D\}$ Leverage scores!

**S3b.** $\Psi = 0$ and for $i = 1, \ldots, d$

- Pick $j_i \in \{1, \ldots, D\}$ w.p. $\pi_{j_i}$
- $[\Psi]_{ij} := \frac{1}{\sqrt{d \pi_{j_i}}}$ Complexity: $\mathcal{O}(DK + d \log d)$

**S4.** $\tilde{\mathbf{Y}} := \Psi \mathbf{Y}$

**Theorem** Any $(1 + \delta)$-approximation $K$-means on $\tilde{\mathbf{Y}}$, yielding $\tilde{\mathbf{A}}_{(1+\delta)}$, satisfies w.p. $\geq 0.2$

$$F(\mathbf{Y}, \tilde{\mathbf{A}}_{(1+\delta)}) \leq \left[1 + (2 + \epsilon)(1 + \delta)\right]F(\mathbf{Y}, \mathbf{A}_{\text{opt}})$$

## Random projection matrices

<table>
<thead>
<tr>
<th>Random projection matrix</th>
<th>Reduced dimension ( (d) )</th>
<th>Projection cost ( (t_{rp}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.i.d. random Gaussian (RG)</td>
<td>( \mathcal{O}(\rho \epsilon^{-2} \log(\frac{\rho}{\zeta})) )</td>
<td>( \mathcal{O}(T D d) )</td>
</tr>
<tr>
<td>Random sign matrix (RS)</td>
<td>( \mathcal{O}(\rho \epsilon^{-2} \log(\rho) \log(D)) )</td>
<td>( \mathcal{O}(T D d) )</td>
</tr>
<tr>
<td>Random Hadamard transform (FHT)</td>
<td>( \mathcal{O}(\rho \epsilon^{-2} \log(\frac{\rho D}{\zeta}) \log(\rho \epsilon^{-2} \zeta^{-1} \log(\frac{\rho D}{\zeta}))) )</td>
<td>( \mathcal{O}(T D \log(d)) )</td>
</tr>
<tr>
<td>Sparse embedding matrix* [Clarkson, Woodruff ’13]</td>
<td>( \mathcal{O}(\rho \epsilon^{-4} \log(\frac{\rho}{\zeta \epsilon})(\rho + \log(\frac{1}{\zeta \epsilon}))) )</td>
<td>( \mathcal{O}(\text{nnz}(Y)) + \text{poly}(d/\epsilon)** )</td>
</tr>
</tbody>
</table>

* Exploits sparsity in data matrix

** \( \text{nnz}(Y) \) := Number of nonzero entries in \( Y \)

\( \text{poly}(T/\epsilon) \) := Polynomial complexity w.r.t. \( T/\epsilon \)

---

Testing randomized clustering

- $T = 1,000$, $D = 2,000$, and $K = 5$ well separated clusters

Roadmap

- Context and motivation
- Critical Big Data tasks
- Optimization algorithms for Big Data
- Randomized learning

- Scalable computing platforms for Big Data
  - MapReduce platform
  - Least-squares and k-means in MapReduce
  - Graph algorithms

- Conclusions and future research directions
Parallel programming issues

- Challenges
  - Load balancing; communications
  - Scattered data; synchronization (deadlocks, read-write racing)

- General vs application-specific implementations
  - Efficient vs development overhead tradeoffs

Choosing the right tool is important!
MapReduce

- General parallel programming (PP) paradigm for large-scale problems
- Easy parallelization at a high level
  - Synchronization, communication challenges are taken care of
- Initially developed by Google

Processing divided in two stages (S1 and S2)

**S1 (Map):** PP for portions of the input data

**S2 (Reduce):** Aggregation of S1 results (possibly in parallel)

Example

\[ \sum_{i} f(w_i) \]

Reduce \[ \sum_{i} \rightarrow f(w_i) \rightarrow \text{Map per processor } i \]
Basic ingredients of MapReduce

- All data are in the form <key, value> e.g., key = “machine,” value = 5

- Map-Reduce model consists of [Lin-Dryer ‘10]

  - Input key-value pairs
  - Mappers
    - Programs applying a Map operation to a subset of input data
    - Run same operation in parallel
    - Produce intermediate key-value pairs
  - Reducers
    - Programs applying a Reduce operation (e.g., aggregation) to a subset of intermediate key-value pairs with the same key
    - Reducers execute same operation in parallel to generate output key-value pairs
  - Execution framework
    - Transparent to the programmer; performs tasks, e.g., sync and com

Running MapReduce

- Each job has three phases

  - **Phase 1: Map**
    - Receive input key-value pairs to produce intermediate key-value pairs

  - **Phase 2: Shuffle and sort**
    - Intermediate key-value pairs sorted; sent to proper reducers
    - Handled by the execution framework
    - Single point of communication
    - “Slowest” phase of MapReduce

  - **Phase 3: Reduce**
    - Receive intermediate key-value pairs the final ones
    - All mappers must have finished for reducers to start!
The MapReduce model

Map phase

Reduce phase

Aggregates intermediate key-value pairs of each mapper

Assign intermediate key-value pairs to reducers e.g. key mod #reducers
Least-squares in MapReduce

- Given $\mathbf{y} \in \mathbb{R}^D$, $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_D]^\top \in \mathbb{R}^{D \times p}$ [$p \ll D$, rank($\mathbf{X}$) = $p$]
  find
  $$\theta_{LS} := \arg \min_{\theta \in \mathbb{R}^p} \|\mathbf{y} - \mathbf{X}\theta\|^2_2 = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

- Map Phase
  - Mappers (here 3) calculate partial sums
    $$\bigcup M_1 \cup M_2 \cup M_3 = \{1, \ldots, D\}$$
    $$M_i \cap M_j = \emptyset$$
    $$\sum_{i \in M_1} \mathbf{x}_i \mathbf{x}_i^\top$$
    $$\sum_{i \in M_2} \mathbf{x}_i \mathbf{x}_i^\top$$
    $$\sum_{i \in M_3} \mathbf{x}_i \mathbf{x}_i^\top$$
    $$\sum_{i \in M_1} y_i \mathbf{x}_i$$
    $$\sum_{i \in M_2} y_i \mathbf{x}_i$$
    $$\sum_{i \in M_3} y_i \mathbf{x}_i$$

- Reduce Phase
  - Reducer aggregates partial sums
  - Computes $(\mathbf{X}^\top \mathbf{X})^{-1}$
  - Finds $\theta_{LS}$

K-means in MapReduce

Serial k-means
- Pick K random initial centroids
- Iterate:
  - Assigning points to cluster of closest centroid
  - Updating centroids by averaging data per cluster

MapReduce k-means
- A central unit picks K random initial centroids
- Each iteration is a MapReduce job (multiple required!)

Mappers
- Each assigned a subset of data
- Find closest centroid, partial sum of data, and #data per cluster
- Generate key-value pairs; key = cluster, value = partial sum, #points

Reducers
- Each assigned a subset of clusters
- Aggregate partial sums for corresponding clusters
- Update centroid location

- : centroids
- : Mapper 1
- : Mapper 2
- : Reducer 1
- : Reducer 2
Graph algorithms

- Require communication between processing nodes across multiple iterations; **MapReduce not ideal!**

**Pregel**
- Based on Bulk Synchronous Parallel (BSP) model
- Developed by Google for use on large graph problems
- Each vertex is characterized by a user-defined value
- Each edge is characterized by source vertex, destination vertex and a user-defined value

**Computation is performed in “Supersteps”**
- A user-defined function is executed on each vertex (parallel)
- Messages from previous superstep are received
- Messages for the next superstep are sent
- Supersteps are separated by global synchronization points

More efficient than MapReduce!

Wrap-up for scalable computing

- MapReduce
  - Simple parallel processing
  - Code easy to understand
  - Great for batch problems
  - Applicable to many problems

- However...
  - Framework can be restrictive
  - Not designed for online algorithms
  - Inefficient for graph processing

- Pregel
  - Tailored for graph algorithms
  - More efficient relative to MapReduce

- Available open source implementations
  - Hadoop (MapReduce)
  - Hama (Pregel)
  - Giraph (Pregel)

- Machine learning algorithms can be found at Mahout!
Roadmap

- Context and motivation
- Critical Big Data tasks
- Optimization algorithms for Big Data
- Randomized learning
- Scalable computing platforms for Big Data
- Conclusions and future research directions
Tutorial summary

- **Big Data modeling and tasks**
  - Dimensionality reduction
  - Succinct representations
  - Vectors, matrices, and tensors

- **Optimization algorithms**
  - Decentralized, parallel, streaming
  - Data sketching

- **Implementation platforms**
  - Scalable computing platforms
  - Analytics in the cloud
Timeliness

- Special sessions in recent IEEE SP Society / EURASIP meetings

ICASSP 2012
“Signal and Information Processing for Big Data”

SSP’12 @ MICHIGAN
“Challenges in High-Dimensional Learning”

GlobalSIP 2013
“Information Processing over Networks”

EUSIPCO 2013
“Trends in Sparse Signal Processing”

ICASSP 2013
“Sparse Signal Techniques for Web Information Processing”

ICASSP 2014
“SP for Big Data”
“Optimization Algorithms for High-Dimensional SP”

SAM 2014
“Tensor-based SP”

GlobalSIP
“Information Processing for Big Data”
Importance to funding agencies

- NSF, NIH, DARPA, DoD, DoE, and US Geological Survey

Sample programs:
- NSF 14-543 (BIGDATA)
- DARPA – ADAMS (Anomaly Detection on Multiple Scales)

DoD: 20 + solicitations
- Data to decisions
- Autonomy
- Human-machine systems

Source: www.whitehouse.gov/sites/default/files/microsites/ostp/big_data_press_release_final_2.pdf
NSF-ECCS sponsored workshop

NSF Workshop on

**Big Data**

*From Signal Processing to Systems Engineering*

_March 21–22, 2013 — Arlington, Virginia, USA_

- Workshop program and slides
  

- Workshop final report
  

Sponsored by

Electrical, Communications and Cyber Systems (ECCS)
Open JSTSP special issue

- Special issue on **Signal Processing for Big Data**
  *IEEE Journal of Selected Topics in Signal Processing*

- **Guest editors**
  - Georgios B. Giannakis – University of Minnesota, USA
  - Raphael Cendrillon – Google, USA
  - Volkan Cevher – EPFL, Switzerland
  - Ananthram Swami – Army Research Laboratory, USA
  - Zhi Tian – Michigan Tech Univ. and NSF, USA

  - Full manuscripts due by July 1, 2014

- Sept.’14 issue of the *IEEE Signal Processing Magazine* (also on SP for Big Data)
Questions?

http://spincom.umn.edu