Big Data Signal Processing for Communication Networks

Zijie Zheng, Peking University

Zhu Han, University of Houston

Geoffrey Ye Li, Georgia Institute of Technology

IEEE GLOBECOM 2017

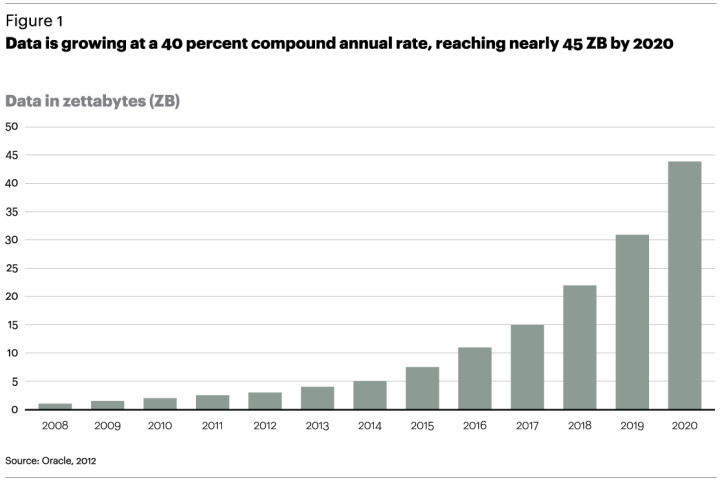
Content

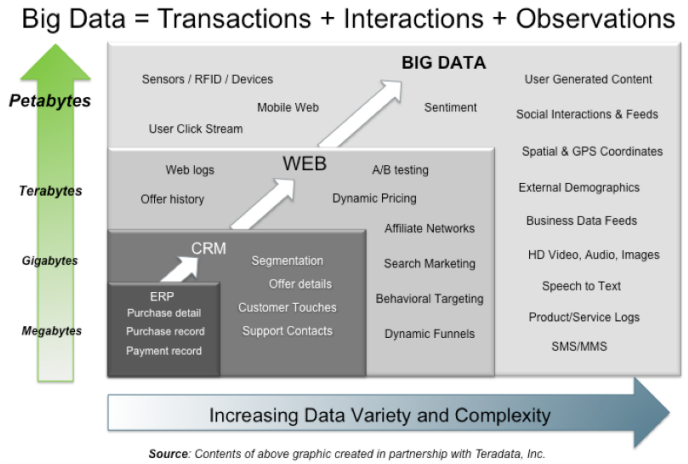
- Overview of Big Data
- Learning Methods
- Commercial Systems
- Large Scale Optimization
- Game Theory based Approaches

Content

- Overview of Big Data
- Learning Methods
- Commercial Systems
- Large Scale Optimization
- Game Theory based Approaches

Big Data 3V Characteristics





- Exponentially increasing data size
- Heterogeneous types of data
  - Possible to analyze ALL available data
- Need to calculate fast
- Key dimensions
  - Volume, Velocity and Variety
  - Variability and Veracity

Cisco White Paper,"Visual networking index: global mobile data traffic forecast update, 2016-2021," Mar. 2017.

Big Data Business

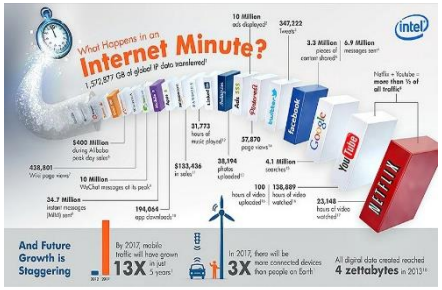
However

- Nobody knows exactly how to handle big data

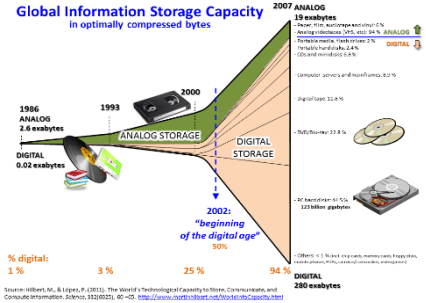
- We zoom to
  - Signal processing techniques
  - Networking applications

Big Data Techniques

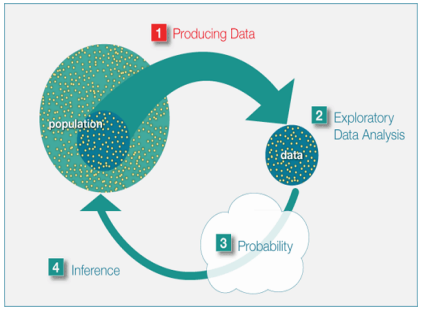
Data Mining



Big Data Storage

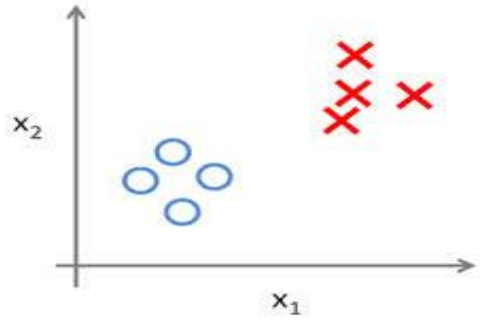


Big Data Sampling



Real Time Processing

Machine Learning Methods

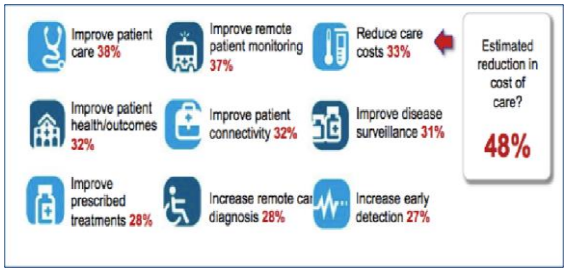


...

Big Data Applications

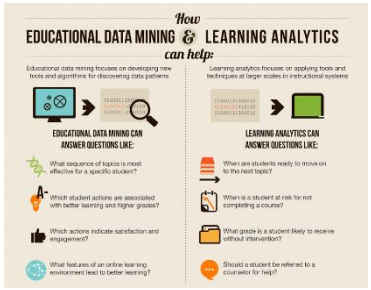
Manufacturing

Healthcare



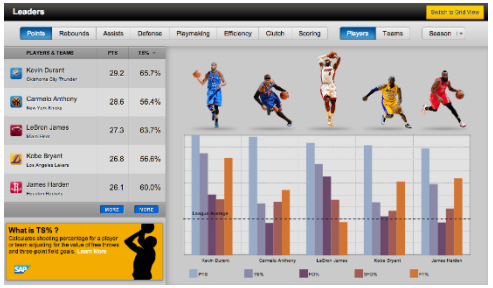
Media

Education



Financial Services

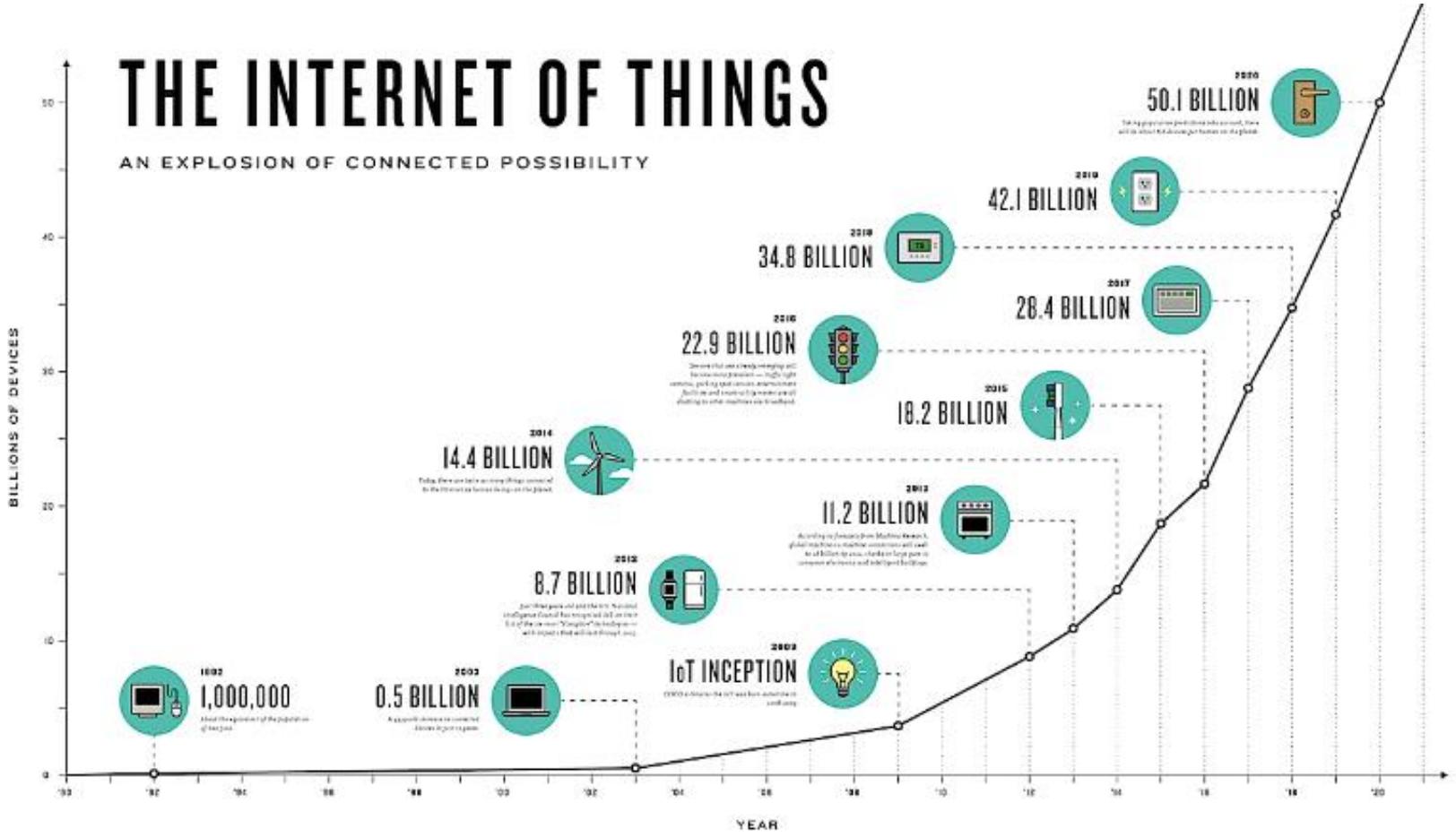
Sports



Communication and Networking

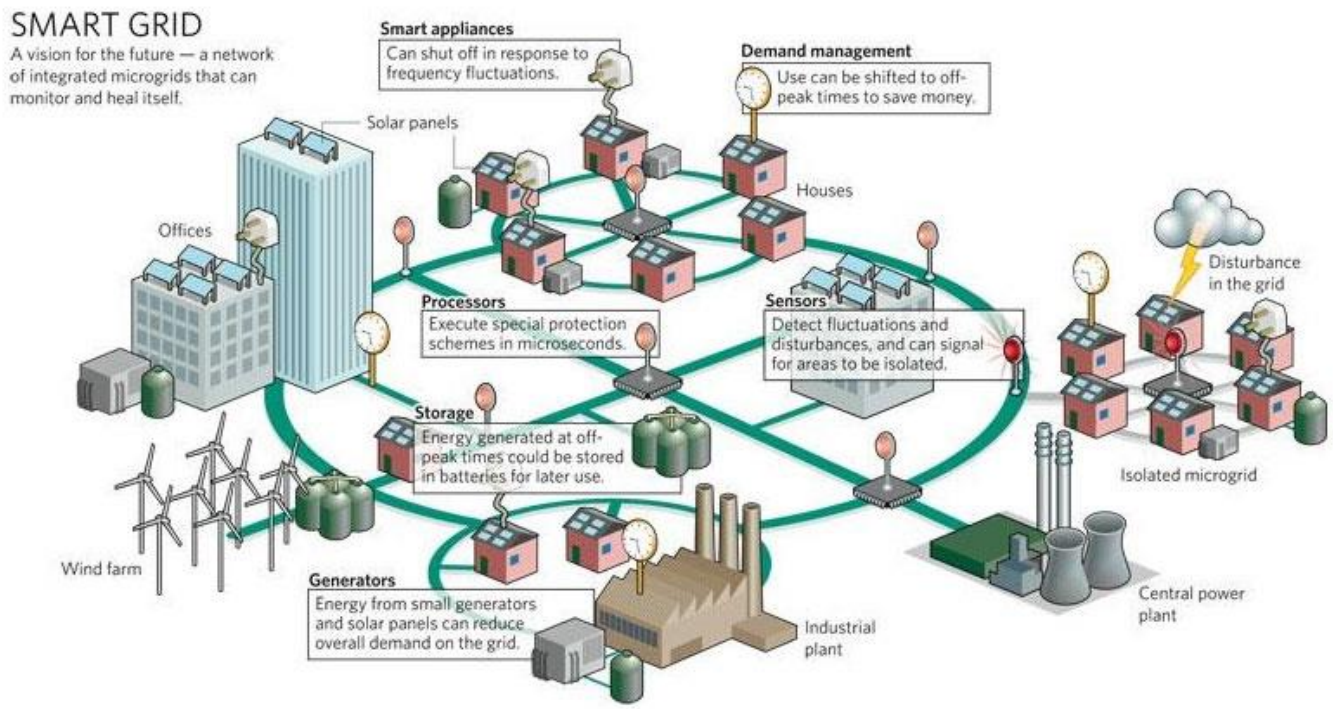
- **Fast cloud computing vs. slow communication**
- **Local, fog vs. cloud**

Internet of Things

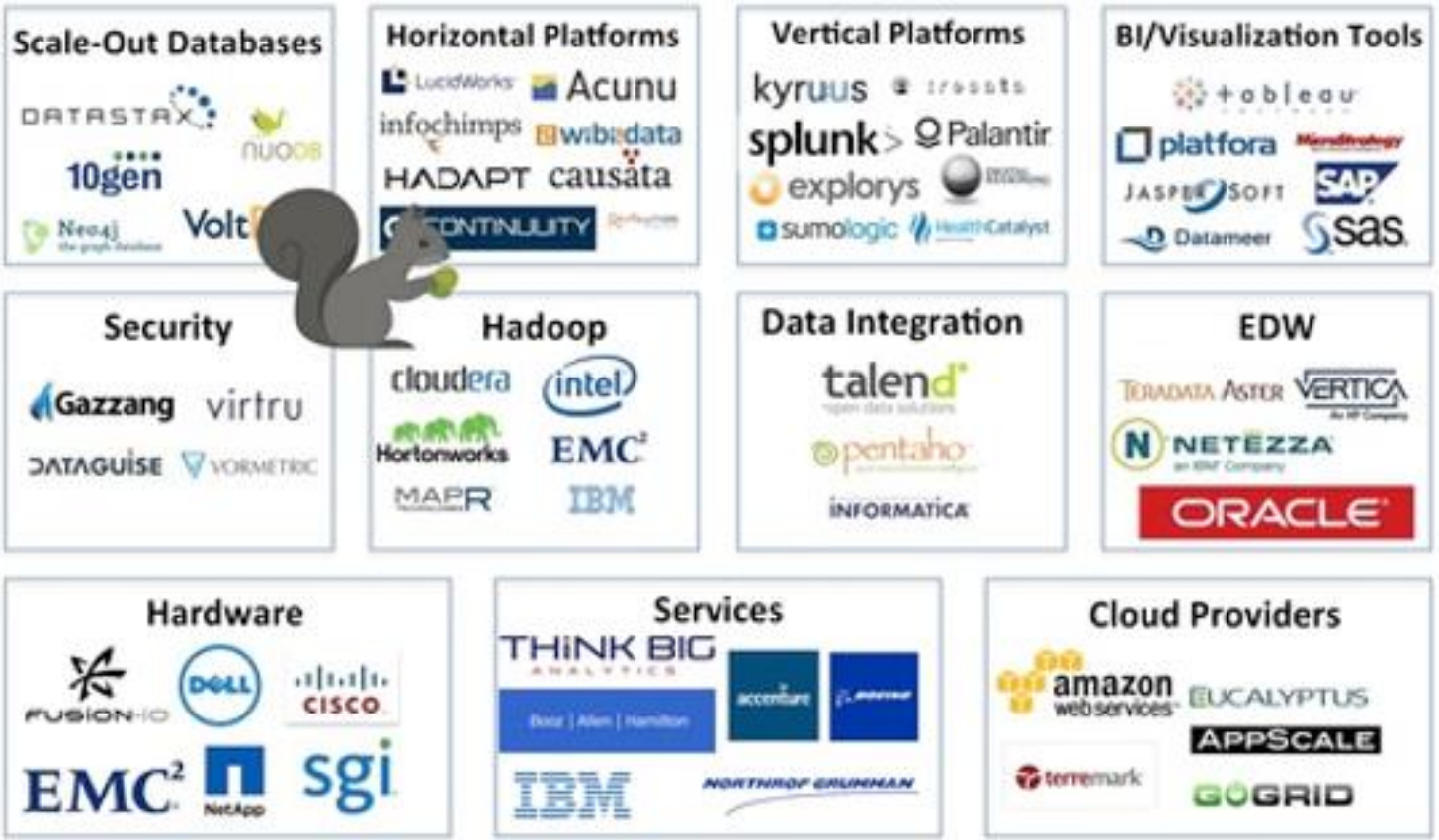


Smart Grid

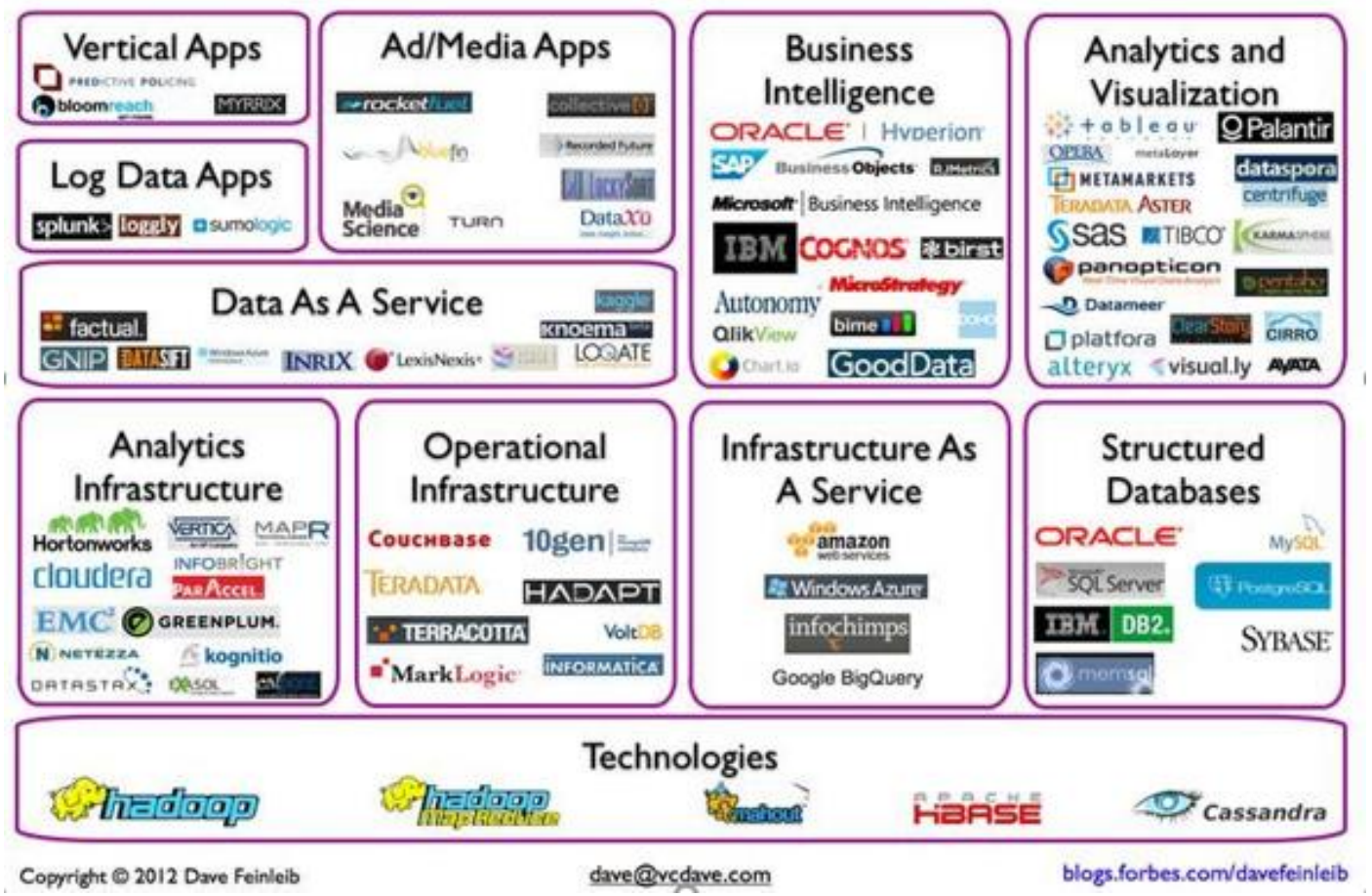
- **In 2009, "American Recovery and Reinvestment Act"**
  - $3.4 billion for SG investment grant program
  - $615 million for SG demonstration program
  - A combined investment of $8 billion in SG capabilities



Big Data Ecosystem



Big Data Landscape



Content

- Overview of Big Data
- Learning Methods
- Large Scale Optimization
- Game Theory based Approaches
- Commercial Systems

Learning Methods

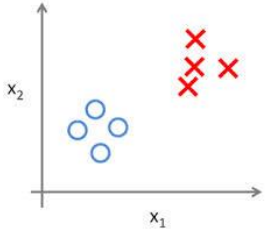
- Classical Machine Learning
- Bayesian Nonparametric learning
- Deep Learning

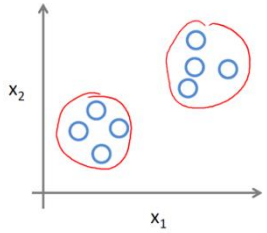
Classical Machine Learning

"Computers: learn without being explicitly programmed"

Types:

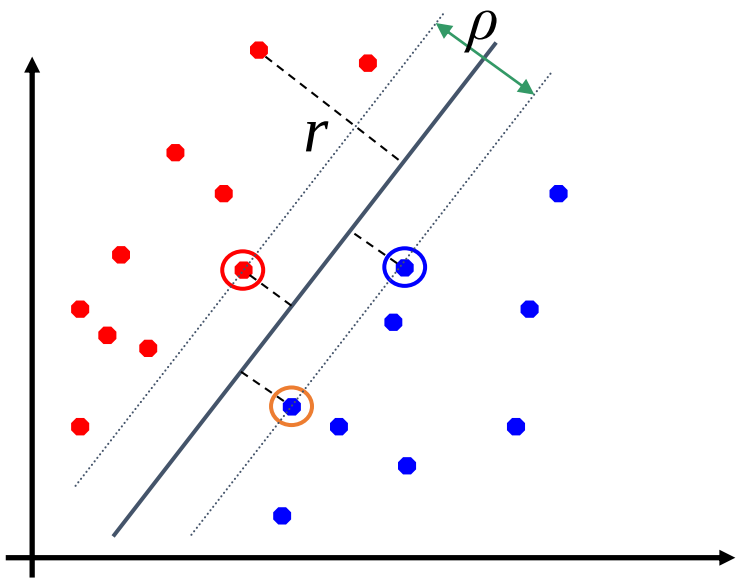
- Supervised learning:
  - Example inputs (features) and their desired outputs (labels)
  - Goal: learn a general rule that maps inputs to outputs
  - SVM, neural networks, etc.
- Unsupervised learning:
  - No labels
  - Find structure in its input
  - Goal: discover hidden patterns in data
  - Clustering, K-means, etc.
- Reinforcement learning:
  - Interact with a dynamic environment (such as driving a vehicle or playing a game against an opponent)
  - Feedback in terms of rewards and punishments as it navigates its problem space
  - Active learning, Q-learning, etc.





Supervised Learning: SVM

- Distance from sample $\mathbf{x}_i$ to the separator: *r*
- **Support vectors:** samples closest to the hyperplane
- **Margin** *ρ:* the distance between support vectors
- **Objective:** maximize the margin *ρ*



$$r = \frac{y_i(w^T x_i + b)}{|w|} = \frac{1}{|w|}$$

$$\rho = \frac{2}{|w|}$$

$$y = -1 : w^T x_i + b \leq -\frac{\rho}{2}$$

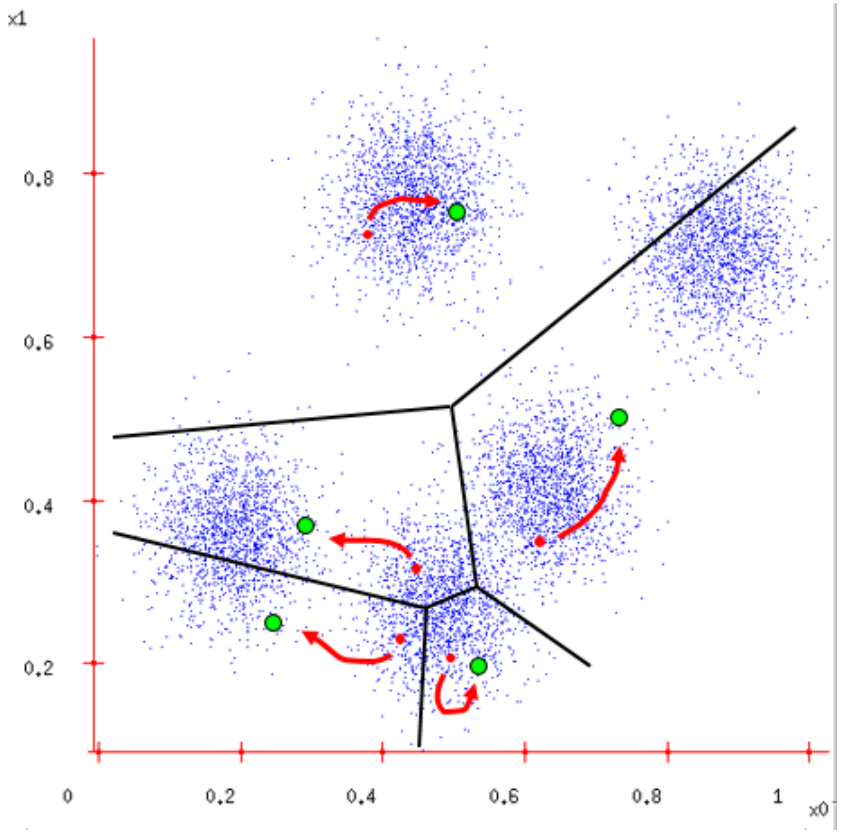
$$y = 1 : w^T x_i + b \geq \frac{\rho}{2}$$

Supervised Learning: SVM Applications

- The best performers for a number of classification tasks ranging from text to genomic data.
- Complex data types beyond feature vectors (e.g. graphs, sequences, relational data) by designing kernel functions for such data.
- Extend to a number of tasks such as regression [Vapnik *et al.* ’97], principal component analysis [Schölkopf *et al.* ’99], etc.

Unsupervised Learning: K-Means

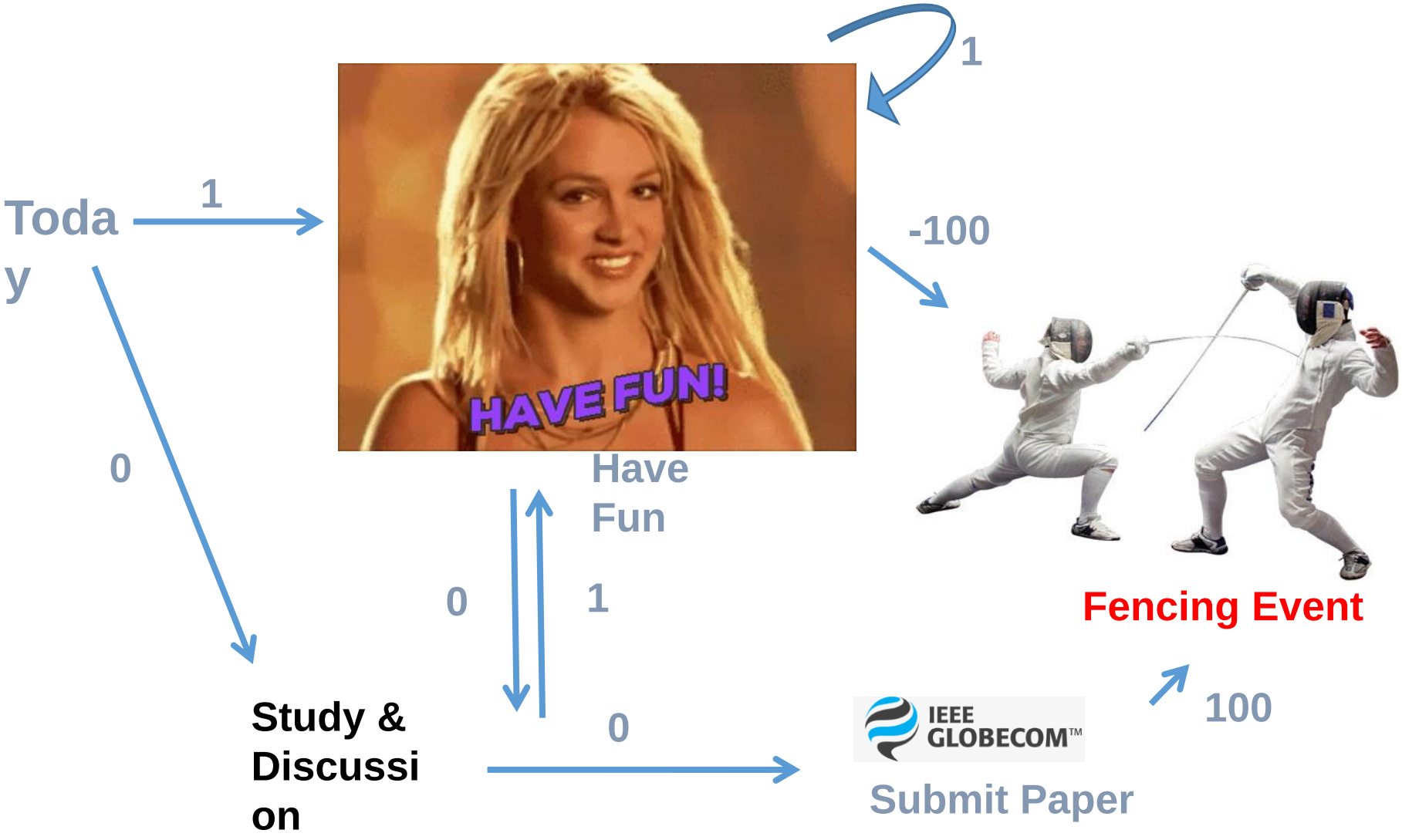
- Ask user how many clusters they'd like. *(e.g. k=5)*
- Randomly guess k cluster center locations
- Each data point: find out which venter it's closest to
- Each center: find the centroid of the points it owns
- Change center
- Repeat until terminated



Unsupervised Learning: K-Means Applications

- Data mining
- Acoustic data in speech understanding to convert waveforms into one of k categories (known as Vector Quantization or Image Segmentation)
- Also used for choosing color palettes on old fashioned graphical display devices and Image Quantization

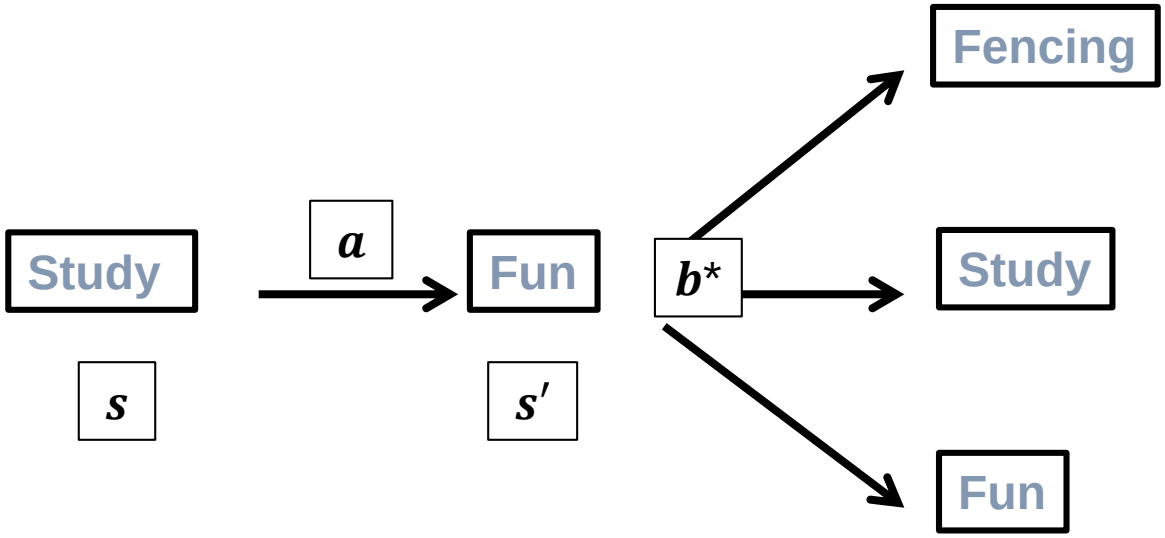
Reinforcement Learning: Q-Learning



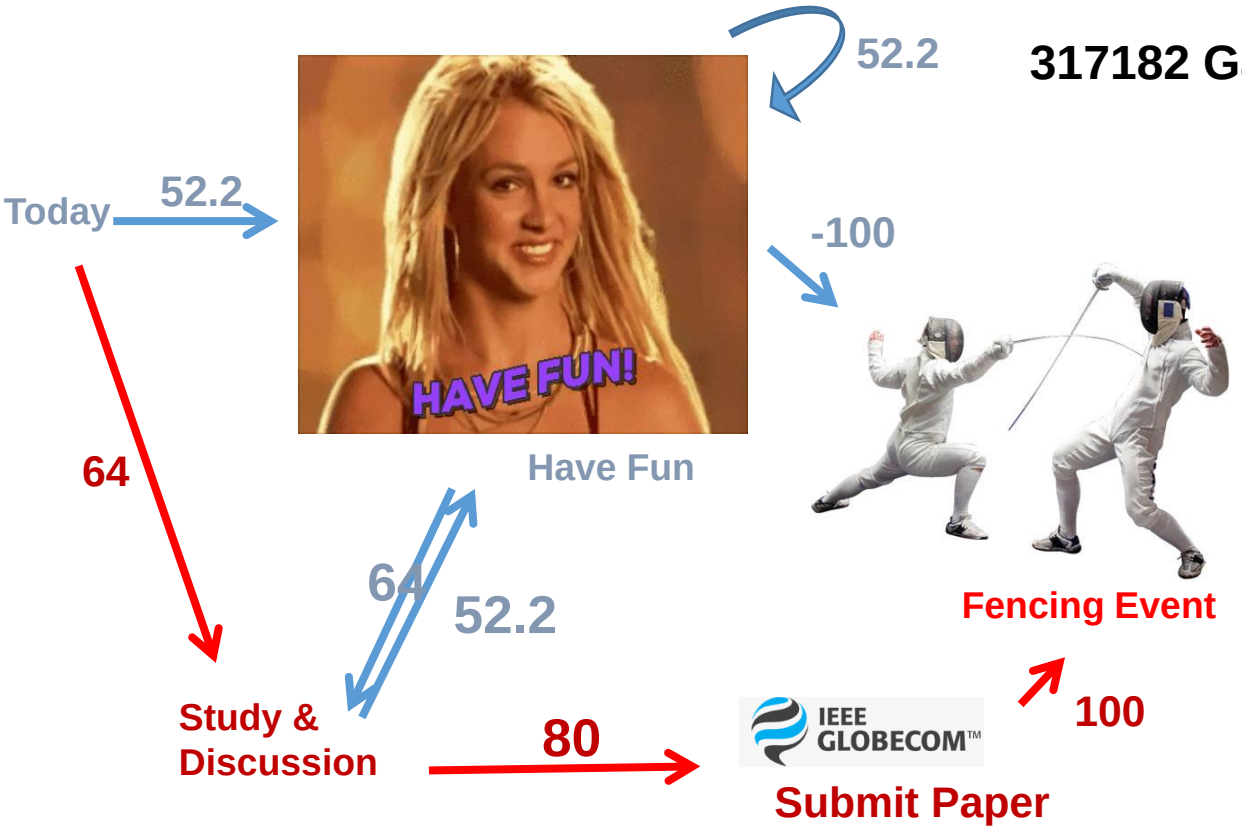
Reinforcement Learning: Q-Learning

Learning Rate Discount Factor

$$Q_{t+1}(s,a) = (1-\alpha_t)Q_t(s,a) + \alpha_t[R_t(s,a) + \beta \max_b Q_t(s',b)]$$



Reinforcement Learning: Results



- 1 million steps
- Learning rate: 0.9999954
- Discount rate: 0.8
- Epsilon: 0.1

Reinforcement Learning: Q-Learning Applications

- Online learning and recommendations systems
- Games: Alphago, Texas Holdem
- Reinforcement Learning in Vehicular Networks

Learning Methods

- Classical Machine Learning
- Bayesian Nonparametric learning
- Deep Learning

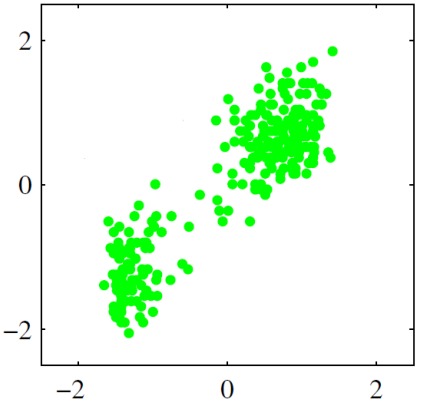
Nam Tuan Nguyen, Rong Zheng, and Zhu Han, “On Identifying Primary User Emulation Attacks in Cognitive Radio Systems Using Nonparametric Bayesian Classification,” IEEE Transactions on Signal Processing, vol. 60, no.3, p.p. 1432- 1445, March 2012

Nam Tuan Nguyen, Rong Zheng, Jie Liu and Zhu Han, “GreenLocs: An Energy Efficient Indoor Place Identification Framework,” ACM Transactions on Sensor Networks, vol. 11, no. 3, article 43, February 2015

Nam Tuan Nguyen, Kae Won Choi, Lingyang Song, and Zhu Han, “Roommates: An Unsupervised Indoor Peer Discovery Approach for LTE D2D Communications,” to appear IEEE Transactions on Vehicular Technology.

Bayesian Nonparametric Learning

- For multi-dimension data
  - Model selection: The number of clusters
  - The hidden process created the observations
  - Latent parameters of the process



- Classic parametric methods (e.g. K-Means)
  - Need to estimate the number of clusters
  - Can have huge performance loss with poor model
  - Cannot scale well

- Nonparametric Bayesian Learning
  - **Nonparametric:** Number of clusters (or classes) can grow as more data are observed and need not to be known as a priori.
  - **Bayesian Inference:** Use Bayesian rule to infer about the latent variables.

Bayesian rule

Posterior Likelihood Prior

$p(\mu|Observation)=p(Observation|\mu)p(\mu)/p(Observation)$

- μ : contain information such as how many clusters, and which sample belongs to which cluster
- μ : nonparametric

Sample the posterior distribution $P(\mu|Observations)$, and get values of the parameter μ.

Bayesian Nonparametric Learning: Example

- A Beta distribution as prior:

$$\text{Beta}(\mu|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\mu^{a-1}(1-\mu)^{b-1}$$

  - Example: a=2, b=2 (head and tail prob. are equal)

- A Binomial distribution as conjugate likelihood:

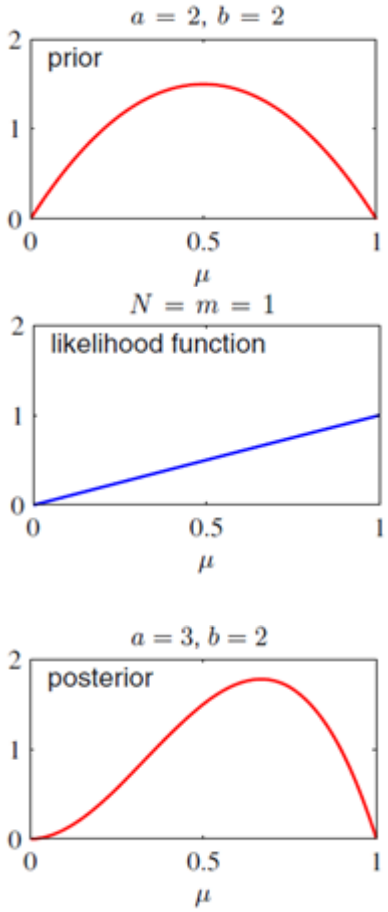
$$\text{Bin}(m|N,\mu) = \binom{N}{m}\mu^{m}(1-\mu)^{N-m}$$

  - One trial (N=1) and the result is one head (m=1)

- Lead to the Posterior:

$$p(\mu|m,l,a,b) = \frac{\Gamma(m+a+l+b)}{\Gamma(m+a)\Gamma(l+b)}\mu^{m+a-1}(1-\mu)^{l+b-1}$$

  - Update of parameters given the observations
  - Higher probability of head



Dirichlet Process

- An extension of the Beta distribution to multiple dimensions

$$Dir(\alpha_1, \ldots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} \pi_i^{\alpha_1 - 1}$$

$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{K} \alpha_i)}$$

  - K: number of clusters
  - πi: weight with marginal distribution
  - α_i: prior

$$Beta(\alpha_i, \sum_{j \neq i} a_j)$$

- **Dirichlet Process:** *G ~ DP(α,H),* if for every finite measurable partition *$A_1, \ldots, A_K$ of Θ*

$$(G(A_1) \cdots G(A_K)) \sim Dir(\alpha H(A_1), \cdots, \alpha H(A_K)) \qquad E[G(A)] = H(A)$$

  - *H(·):* the mean of the DP
  - *α:* strength of the prior

Bayesian Nonparametric Update

- Have *t* observation $x_1,\ldots,x_t$ Define $n_i = \#\{i : x_t \in A_i\}$
- The posterior distribution on Θ

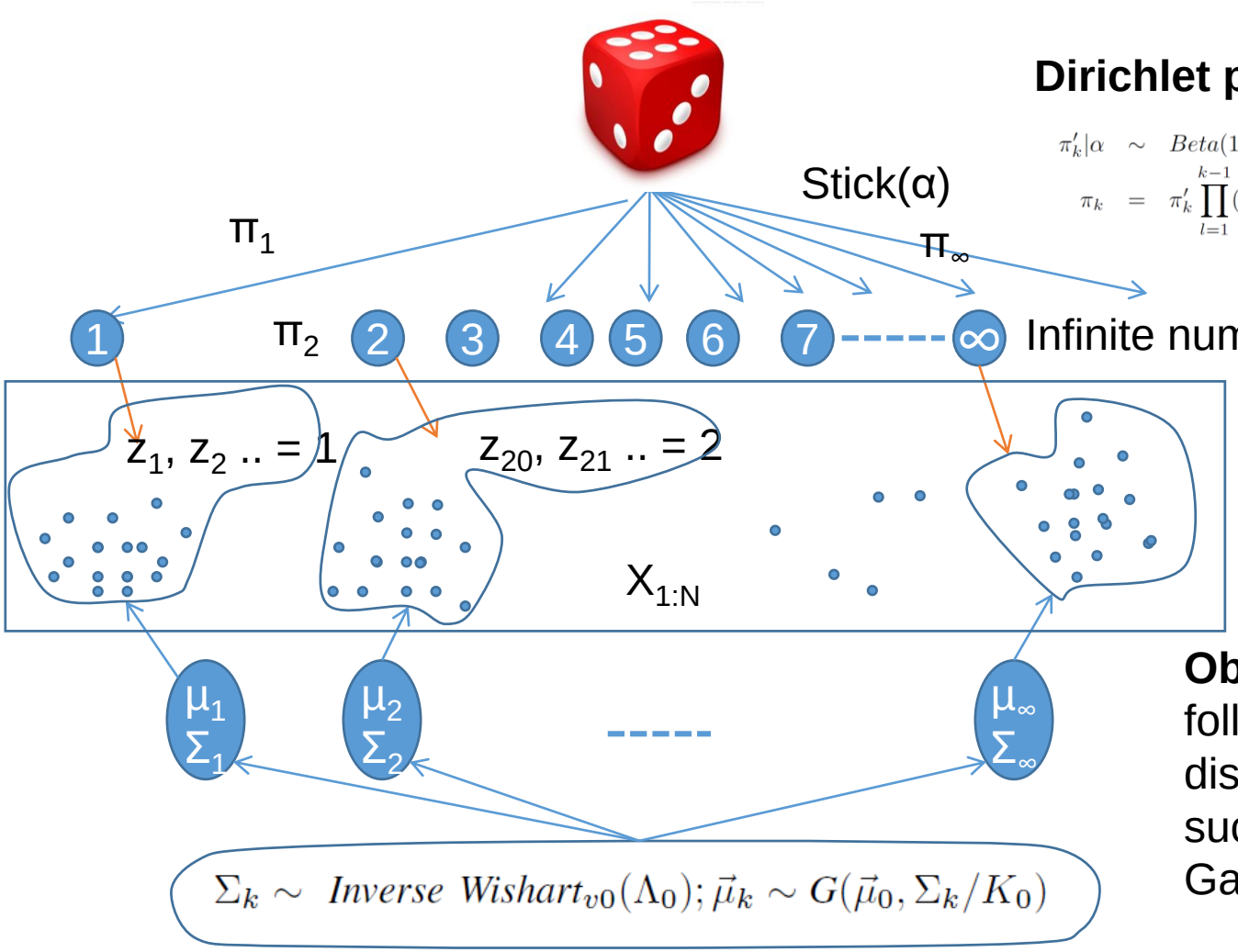
$$(G(A_1)\cdots G(A_K)) \mid x_1 \cdots x_t \sim Dir(\alpha H(A_1)+n_1,\cdots,\alpha H(A_K)+n_K)$$

- The posterior Dirichlet process

$$G \mid x_1 \cdots x_t \sim DP(\alpha + t, \frac{\alpha}{\alpha + t} H + \frac{t}{\alpha + t} \frac{n}{t})$$

  - Small number of observation t, the prior dominates
  - t increases, the prior has less and less impact
  - α: control the balance between the impact of prior and trials
  - Learn and combine any distributions

Infinite Gaussian Mixture Model



Dirichlet process

$$\pi'_k|\alpha \sim Beta(1,\alpha)$$
$$\pi_k = \pi'_k \prod_{l=1}^{k-1}(1-\pi'_l),\ k \to \infty$$

Indicators
created according to multinomial distribution.

Observations:
follow a distribution such as Gaussian.

Chinese Restaurant Process

- Goal: $P(z_i = k|\vec{Z}_{-i}, \alpha, \vec{\theta}, \vec{H}, \vec{X}) = P(z_i = k|\vec{Z}_{-i}, \alpha, \vec{\theta}_k, \vec{H}, \vec{x}_i)$

$$\sim P(\vec{x}_i|\vec{\theta}_k)P(z_i = k|\vec{Z}_{-i}, \alpha),$$

$\vec{Z}_{-i}$: set of all other labels except the current one, i[th]

Posterior

Likelihood (e.g. given as Gaussian)

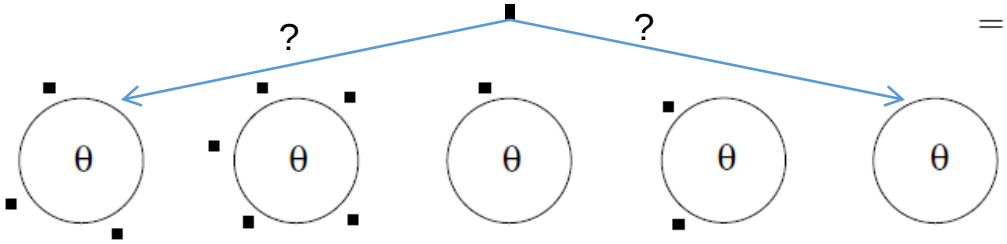
Prior (Chinese Restaurant Process)

Probability assigned to a represented class

$$P(z_i = k|\vec{Z}_{-i}, \alpha) = \frac{n_{k,-i}}{\alpha + N - 1}$$

Probability assigned to an unrepresented class

$$P(z_i \neq z_j, \forall j \neq i|z_{-i}, \alpha) = 1 - \frac{\sum_{j=1}^{K} n_{j,-i}}{\alpha + N - 1} = \frac{\alpha}{\alpha + N - 1}.$$



$n_{k,-i}$:the number of observations in the same class, k, excluding the current one, i[th]

Inference model: Posterior Distributions

- Given the prior and the likelihood, the posterior:
  - Probability of assigning to a unrepresented cluster:

$$P \quad (z_i \neq j, \forall j \neq i | Z_{-i}, X; \alpha, \vec{H}) \qquad (1)$$
$$\sim \quad \frac{\alpha}{\alpha + N - 1} t_{\upsilon_0 - D + 1}(\vec{\mu}_0, \Lambda_0(\kappa_0 + 1)/(\kappa_0(\upsilon_0 - D + 1))).$$

t is the student-t distribution

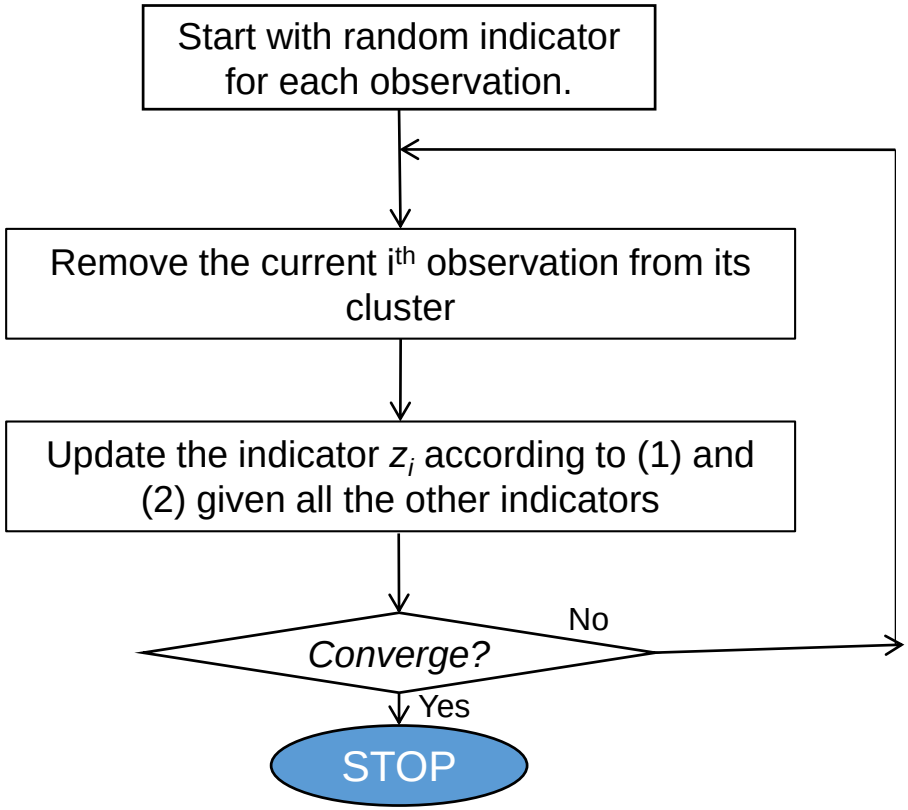
  - Probability of assigning to a represented cluster:

$$P(z_i = k | Z_{-i}, X; \alpha, \vec{H}) \qquad (2)$$
$$\sim \quad \frac{n_{k,-i}}{\alpha + N - 1} t_{\upsilon_n - D + 1}(\vec{\mu}_n, \Lambda_n(\kappa_n + 1)/(\kappa_n(\upsilon_n - D + 1))).$$

Intuitive:
Provide a stochastic gradient!

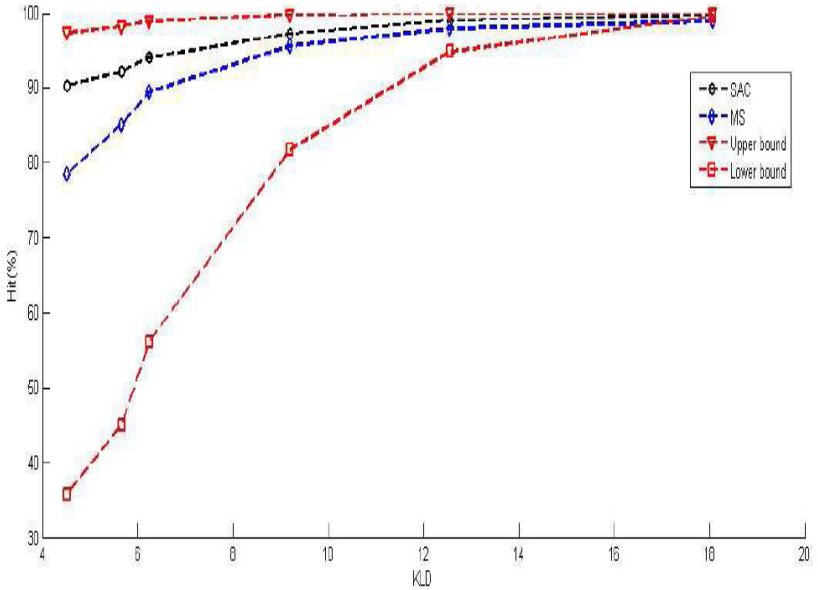
$$\begin{aligned}
\vec{\mu}_n &= \frac{\kappa_0}{\kappa_0 + N}\vec{\mu}_0 + \frac{N}{\kappa_0 + N}\bar{X}, \\
\kappa_n &= \kappa_0 + N, \ \nu_n = \nu_0 + N, \\
\Lambda_n &= \Lambda_0 + S + \frac{\kappa_0 n}{\kappa_0 + N}(\bar{X} - \vec{\mu}_0)(\bar{X} - \vec{\mu}_0)^T, \\
\bar{X} &= (\vec{x}_1 + \vec{x}_2 + \cdots + \vec{x}_N)/N,
\end{aligned}$$

Inference model: Gibbs sampler



Chinese Restaurant Process: Results

- Two Gaussian distributed clusters with KL divergence (KLD) 4.5



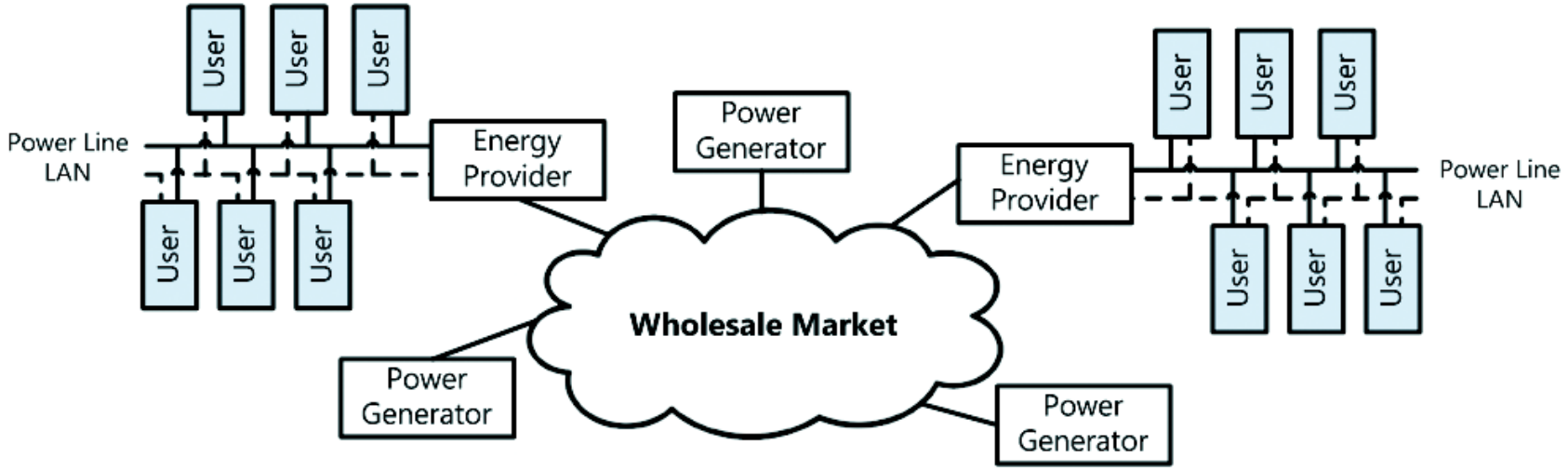
- Intuition why it works so well
  - Not the boundary or threshold. But clustering so that each cluster looks more like the distribution (Gaussian)
  - No prior information on probabilities

Bayesian Nonparametric Learning: Applications

- Smart grid
- Security for wireless devices
- Location based services

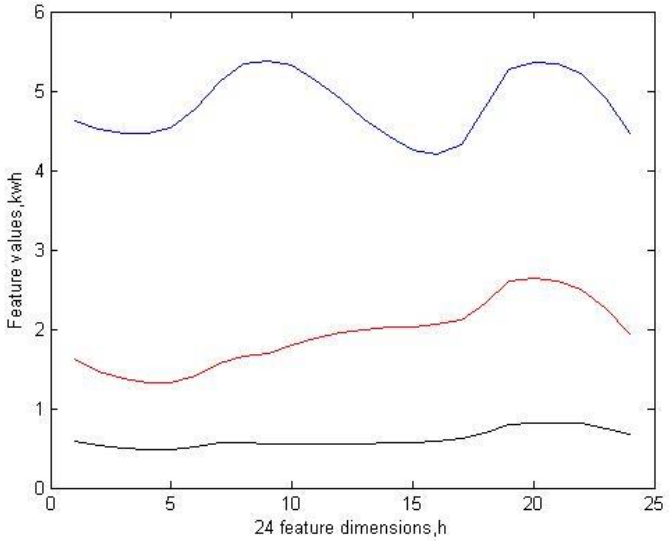
Smart Pricing for Maximizing Profit

- The profit = sum of utility bill – cost to buy power
  - Different shape of loads cost different
  - Incentive using pricing to change the loads
  - The cost reduction is greater than loss of bills



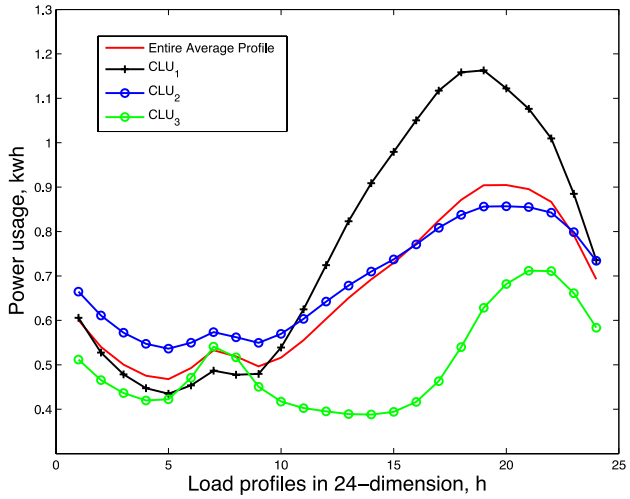
Load Profiling

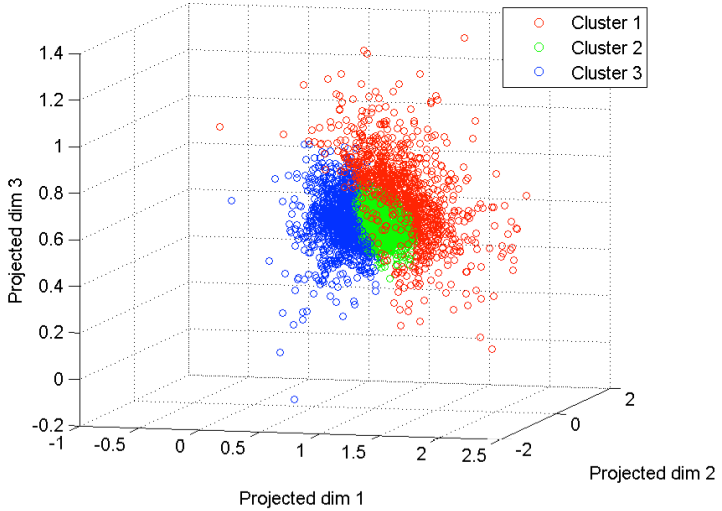
- From smart meter data, try to tell users' usage behaviors
  - CEO, 1%, audience here
  - Worker, middle class, myself
  - Homeless, slave, Ph.D. students



Load Profiling Results

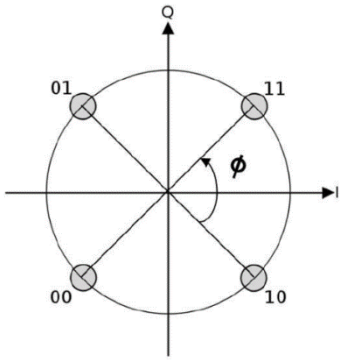
- Utility company wants to know benchmark distributions
  - Nonparametric: do not know how many benchmarks
  - Bayesian: posterior distribution might be time varying
  - Scale: Daily, weekday, weekend, monthly, yearly

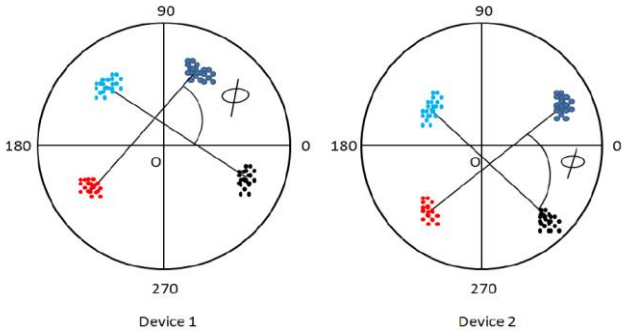




Security for Wireless Devices

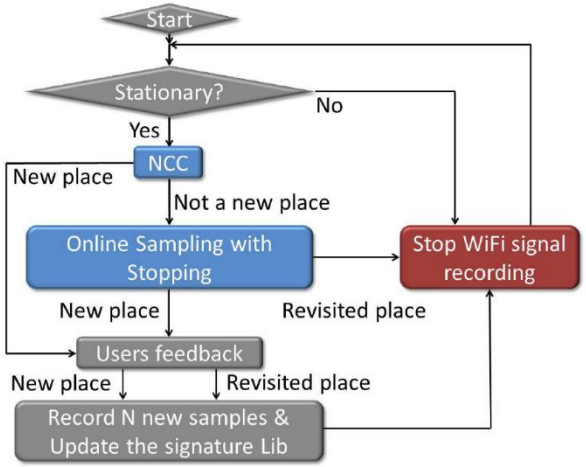
- Prime User emulation (PEU) attack detection
  - In Cognitive radio, a malicious node can pretend to be a Primary User (PU) to keep the network resources (bandwidth) for his own use
  - Collect device dependent fingerprints and classify the fingerprints
    - The Carrier Frequency Difference (CFD)
      - Defined as the difference between the carrier frequency of the ideal signal and that of the transmitted signal.
      - Depends on the oscillator within each device.
    - The Phase Shift Difference (PSD)
      - Using QPSK modulation technique.
      - Transmitter amplifiers for I-phase and Q-phase might be different.





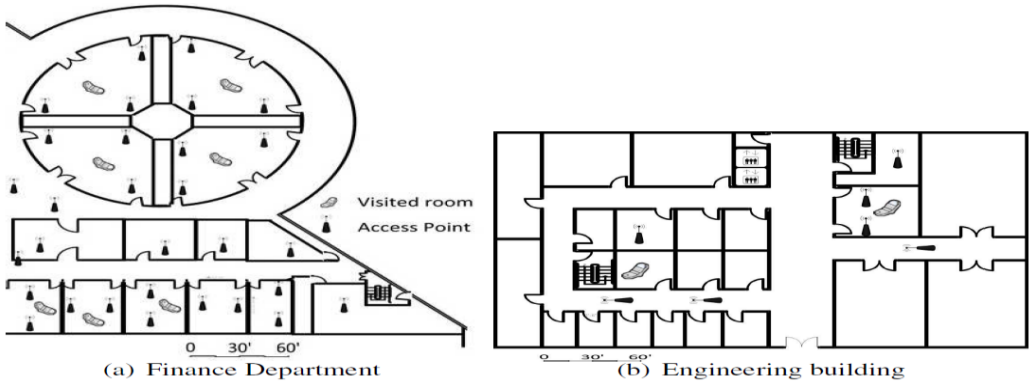
Location Based Services

- Major tasks to enable LBS:
  - Localize.
  - Estimate dwelling time
  - Prediction: Where to go next?
- What's given:
  - Mobile devices are in indoor environments
  - WiFi scans
- Goals:
  - Identifying revisited location
  - Automatically profiling new location
  - Online sampling to reduce the complexity
  - Predicting the next possible locations

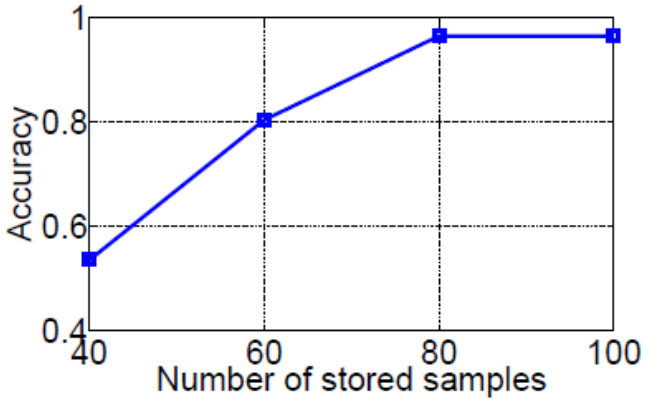


Location Based Services: Experimental Results

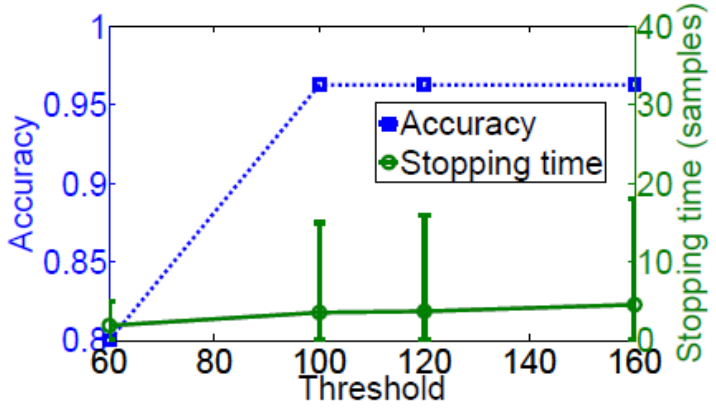
- Dataset:
  - 4 weeks long
  - 5 phones



(a) Finance Department (b) Engineering building



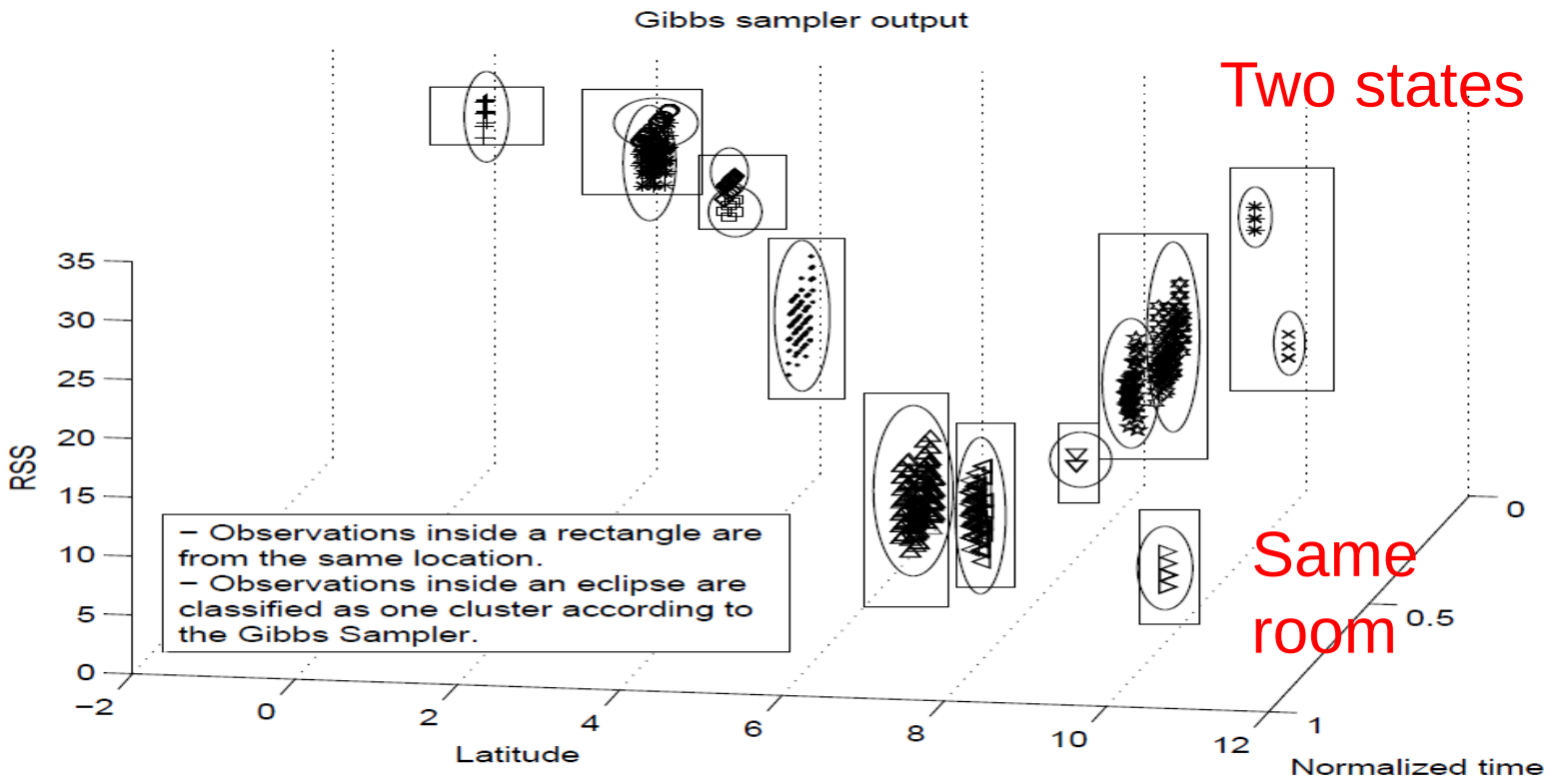
(a) Accuracy vs N.



(b) Accuracy and stopping time vs $c_+(-c_-)$

Location Based Services: Experimental Results

- Future Location Prediction



Learning Methods

- Classical Machine Learning
- Bayesian Nonparametric learning
- Deep Learning

Mohammad Abu Alsheikh, Dusit Niyato, Shaowei Lin, Hwee-Pink Tan, and Zhu Han, "Mobile Big Data for Context-Awareness Using Deep Learning and Apache Spark," IEEE Network Magazine, special issue on Mobile Big Data, vol. 30, no. 3, pp. 22-29, May-June 2016.

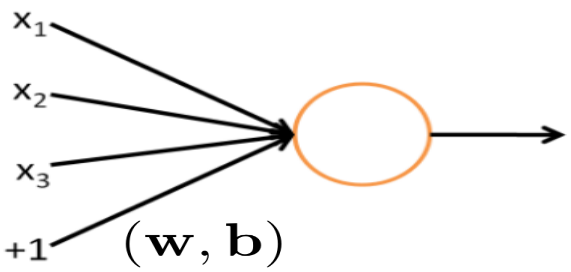
Xunsheng Du, Huaqing Zhang, Hien Van Nguyen, and Zhu Han, "Stacked LSTM Deep Learning Model for Resource Allocation in Vehicle-to-Vehicle Communication," IEEE 86th Vehicular Technology Conference, Fall, Toronto, Canada, September 2017.

Erte Pan and Zhu Han, "Non-parametric Bayesian Learning with Deep Learning Structure and Its Applications in Wireless Networks," IEEE globalsip, Atlanta, December, 2014.

Nam Tuan Nguyen, Yichuan Wang, Husheng Li, Xin Liu and Zhu Han, "Extracting Typical Users' Moving Patterns Using Deep Learning," IEEE Globe Communication Conference, Anaheim, CA, December 2012.

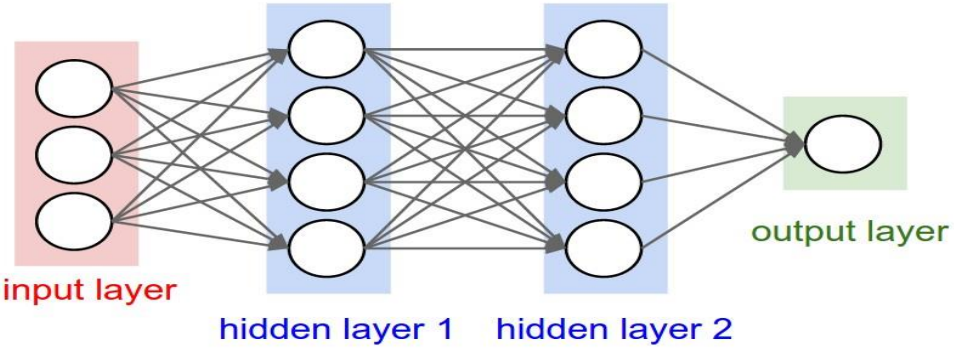
Deep Learning: Basic Idea

- Add Hidden Layers in Neural Networks



$$f(\mathbf{w}^T\mathbf{x} + b)$$

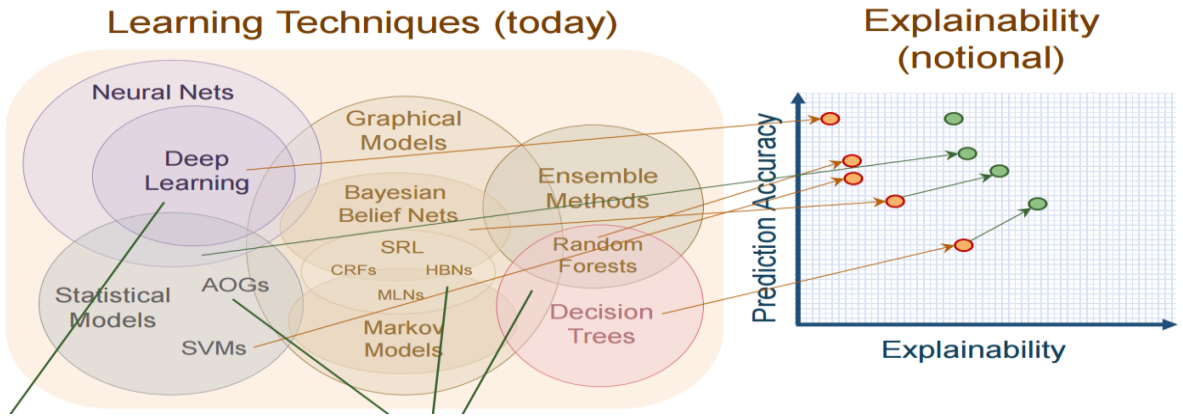
Linear transformation followed by non-linear rectification



- More parameters
- More non-linear parts

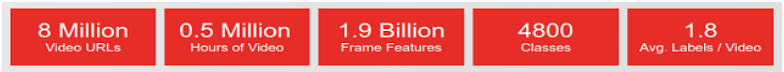
Deep Learning: Motivations

- Classic Methods
  - Do not have a lot of data, or
  - Training data have categorical features
  - A more explainable model
  - A high run-time speed
- Deep Learning
  - A lot of training data of the same or related domain
  - Improve Domain Adaptation
  - Much slower
  - Appropriate scaling and normalization have to be done



Why Now?

- Big annotated datasets become available:
  - ImageNet:
  - Google Video:
  - Mechanical Turk
  - Crowdsourcing



- GPU processing power

- Better stochastic gradient descents:
  - AdaGrad, AdamGrad, RMSProp

Typical Deep Neural Networks

- Convolutional Neural Networks (CNNs)
- Recurrent Neural Networks (RNNs)
- Deep Belief Networks

Convolutional Neural Networks (CNNs)

INPUT IMAGE

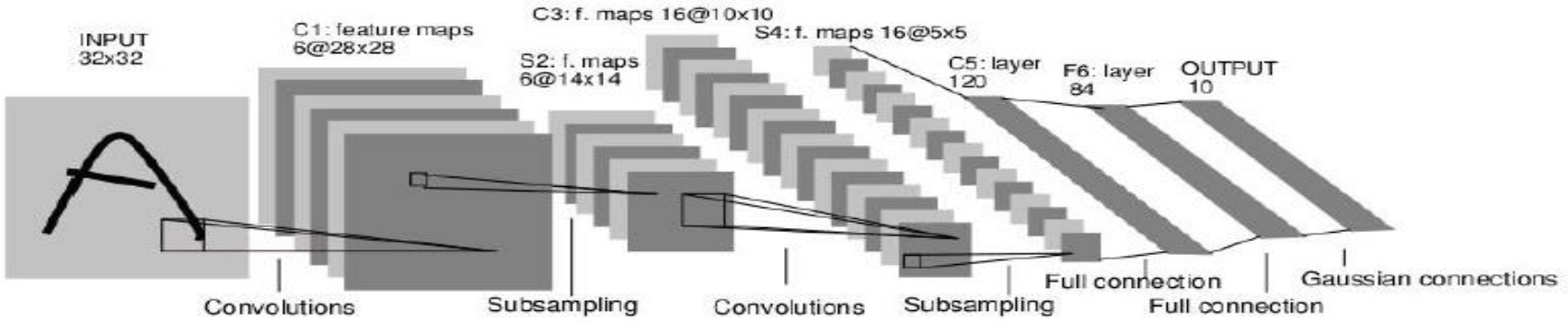
CONVOLUTION (LEARNED)

NON-LINEAR RECTIFIER

POOLING

FEATURE MAPS

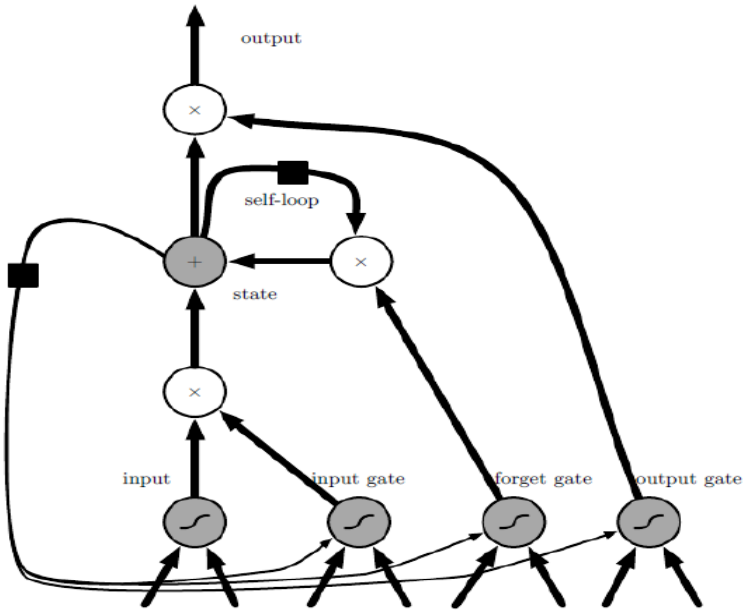
Work well in image processing!



LeCun, Yann. "LeNet-5, convolutional neural networks". Retrieved Nov. 2013.

Recurrent Neural Networks (RNNs)

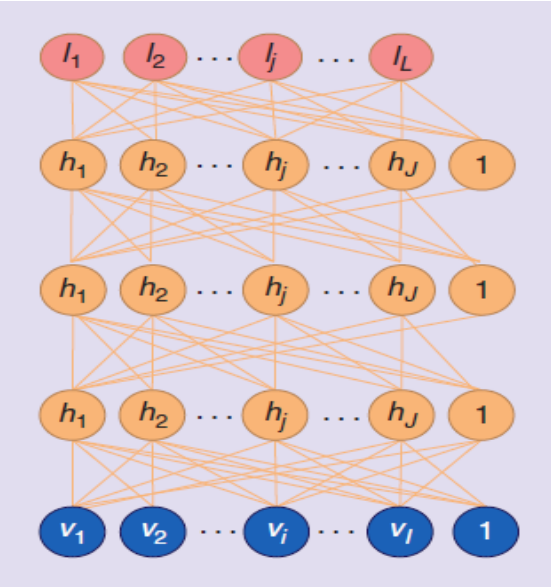
- Produce an output at each time step and have recurrent connections between hidden units
  - Long Short-Term Memory

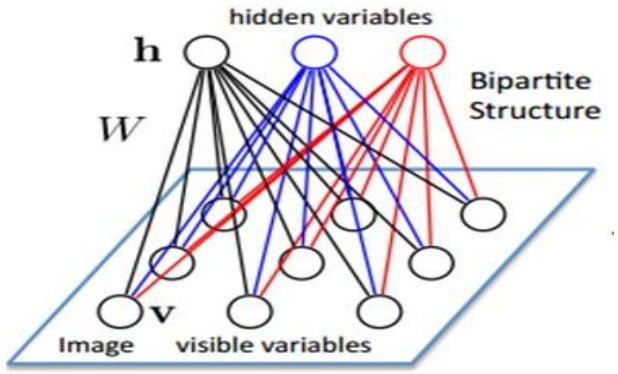


- Unconstrained handwriting recognition (Graves et al., 2009),
- Speech recognition (Graves et al., 2013; Graves and Jaitly, 2014)
- Handwriting generation (Graves, 2013),
- machine translation (Sutskever et al., 2014) Image captioning (Kiros et al., 2014;Vinyals et al., 2014; Xu et al., 2015)
- Parsing (Vinyals et al., 2014a).

Deep Belief Networks

- Each link associates with a probability
- Parametric





The energy of the joint configuration:

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

$\theta = \{W, a, b\}$ model parameters.

- Applied in clustering

G. E. Hinton, S. Osindero, and Y. W. Teh, "A Fast Learning Algorithm for Deep Belief Nets", *Neural Computation*. Vol. **18**, No. 7, pp.1527–1554, 2006.

Comparison

	Similarities	Differences
Convolutional Neural Networks	1. Multiple Layers 2. Use Back-propagation Algorithm for training 3. Can be combined together to create more powerful networks	1. More suitable for data with grid structures 2. Much fewer parameters 3. Very efficient training with GPUs
Recurrent Networks		1. Having memory of past (suitable for tasks like speech recognition) 2. Not able to take big input such as images or videos
Deep Belief Networks		1. Generative model (can generate realistic looking data after initializing at random variable) 2. Used much less due to inefficiency

Applications

- Computer Vision
  - Object detection
  - Add & repair missing details in high-resolution photo
- Speech Recognition
  - Enable a text-to-speech system that is *almost* in distinguishable from human voice
  - Compose classical music that likely to be created by a human
- Natural Language Processing
- Online Recommendation Systems
- Automatic Driving
  - Google Driverless Car
- Wireless Resource Allocation
- Vehicular Communication Network
- Smart Grid

Summary

- Classical Machine Learning
- Bayesian Nonparametric learning
  - Non-parametric
  - Smart grid & Location Based Services
- Deep Learning
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Deep Belief Networks

Content

- Overview of Big Data
- Learning Methods
- Commercial Systems
- Large Scale Optimization
- Game Theory based Approaches

Commercial Systems

- TensorFlow
- MapReduce

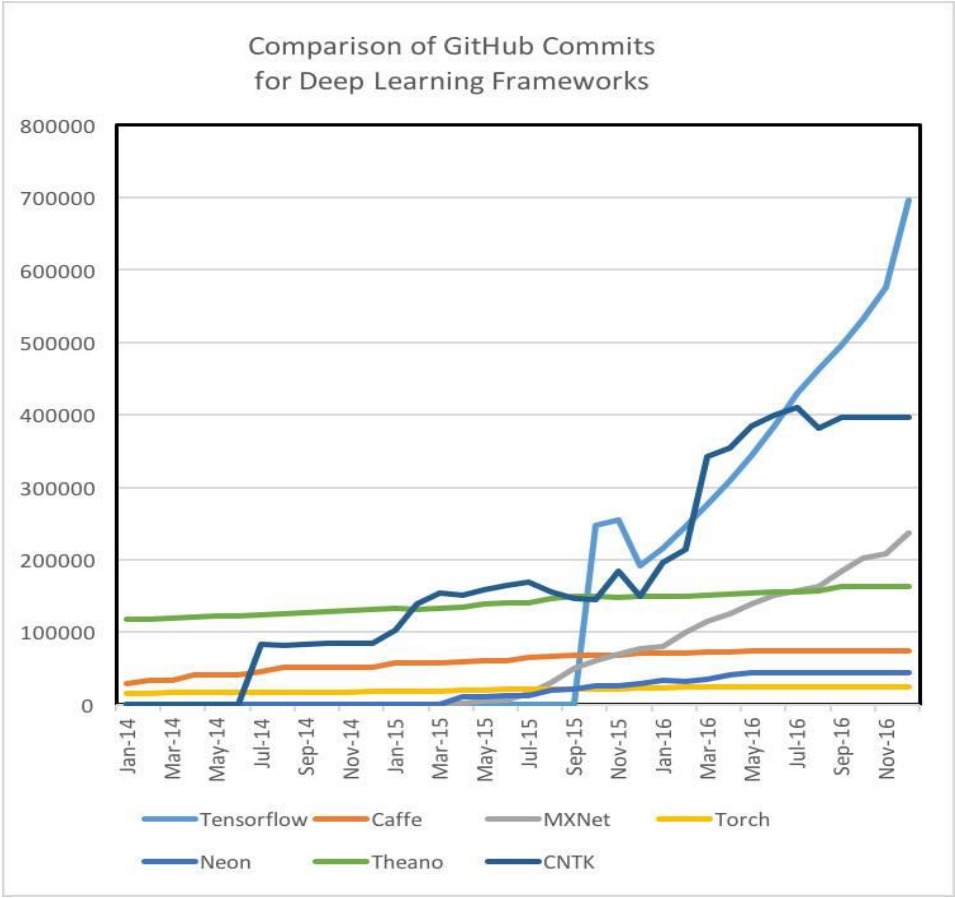
TensorFlow

- A deep learning library open-sourced by Google
- Originally developed by the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well
- Provide primitives for defining functions on tensors and automatically computing their derivatives.

https://www.tensorflow.org/

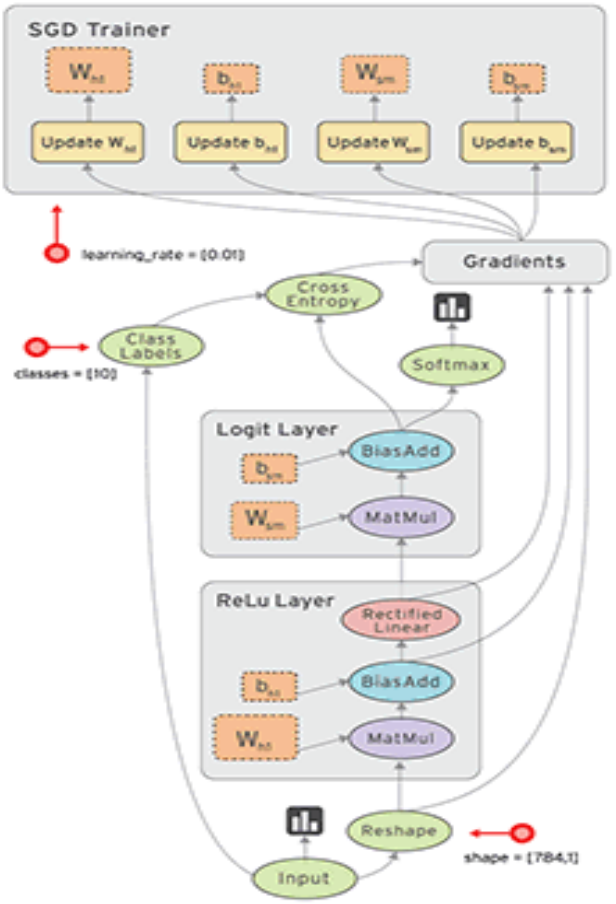
Comparison on Deep Learning Libraries

- TensorFlow
  - Google
- Caffe
  - UC-berkeley
- MXNet
- Theano
  - Université de Montréal
- Torch
  - Facebook
- Microsoft CNTK
- Neon
  - Intel



How Does It Work?

- Describe mathematical computation with a directed graph of nodes & edges
  - Nodes in the graph represent mathematical operations
  - Edges describe the i/o relationships between nodes
  - Data edges carry dynamically-sized multidimensional data arrays, or **tensors**



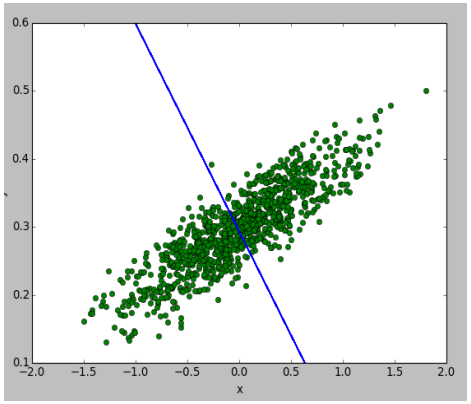
How to Use It?

- Programming with APIs
  - Define features and labels, loss functions, select algorithm, set iteration time…

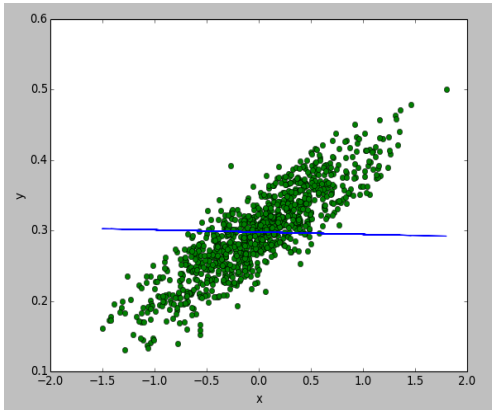
```python
import numpy as np
import matplotlib.pyplot as plt

# number of random points
num_puntos = 1000
# list of random points (pair of x and y)
conjunto_puntos = []

for i in xrange(num_puntos):
    x1 = np.random.normal(0.0,0.55)
    y1 = x1*0.1 + 0.3 + np.random.normal(0.0,0.03)
    conjunto_puntos.append([x1,y1])

x_data = [v[0] for v in conjunto_puntos]
y_data = [v[1] for v in conjunto_puntos]


# plot the those random points in 2D
plt.plot(x_data,y_data,'go',label='Original data')
plt.legend()
plt.show()
```

```python
#--------------------fit  points-----------------------------#
W = tf.Variable(tf.random_uniform([1],-1.0,1.0))
b = tf.Variable(tf.zeros([1]))
y = W * x_data + b

loss = tf.reduce_mean(tf.square(y-y_data))
optimizer = tf.train.GradientDescentOptimizer(0.5)
train = optimizer.minimize(loss)

init = tf.initialize_all_variables()

sess = tf.Session()
sess.run(init)

#--------------------plot fitting line-----------------------#
for step in xrange(16):
    sess.run(train)

    plt.plot(x_data,y_data,'go')
    plt.plot(x_data,sess.run(W) * x_data + sess.run(b))
    plt.xlabel('x')
    plt.xlim(-2,2)
    plt.ylim(0.1,0.6)
    plt.ylabel('y')
    plt.legend()
    plt.show()
```

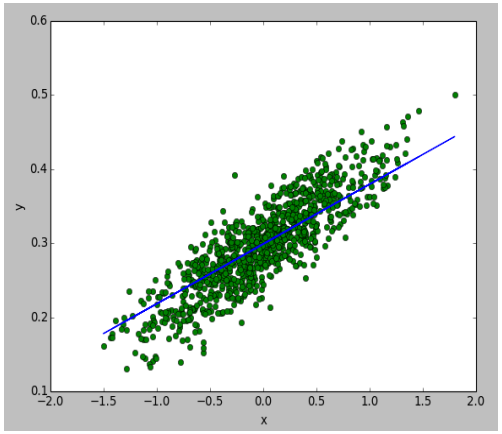
Example: Linear Regression



1 iteration



8 iterations



16 iterations

Commercial Systems

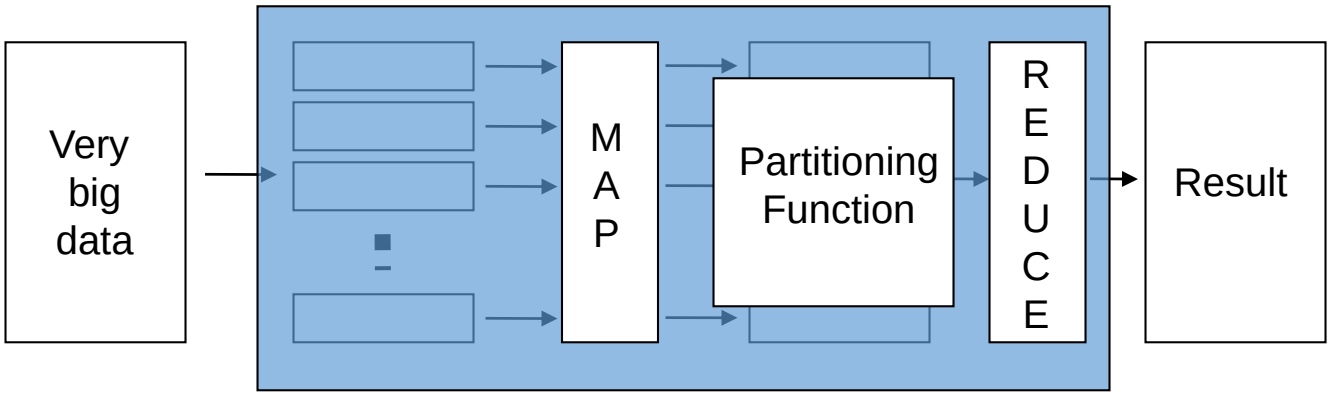
- TensorFlow
- MapReduce

MapReduce: Basic Idea

- Parallel programming model meant for large clusters
  - User implements Map() and Reduce()
- Parallel computing framework
  - Libraries take care of EVERYTHING else
    - Parallelization
    - Fault Tolerance
    - Data Distribution
    - Load Balancing
- Useful model for many practical tasks (large data)

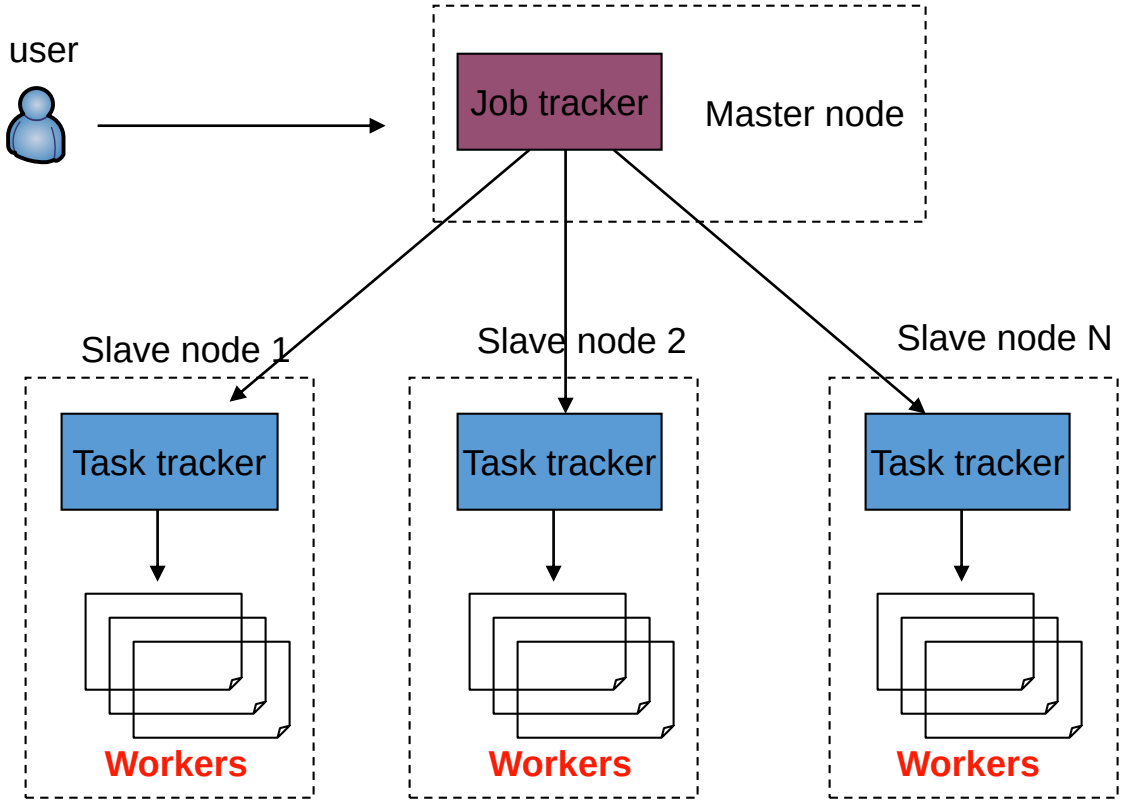
http://hadoop.apache.org/

Processing



- Map:
  - Accepts *input* key/value pair
  - Emits *intermediate* key/value pair
- Reduce :
  - Accepts *intermediate* key/value* pair
  - Emits *output* key/value pair

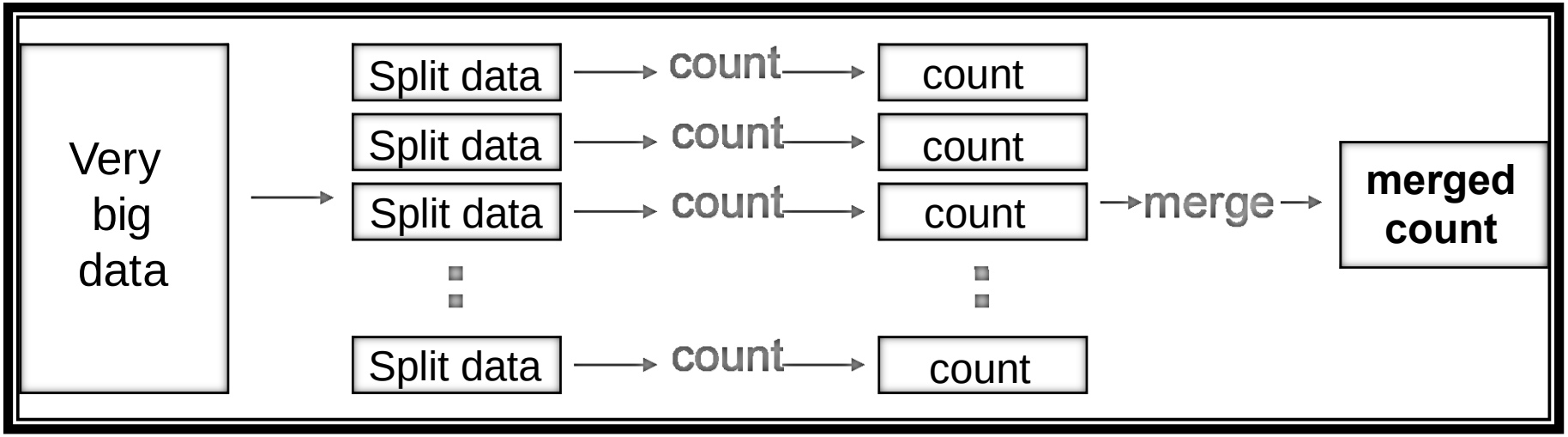
Architecture



Three Steps to Use

- Indicate
  - Input/output files
  - M: number of map tasks
  - R: number of reduce tasks
  - W: number of machines
- Write map and reduce functions
- Submit the job

Example: Word Count



- Map()
  - Input <filename, file text>
  - Parses file and emits <word, count> pairs
    - eg. <"hello", 1>
- Reduce()
  - Sums values for the same key and emits <word, TotalCount>
    - eg. <"hello", (3 5 2 7)> => <"hello", 17>

More Distributed Systems

- Hadoop: Realize MapReduce
  - Offline big data processing
  - Web searching
  - Parallel computing
- Spark: Working on Internal Storage
  - Online big data processing, fast input and output
  - Repeated operation for the same data
- Storm: Data Flow
  - Realtime data streaming

Summary

- TensorFlow
  - Deep Learning Library
- MapReduce
  - Parallel Computing System
  - Hadoop & Spark & Storm

Content

- Overview of Big Data
- Learning Methods
- Commercial Systems
- Large Scale Optimization
- Game Theory based Approaches

Large Scale Optimization

- Multi-block Optimization
- Compressive Sensing
- Sublinear Algorithms

Multi-block Optimization

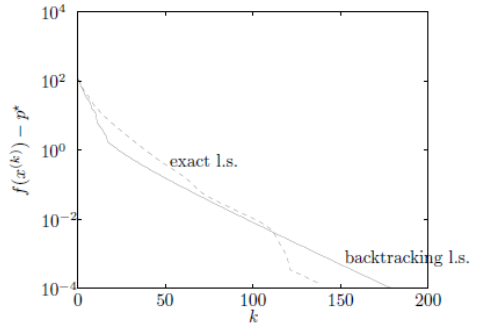
- Generally, optimization algorithms for solving problems of such huge sizes should satisfy:
  - Simple Sub-problem: Each of their computational steps must be simple and easy
  - Parallel Implementable: Algorithm is implementable in distributed and/or parallel manner
  - Fast Convergence: A high-quality solution can be found using a small number of iterations

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{c}$$

$$\min_{\mathbf{x}_1\in\mathcal{X}_1,\mathbf{x}_2\in\mathcal{X}_2} f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \quad \text{s.t.} \quad \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{c}.$$

$$\mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \{\mathbf{x}_j^k\}_{j\neq i}, \boldsymbol{\lambda}^k) + \tfrac{1}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{P}_i}^2$$

$$\varepsilon = o(1/k)$$



S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge Univ., 2004.

Optimization Preliminary: Dual Ascent Methods

- Consider an optimization problem of the form

$$\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{x} = \mathbf{c}$$

The Lagrangian	$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x} - \mathbf{c})$
Dual function	$g(\boldsymbol{\lambda}) = \inf_x \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$
Dual Problem	$\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda})$ Lower bound
Optimal solution	$\mathbf{x}^* = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^*)$
Dual Ascent (Gradient Descent)	$\begin{cases} \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho^k(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{c}) \end{cases}$

KKT(**Karush–Kuhn–Tucker**) Condition – Strong Convexity

Require an appropriate step size ρ and assumptions of strong convexity of the objective function f.

Optimization Preliminary: Method of Multipliers

- Introduce an augmentation $\|Ax - c\|_2^2$ to the Lagrangian:
- The Augmented Lagrangian: $\mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \boldsymbol{\lambda}^\top(\mathbf{A}\mathbf{x} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{c}\|_2^2,$
- Method of Multipliers: $\begin{cases} \mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{c}), \end{cases}$
- Pros:
  - Stable, robust and fast compare with the dual ascent method.
  - No need to tune the parameter ρ during each iteration.
- Cons:
  - Difficult to decouple and parallelize due to the augmentation $\|Ax - c\|_2^2$

Two Block Optimization: ADMM

- ADMM: Alternating Direction Method of Multipliers

(R. Glowinski and A. Marrocco, 1975, D. Gabay and B. Mercier, 1976, S. Boyd, 2011)

- The general form of problem that ADMM can solve is expressed as:

$$\min_{\mathbf{x}_1 \in \mathcal{X}_1, \mathbf{x}_2 \in \mathcal{X}_2} f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) \quad \text{s.t.} \quad \mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 = \mathbf{c}. \qquad (1)$$

- The augmented Lagrangian for (1) is

$$\mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2, \boldsymbol{\lambda}) = f_1(\mathbf{x}_1) + f_2(\mathbf{x}_2) + \boldsymbol{\lambda}^\top(\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}_1\mathbf{x}_1 + \mathbf{A}_2\mathbf{x}_2 - \mathbf{c}\|_2^2,$$

- A Gauss-Seidel iterations of x_1 and x_2 as follows

$$\begin{cases} \mathbf{x}_1^{k+1} = \arg\min_{\mathbf{x}_1} \mathcal{L}_\rho(\mathbf{x}_1, \mathbf{x}_2^k, \boldsymbol{\lambda}^k), \\ \mathbf{x}_2^{k+1} = \arg\min_{\mathbf{x}_2} \mathcal{L}_\rho(\mathbf{x}_1^{k+1}, \mathbf{x}_2, \boldsymbol{\lambda}^k), \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\mathbf{A}_1\mathbf{x}_1^{k+1} + \mathbf{A}_2\mathbf{x}_2^{k+1} - \mathbf{c}). \end{cases}$$

- Global convergence for convex optimization with a convergence rate $o(\frac{1}{k})$

R. Glowinski and A. Marrocco, Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéares, *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, 1975

D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximations, *Computers and Mathematics with Applications*, 1976.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers", *Found. and Trends in Mach. Learning*, vol. 3, no. 1, pp. 1-122, Nov. 2011.

Gauss-Seidel Multi-Block ADMM (Sequential)

- Consider the following convex optimization problem

$$\begin{aligned} \min_{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N} \quad & f(\mathbf{x}) = f_i(\mathbf{x}_i) + \ldots + f_N(\mathbf{x}_N), \\ \text{s.t.} \quad & \mathbf{A}_i\mathbf{x}_i + \ldots + \mathbf{A}_N\mathbf{x}_N = \mathbf{c}, \\ & \mathbf{x}_i \in \mathcal{X}_i, \quad i = 1, \ldots, N. \end{aligned} \tag{2}$$

- The augmented Lagrangian for (2):

$$\mathcal{L}_\rho(\{\mathbf{x}_i\}_{i=1}^N, \boldsymbol{\lambda}) = \sum_{i=1}^N f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^\top(\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{c}) + \frac{\rho}{2}\|\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{c}\|_2^2$$

- A Gauss-Seidel Multi-block ADMM:

$$\begin{cases} \mathbf{x}_i = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\{\mathbf{x}_j^{k+1}\}_{j<i}, \mathbf{x}_i, \{\mathbf{x}_j^k\}_{j>i}, \boldsymbol{\lambda}^k), & i = 1, \ldots, N. \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \rho(\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i^{k+1} - \mathbf{c}). \end{cases}$$

(Block Coordinate Descent…)

M. Hong and Z. Luo, “On the linear convergence of the alternating direction method of multipliers,” online at http://arxiv.org/abs/1208.3922,2012.
C. Chen, B. He, Y. Ye, and X. Yuan, “The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent,” Online at http://web.stanford.edu/~yyye/ADMM 5, 2014.
B. He, M. Tao, and X. Yuan, “Alternating direction method with Gaussian back substitution for separable convex programming,” SIAM Journal of Optimization, vol. 22, no. 2, pp. 313–340, 2012.

Proximal Jacobian Multi-Block ADMM (Parallel)

- Recall the augmented Lagrangian:

$$\mathcal{L}_\rho(\{\mathbf{x}_i\}_{i=1}^N, \boldsymbol{\lambda}) = \sum_{i=1}^N f_i(\mathbf{x}_i) + \boldsymbol{\lambda}^\top (\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{c}) + \frac{\rho}{2}\| \sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i - \mathbf{c}\|_2^2$$

- A proximal term is added to the augmented Lagrangian, and the update of x_i is performed concurrently:

$$\begin{cases} \mathbf{x}_i^{k+1} = \arg\min_{\mathbf{x}_i} \mathcal{L}_\rho(\mathbf{x}_i, \{\mathbf{x}_j^k\}_{j\neq i}, \boldsymbol{\lambda}^k) + \frac{1}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_{\mathbf{P}_i}^2, \\ \boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \gamma\rho(\sum_{i=1}^N \mathbf{A}_i\mathbf{x}_i^{k+1} - \mathbf{c}), \end{cases} \quad \forall i = 1, \ldots, N.$$

Proximal

where $\|\mathbf{x}_i\|_{\mathbf{P}_i}^2 = \mathbf{x}_i^\top \mathbf{P}_i \mathbf{x}_i$ for some symmetric and positive semi-definite matrix $\mathbf{P}_i \succeq 0$.

- The involvement of the proximal term
  - Make subproblem of x_i strictly or strongly convex
  - Ensure the convergence
  - Easier to solve

B. He, L. Hou, and X. Yuan, "On full jacobian decomposition of the augmented lagrangian method for separable convex programming," online at http://www.optimization-online.org/DB HTML/2013/05/3894.html, 2013.
W. Deng, M. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o(1/k) convergence," online at ttp://arxiv.org/abs/1312.3040, 2014.

Additional Issues

- Indecomposable

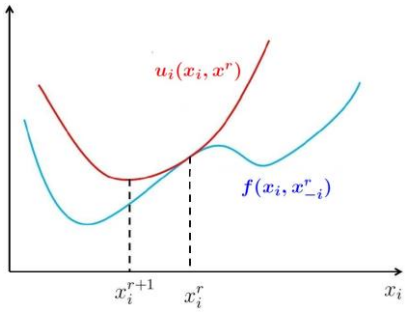
$$f(x) = f(x_1, x_2, ..., x_N) \neq f_1(x_1) + f_2(x_2) + ... f_N(x_N)$$

Decompose to Consensus Problem

$$f(x) = f(x_1, x_2, ..., x_N) \neq f_1(x_1, x_2^1, ..., x_N^1) + f_2(x_1^2, x_2, ...) + ... f_N(x_1^N, x_2^N, ..., x_N)$$

$$s.t. \quad x_i = x_i^j$$

- Non-Convexity
  - Relax to convex problems
  - BSUM(Block Successive Upper Bound Minimization)



- Mixed-Integer
  - Branch and Bound (x_1 not integer => $\lfloor x_1 \rfloor$ & $\lfloor x_1 \rfloor + 1$)

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers”, *Found. and Trends in Mach. Learning*, vol. 3, no. 1, pp. 1-122, Nov. 2011.

M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, “A unified algorithmic framework for block structured optimization involving big data," IEEE Signal Processing Magazine, vol. 33, no. 1, pp. 57 - 77, 2016.
M. Razaviyayn, M. Hong, and Z.-Q. Luo, “A unified convergence analysis of block successive minimization methods for nonsmooth optimization," SIAM Journal on Optimization, vol. 23, no. 2, pp. 1126 - 1153, 2013.

Summary of Methods

- For a non-convex, mixed-integer, indecomposable, large-scale optimization…

Issues	Tricks
Indecomposable	Consensus problem
Non-convexity	Relaxition, BSUM
Mixed-integer	Branch and Bound
Deconvergence in Parrallel Update	Proximal
Slow in Sequential Update	Augmented Lagrangian

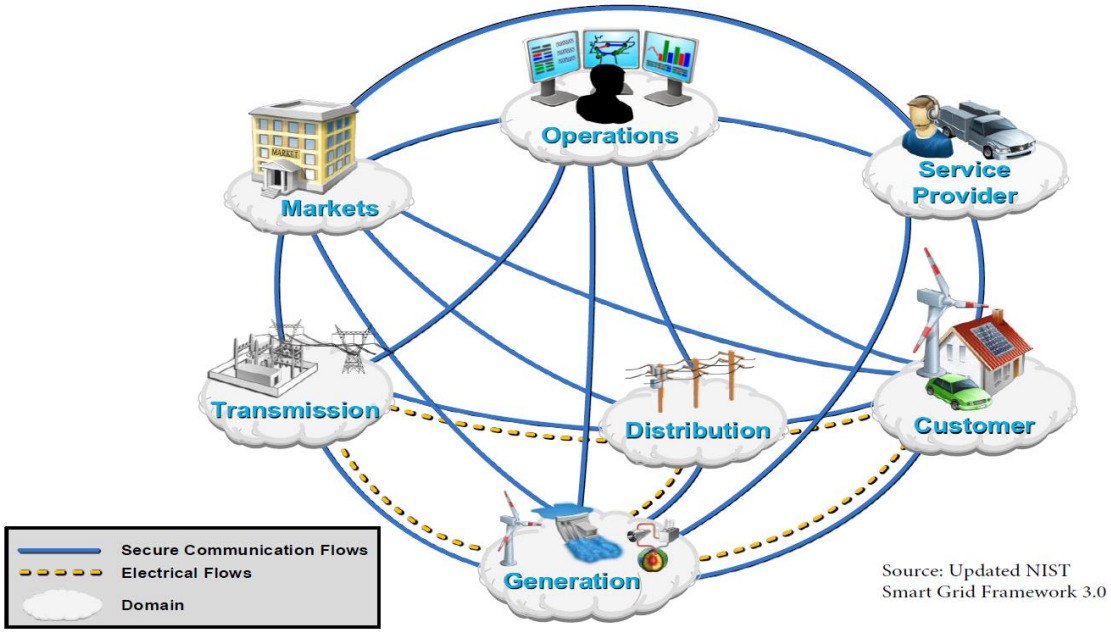
- Before those…
  - Simulations & Proofs case by case
  - Experienced Adaptions

    Multi-block algorithms have their own conditions in mathematics except for convexity.

Multi-Block Optimization: Applications

- Security Constrained Optimal Power Flow
- Data Offloading in Software Defined Networks

Smart Grid: Big Data



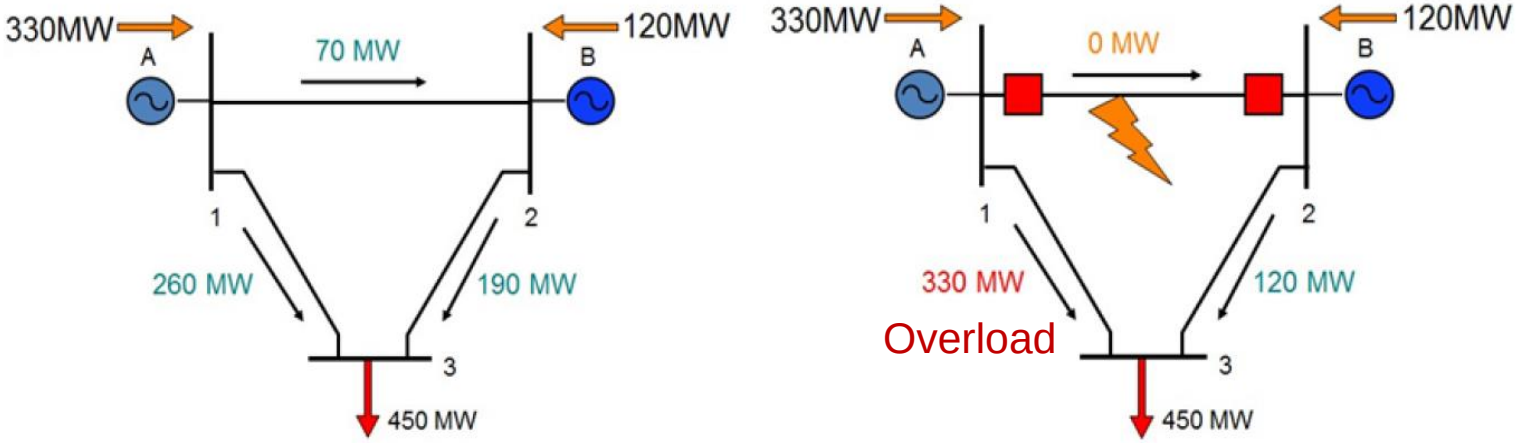
- The anticipated smart grid data deluge:
  - North American power grid: generate 4.15 TB data per day
  - 61.8 million smart meters deployed in the U.S. by the end of 2013
  - Every one million users produce 27.3TB per year

Blackout

- Increasing integration between cyber operations and physical infrastructures for generation, transmission, and distribution control

- The security and reliability are not guaranteed

Contingency Analysis



- Assume line 1-2 is disconnected.
- Generators A and B cannot change productions quickly.
- The flows over other lines would increase.
- Trigger cascading failure.

Security Constrained Optimal Power Flow (SCOPF)

- SCOPF: Minimizing the cost of system operation while satisfying a set of postulated contingency constraints.

$$
\begin{aligned}
\min_{\{\mathbf{x}^c\}_0^C;\{\mathbf{u}^c\}_0^C} \quad & f^0(\mathbf{u}^0) \quad \text{scheduling objective} \\
\text{s.t.} \quad & \mathbf{g}^0(\mathbf{x}^0,\mathbf{u}^0)=0, \text{power flow equations} \\
& \mathbf{h}^0(\mathbf{x}^0,\mathbf{u}^0)\le 0, \text{operating limits for base case} \\
& \mathbf{g}^c(\mathbf{x}^c,\mathbf{u}^c)=0, \text{power flow equations} \\
& \mathbf{h}^c(\mathbf{x}^c,\mathbf{u}^c)\le 0, \text{operating limits for contingency k} \\
& \|\mathbf{u}^0-\mathbf{u}^c\|^2 \le \mathbf{\Delta}_c, c=1,\ldots,C, \text{security constrains}
\end{aligned}
$$

- Challenges:
  - Number of constraints is prohibitive.
  - How to find the best operating point with a scalable algorithm?

L. Liu, A. Khodaei, W. Yin and Z. Han, "A distribute parallel approach for big data scale optimal power flow with security constraints," *2013 IEEE International Conference on Smart Grid Communications (Smart Grid Comm)*, Vancouver, BC, 2013, pp. 774-778.

L. Liu, M. Esmalifalak, Q. Ding, V. A. Emesih and Z. Han, "Detecting False Data Injection Attacks on Power Grid by Sparse Optimization," in *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 612-621, March 2014.

DC SCOPF

- ‘N-1' contingency, corrective setting:

$$
\begin{aligned}
\min_{\{\theta^c\}_{c=0}^{C};\{\mathbf{P}^{g,c}\}_{c=0}^{C}} \quad & \sum_{i\in\mathcal{G}} f_i^g(\mathbf{P}_i^{g,0}) \\
\text{subject to} \quad & \mathbf{B}_{bus}^0\boldsymbol{\theta}^0 + \mathbf{P}^{d,0} - \mathbf{A}^{g,0}\mathbf{P}^{g,0} = 0, \\
& \mathbf{B}_{bus}^c\boldsymbol{\theta}^c + \mathbf{P}^{d,c} - \mathbf{A}^{g,c}\mathbf{P}^{g,c} = 0, \\
& |\mathbf{B}_f^0\boldsymbol{\theta}^0| - \mathbf{F}_{max} \leq 0, \\
& |\mathbf{B}_f^c\boldsymbol{\theta}^c| - \mathbf{F}_{max} \leq 0, \\
& \underline{\mathbf{P}^{g,0}} \leq \mathbf{P}^{g,0} \leq \overline{\mathbf{P}^{g,0}}, \\
& \underline{\mathbf{P}^{g,c}} \leq \mathbf{P}^{g,c} \leq \overline{\mathbf{P}^{g,c}}, \\
& |\mathbf{P}^{g,0} - \mathbf{P}^{g,c}| \leq \boldsymbol{\Delta}_c, \\
& i \in \mathcal{G}, \quad c = 1,\ldots,C,
\end{aligned}
$$

where $\mathbf{B}_{bus}$ and $\mathbf{B}_f$ can be modified from the bus admittance matrix $\mathbf{Y}_{bus}$. $\mathbf{A}^{g,c}$ is the generator connection matrix.

A Distributed Approach by ADMM

- Introduce a slack variable $\boldsymbol{p}^c$ to rewrite $|P^{g,0} - P^{g,c}| \leq \Delta_c$ as

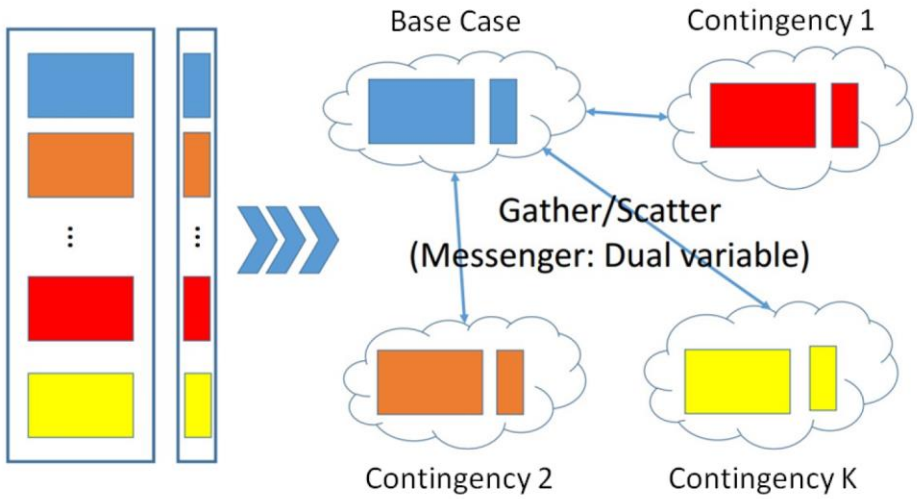
$$\mathbf{P}^{g,0} - \mathbf{P}^{g,c} + \mathbf{p}^c = \boldsymbol{\Delta}_c \qquad (3)$$
$$0 \leq \mathbf{p}^c \leq 2\boldsymbol{\Delta}_c, \quad c = 1, \ldots, C.$$

- The partial scaled augmented Lagrangian associated with (3) can be calculated with follows

$$\mathcal{L}_\rho(\{\mathbf{P}^{g,c}\}_{c=0}^{C}; \{\mathbf{p}^c\}_{c=1}^{C}; \{\boldsymbol{\mu}^c\}_{c=1}^{C})$$
$$= \sum_{i \in \mathcal{G}} f_i^g(\mathbf{P}_i^{g,0}) + \sum_{c=1}^{C} \frac{\rho^c}{2} \|\mathbf{P}^{g,0} - \mathbf{P}^{g,c} + \mathbf{p}^c - \boldsymbol{\Delta}_c + \boldsymbol{\mu}^c\|_2^2.$$

- Iterate till convergence
  - 1. Update $\{P^{g,0}\}$
  - 2. Update $\{P^{g,0}, p^c\}$
  - 3. Update dual variable μ^c

Distributed Implementation



- On multi-core machine
- High performance computer cluster using MPI (message passing interface)
- On cloud using Hadoop or Apache Spark

Numerical Results

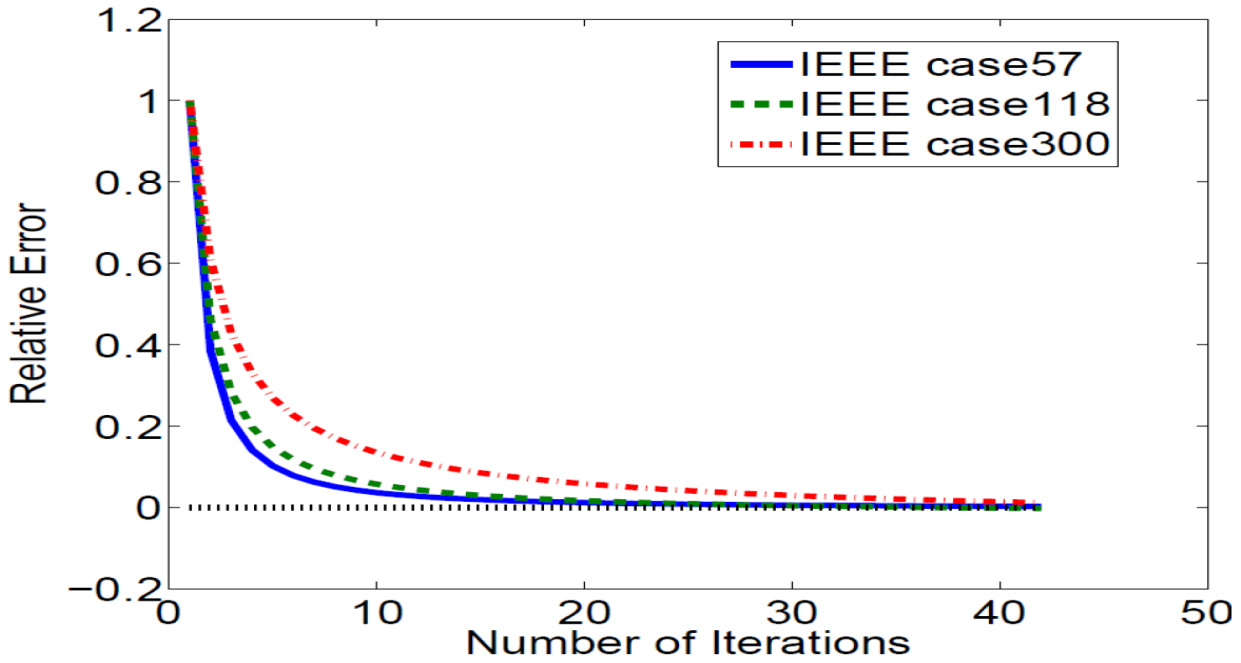


Figure: Convergence performance.

Evaluation setup: Modified data of IEEE 57 bus, IEEE 118 bus and IEEE 300 bus generated by MATPOWER

Multi-Block ADMM: Applications

- Security Constrained Optimal Power Flow
- Data Offloading in Software Defined Networks

Mobile Data Offloading

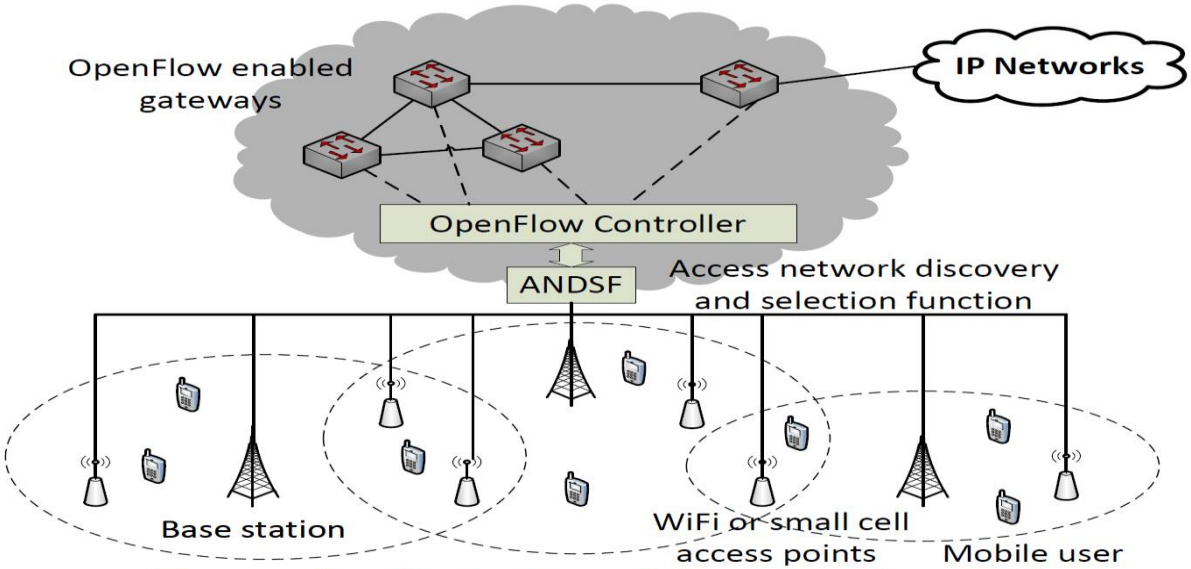


Figure: An illustration of the network model

- Mobile data offloading: Offload traffic from cellular networks to alternate wireless technologies.
- Software defined network (SDN) at the edge: Dynamically route the traffic in a mobile network.

Y. Li, G. Shi, W. Yin, L. Liu and Z. Han, "A Distributed ADMM Approach with Decomposition-Coordination for Mobile Data Offloading," in *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1-1.

Problem Formulation

- Utility of base stations: $\sum_{b=1}^{B} U_b(\mathbf{x}_b)$
- Cost of access points: $\sum_{a=1}^{A} L_a(\mathbf{y}_a)$
- Total revenue: $\sum_{b=1}^{B} U_b(\mathbf{x}_b) - \sum_{a=1}^{A} L_a(\mathbf{y}_a)$

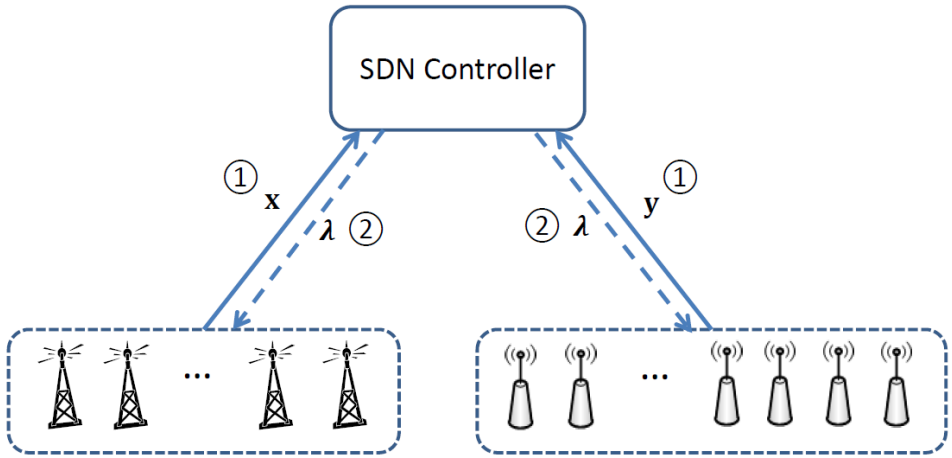
- Equivalent revenue maximization problem:

$$\min_{\{\mathbf{x}_1,\ldots,\mathbf{x}_B\},\{\mathbf{y}_1,\ldots,\mathbf{y}_A\}} \quad \sum_{a=1}^{A} L_a(\mathbf{y}_a) - \sum_{b=1}^{B} U_b(\mathbf{x}_b), \quad \text{Service revenue}$$

$$\text{s.t} \quad \sum_{b=1}^{B} y_{ab} \leq C_a, \quad \forall a, \quad \text{Capacity constraint}$$

$$x_{ba} = y_{ab}. \quad \forall a, b \quad \text{Consensus}$$

Proximal Jacobian ADMM



① Gather: BSs and APs concurrently update $\mathbf{x}$ and $\mathbf{y}$, which are gathered by controller.

② Scatter: Controller simply updates $\boldsymbol{\lambda}$, which are scattered to BSs and APs

- Iterative gather-scatter scheme (Map-Reduce)
- Signaling: $p_{ab}^k = (y_{ab}^k - \frac{\lambda_{ab}^k}{\rho}),\ q_{ba}^k = (x_{ba}^k + \frac{\lambda_{ab}^k}{\rho})$

Numerical Results

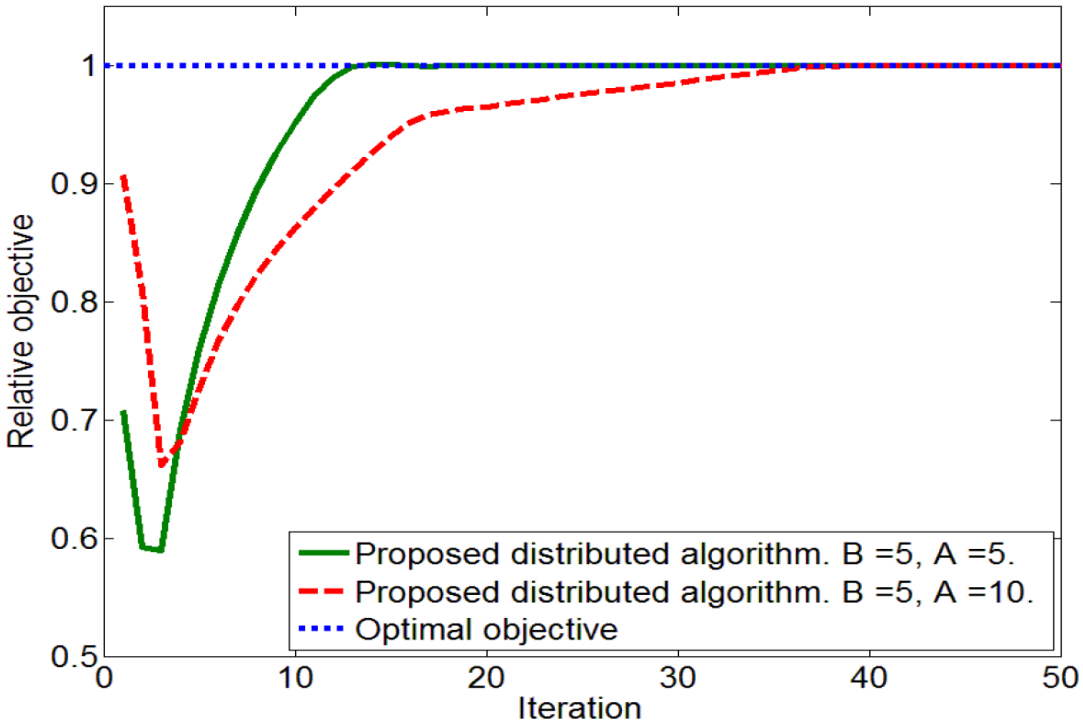


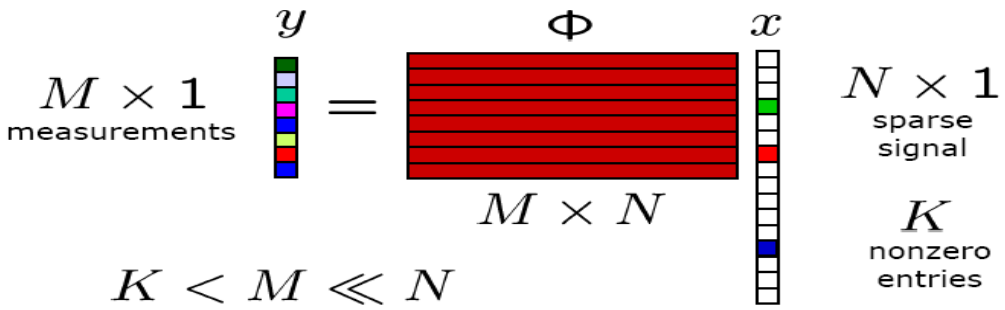
Figure: Convergence performance

Evaluation setup: B = 5 base stations and A = {5,10} access points. Ca = 10Mbps

Large Scale Optimization

- Multi-block Optimization
- Compressive Sensing
- Sublinear Algorithms

Sparse Signal Recovering



- Sparse x
- Traditional Method: Detect all x
- Random linear projection
- Dimension reduction from x to y
  - M > Klog(N/K)
- Recovery algorithm for ill-posed problem

D. L. Donoho, “Compressed sensing,” IEEE Trans. Inf. Theory, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

LASSO

- LASSO: Least absolute shrinkage and selection operator (R. Tibshirani, 1996)

- Sparsity: L_0-norm Problem

$$\arg\min_{\boldsymbol{x}\in\mathbb{R}^N} ||\boldsymbol{\Phi x}-\boldsymbol{y}||_2^2 \quad \text{s.t.} \quad ||\boldsymbol{x}||_0 \le k$$

$$\Rightarrow \quad \arg\min_{\boldsymbol{x}\in\mathbb{R}^N} \frac{1}{2}||\boldsymbol{\Phi x}-\boldsymbol{y}||_2^2 + \lambda||\boldsymbol{x}||_0$$

Barrier

- For a vector x

p-norm: $||x||_p = (|x_1|^p + |x_2|^p + ... + |x_N|^p)^{\frac{1}{p}}$

L_0-norm: number of non-zero elements

L_1-norm: sum of absolute values

L_2-norm: sqrt sum of squares

L_∞-norm: max (x)

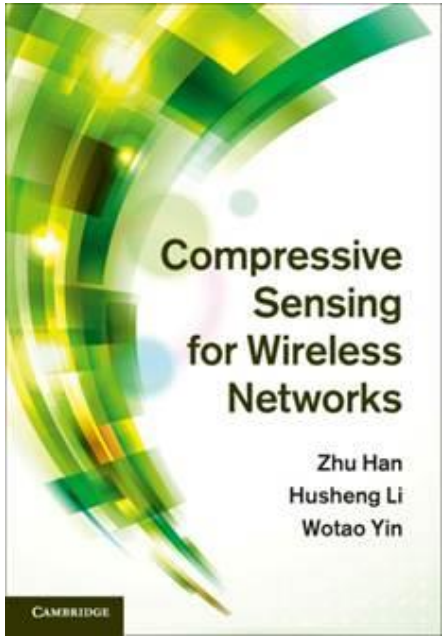
- Sparsity Relaxation: L_1-norm Problem, LASSO

$$\arg\min_{\boldsymbol{x}\in\mathbb{R}^N} \frac{1}{2}||\boldsymbol{\Phi x}-\boldsymbol{y}||_2^2 + \lambda||\boldsymbol{x}||_1$$

S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge Univ., 2004.

Algorithms

- Classic solvers and omitted solvers (ADMM & BSUM)
- Other algorithms
  - Shrinkage Operate
  - Prox-linear Algorithms
  - Dual Algorithms
  - Bregman Method
  - Homotopy Algorithm and Parametric Quadratic Programming
  - Continuation, Varying Stepsizes and Line Search
  - Greedy Algorithms
    - Greedy Pursuit Algorithms
    - Iterative Support Detection
    - Hard Thresholding



http://www.math.ucla.edu/~wotaoyin/summer2013/lectures.html

Application: Compressive Spectrum Sensing

- Cognitive Radio Motivation:
  - Most of the licensed spectrum is not used by the licensed users
- How Cognitive Radio Work
  - Secondary (unlicensed) users detect the spectrum holes (unoccupied spectrum) and utilize the spectrum at the absence of the primary (licensed) users
- Advantage of Cognitive Radio

  Improve radio spectrum utilization
- Key Enabler

  Spectrum sensing (possibly wideband)

sensing wideband

$$\arg\min_{\boldsymbol{x}\in\mathbb{R}^N} \frac{1}{2}||\Phi\boldsymbol{x}-\boldsymbol{y}||_2^2+\lambda||\boldsymbol{x}||_1$$

Y. L. Polo, Y. Wang, A. Pandharipande, and G. Leus, “Compressive wide-band spectrum sensing,” in Proc. IEEE Int. conf. on Acoust., Speech and Signal Process. (ICASSP), Taipei, Apr. 2009, pp. 2337–2340.

X. Zhang, Y. Ma, Y. Gao and S. Cui, "Real-time Adaptively-Regularized Compressive Sensing in Cognitive Radio Networks," in *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1-1.

Large Scale Optimization

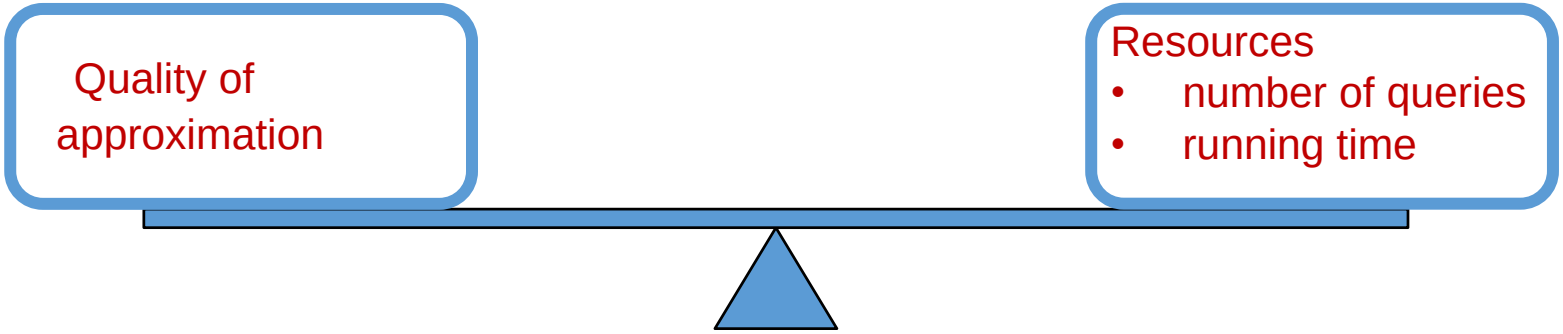
- Multi-block Optimization
- Compressive Sensing
- Sublinear Algorithms

Sublinear Algorithms: Motivation & Basic Concept

Faster than linear algorithms

- N=size of data, we want o(n), not O(n)
- Sublinear Time
  - Queries
  - Samples
- Sublinear Space
  - Data Stream
  - Sketching

- Deterministic Algorithms



Sublinear Algorithms: Example

- Consider a tall skyscraper building and you do not know the total number of this building.
- Now you want to test how high a cat can fly.
- When you throw a cat out of window from floor 1, 2, … n, the cat will survive;
- When you throw a cat out of window from floor n+1, n+2,..., the cat will die.
- The question to develop an efficient method to determine n given that you have 2 cats.

Sublinear Algorithms: Deterministic Algorithm

- Test each floor? 1,2,3,4,5….
  - This will lead a linear algorithm O(n).
- Double floors every time? 1,2,4,8,16…
  - This will lead to O(log n), but you need O(log n) cats.
- The Two-Cat Algorithm $O(\sqrt{n})$

```
1: i = −1, l = 1
2: for the first cat is still alive do
3:     i++;
4:     test floor l = l + i;
   l = l − i;
5: for the second cat is still alive do
6:     l++;
7:     test floor l;
8: Output l;
```

1,3,6,10,15,21,28,…

Future Challenges

- For large-scale communication networks:
  - Tradeoff between error & computing complexity
  - Linear or NP problem (not NP-Complete/NP-Hard) may be already hard to solve…
  - Combination with multi-block optimization

Slides from Dr. Ronnit Rubinfeld's website http://people.csail.mit.edu/ronitt/sublinear.html
Slides from Dr. Dana Ron's website http://www.eng.tau.ac.il/~danar/talks.html
D. Wong, Y. Long, F. Ergun, "A layered architecture for delay sensitive sensor networks" http://www.cse.psu.edu/~sxr48/
Dan Wang and Zhu Han, "Sublinear Algorithms for Big Data Applications," Springer, 2015.

Summary

- Multi-block Optimization
  - Convergence, Distributed, & Parallel
  - Smart Grid & Mobile Data Offloading
- Compressive Sensing
  - Sparsity & LASSO
  - Cognitive Spectrum Sensing
- Sublinear Algorithms
  - Sublinear Convergence

Content

- Overview of Big Data
- Learning Methods
- Commercial Systems
- Large Scale Optimization
- Game Theory based Approaches

Game Theory for Big Data Processing

- Introduction of Game Theory
- Two Specific Games & Applications

Introduction

- **John von Neuman** (1903-1957) co-authored, Theory of Games and Economic Behavior, with Oskar Morgenstern in 1940s, establishing game theory as a field.
- **John Nash** (1928-2015) developed a key concept of game theory (Nash equilibrium) which initiated many subsequent results and studies.
- Since 1970s, game-theoretic methods have come to dominate microeconomic theory and other fields.
- **Nobel Prizes**
  - Nobel prize in Economic Sciences 1994 awarded to **Nash**, **Harsanyi** (Bayesian games) and **Selten** (subgame perfect equilibrium).
  - 2005, **Auman** and **Schelling** got the Nobel prize for having enhanced our understanding of cooperation and conflict through game theory.
  - 2007, **Leonid Hurwicz, Eric Maskin and Roger Myerson** won Nobel Prize for having laid the foundations of mechanism design theory.
  - 2012, **Alvin Elliot Roth**, and **Lloyd Shapley** won Nobel Prize for the theory of stable allocations and the practice of market design.
  - 2014, **Jean Tirole**, won Nobel Prize for the analysis of market power and regulation.

Introduction

- Game theory - mathematical models and techniques developed in economics to analyze interactive decision processes, predict the outcomes of interactions, identify optimal strategies
- Fundamental component of game theory is the notion of a *game*.

  A game is described by a set of rational *players*, the *strategies* associated with the players, and the *payoffs/utilities* for the players. A rational player has his own interest, and therefore, will act by choosing an available strategy to achieve his interest (maximize/minimize utilities).

  A player is assumed to be able to evaluate exactly or probabilistically the outcome or payoff (usually measured by the utility) of the game which depends not only on his action but also on other players' actions.

Example: Nash Equilibrium & Prisoner's Dilemma

- Two suspects in a major crime held for interrogation in separate cells
  - If they both stay quiet, each will be convicted with a minor offence and will spend **1 year** in prison
  - If one and only one of them finks, he will be freed and used as a witness against the other who will spend **4 years** in prison
  - If both of them fink, each will spend **3 years** in prison
- Components of the Prisoner's dilemma
  - **Rational Players:** the prisoners
  - **Strategies:** Stay quiet (Q) or Fink (F)
  - **Solution:** What is the Nash equilibrium/Pareto optimum of the game?

- Pareto Optimum :

Cannot Improve oneself & not damage others.

- Nash Equilibrium: Best response.

Pareto Optimum

	P2 Quiet	P2 Fink
P1 Quiet	1,1	4,0
P1 Fink	0,4	3,3

Nash Equilibrium

Rich Game Theoretical Approaches

- **Non-cooperative Static Game**: play once
- **Repeated Game**: play multiple times
- **Dynamic Game**: optimization utility over time
  - Evolutional game
  - Stochastic game
- **Cooperative Game**
  - Nash Bargaining Game
  - Coalitional Game
- **Other Economic Approaches**: relevant to the game
  - Contract Theory
  - Auction Theory
  - Mean Field Theory
- …

D. Fudenberg and J. Tirole, Game Theory, Cambridge, MA, USA: MIT Press

T. Başar and G. J. Olsder. *Dynamic Noncooperative Game Theory.* Academic Press, London/New York, 1982; second printing 1989.

Z. Han, D. Niyato, W. Saad, T. Basar, and A. Hjorungnes, *Game Theory in Wireless and Communication Networks: Theory, Models and Applications*, Cambridge, U.K.: Cambridge Univ., 2011.

V. Krishna, *Auction Theory*, Academic Press, 2003.

P. Bolton and M. Dewatripont, *Contract Theory*. Cambridge, MA, USA: MIT Press, 2005.

Comparison with Optimization Methods

- **Broad the scope to model a problem**

  $\min f(x)$

  - **Non-cooperative Static Game**:
    Nash equilibrium, Pareto optimum, Stackelberg equilibrium….
  - **Dynamic Game**:
    Bayesian Nash equilibrium, subgame equilibrium…
  - **Cooperative Game:**
    Nash Bargaining solution, core, kernel, nucleus…
  - Some can / cannot be written as an simple optimization problems

- **Explain or Deal with the interactions between different entities**
  - Different entities: service providers, base stations, users, …
  - Explain: Not necessary to solve the problem in a distributed/ parallel manner
  - Deal with: Usually need to solve the problem in a distributed/ parallel manner

Two Specific Games

- Explain & Deal with the interactions between different entities
- For big data/ large-scale network: fast & scalable

- Matching Theory
- Hierarchical Game

Zhu Han, Yunan Gu, and Walid Saad, “Matching Theory for Wireless Networks,” Springer Science + Business Media, LLC, 2017.

Siavash Bayat, Yonghui Li, Zhu Han, and Lingyang Song, “Matching Theory: Applications in wireless communications,” IEEE Signal Processing Magazine, vol. 33, no. 6, p.p. 1053-5888, November 2016.

Yunan Gu, Walid Saad, Merouane Debbah, Mehdi Bennis, and Zhu Han, “Matching Theory for Emerging Wireless Networks: Fundamentals and Applications,” IEEE Communication Magazine, special issue on Emerging Applications, Services and Engineering for Cognitive Cellular Systems, vol. 53, no. 5, p.p. 52-59, May 2015.

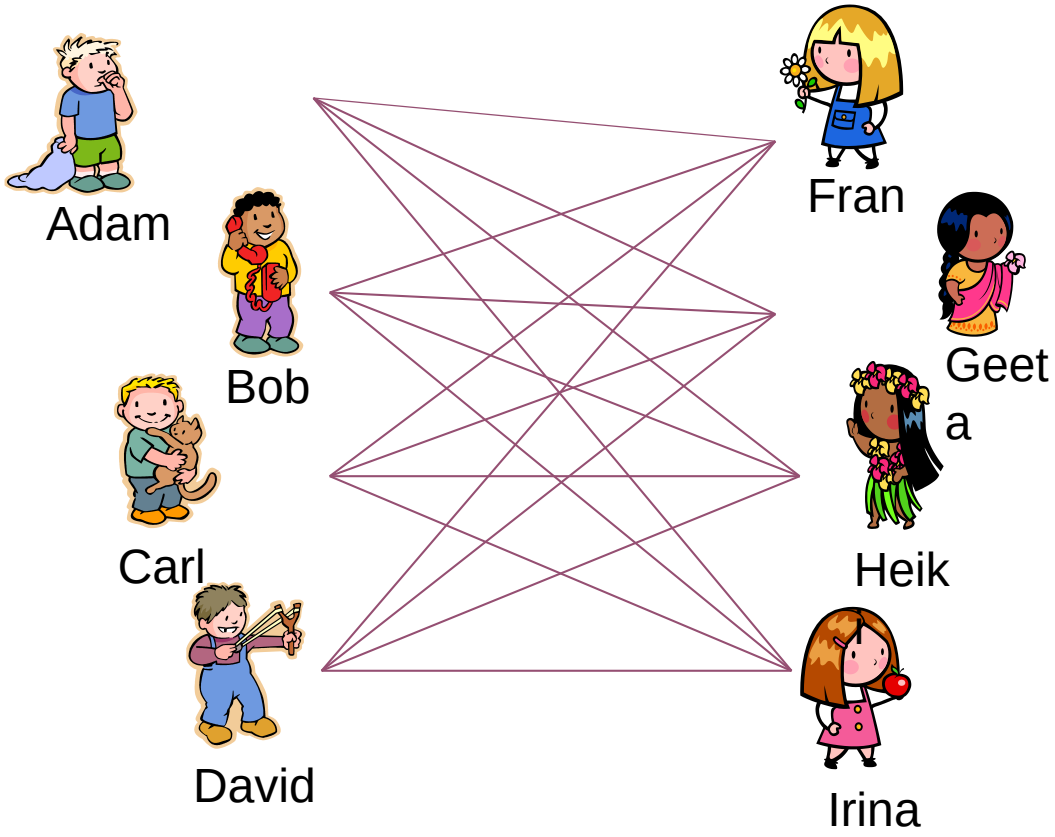
Matching Theory: Introduction

- The Nobel Prize in Economic Sciences 2012

Lloyd S. Shapley
Developed the theory in the 1960s

Alvin E. Roth
Generated further analytical development
practical design of market institutions

Stable Marriage Problem

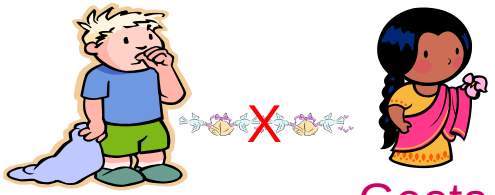


- Women and men be matched
- Respecting their individual preferences

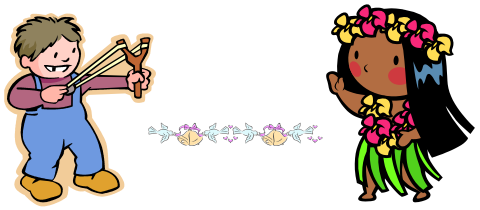
- Example of preferences: Adam: Geeta, Heiki, Irina, Fran

D. Gale and L. S. Shapley ``College Admissions and the Stability of Marriage". *American Mathematical Monthly*. No. **69**, Page, 9–14, 1962.

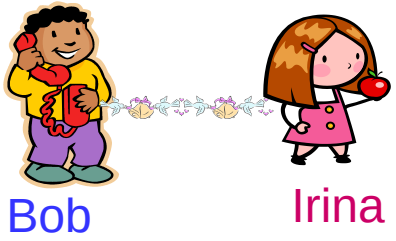
Blocking Pair & Stable Matching

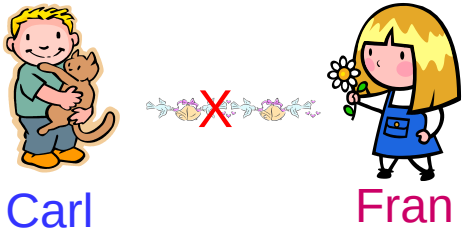


Geeta prefers Carl to Adam!



Blocking Pair





Carl likes Geeta better than Fran!

Stable Matching: No blocking pair.

GS(Gale-Shapley) Algorithm

- The Gale-Shapley algorithm can be set up in two alternative ways:
  - men propose to women
  - women propose to men
- Each man proposing to the woman he likes the best
  - Each woman looks at the different proposals she has received (if any)
  - retains what she regards as the most attractive proposal (but defers from accepting it) and rejects the others
- The men who were rejected in the first round
  - Propose to their second-best choices
  - The women again keep their best offer and reject the rest
- Continues until no men want to make any further proposals
- Each of the women then accepts the proposal she holds
- The process comes to an end

GS Algorithm

Geeta, Heiki, Irina, Fran

Adam

Irina, Fran, Heiki, Geeta

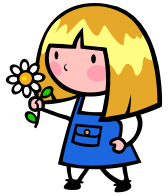
Bob

Geeta, Fran, Heiki, Irina

Carl

Irina, Heiki, Geeta, Fran

David



Adam, Bob, Carl, David

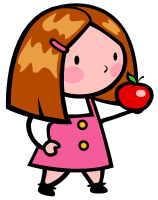
Fran

Carl, David, Bob, Adam

Geeta

Carl, Bob, David, Adam

Heiki



Adam, Carl, David, Bob

Irina

GS Algorithm

Geeta, Heiki, Irina, Fran

Adam

Irina, Fran, Heiki, Geeta

Bob

This is a stable matching

Geeta, Fran, Heiki, Irina

Carl

Irina, Heiki, Geeta, Fran

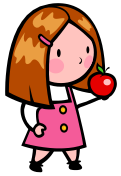
David

Fran

Carl > Adam

Geet

Heiki



David > Bob

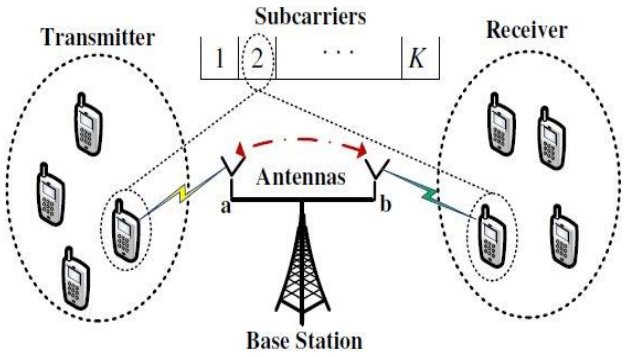
Irina

Properties of GS Algorithm

- Complete matching: everyone gets married
- Stable matching: no blocking pair
- Runtime complexity: $O(n^2)$

Application: Full-Duplex OFDMA Networking

- One BS, M transmitters (TXs), M receivers (RXs), and K subcarriers
- A transceiver unit
- Self interference between antennas
- Transceiver-subcarrier pairing
  - One TX can only be paired with one RX
  - One subcarrier can only be paired with one TX-RX pair
- The transmit power of the BS is fixed
- Objective: matching the TXs, RXs, and subcarriers to each other and adjust the BS power level in each subcarrier such that the total network's sum-rate is maximized.



Boya Di, Siavash Bayat, Lingyang Song, and Yonghui Li, "Radio Resource Allocation for Full-Duplex OFDMA Networks Using Matching Theory," *IEEE INFOCOM - Student Activities (Posters)*, Toronto, Canada, May, 2014.

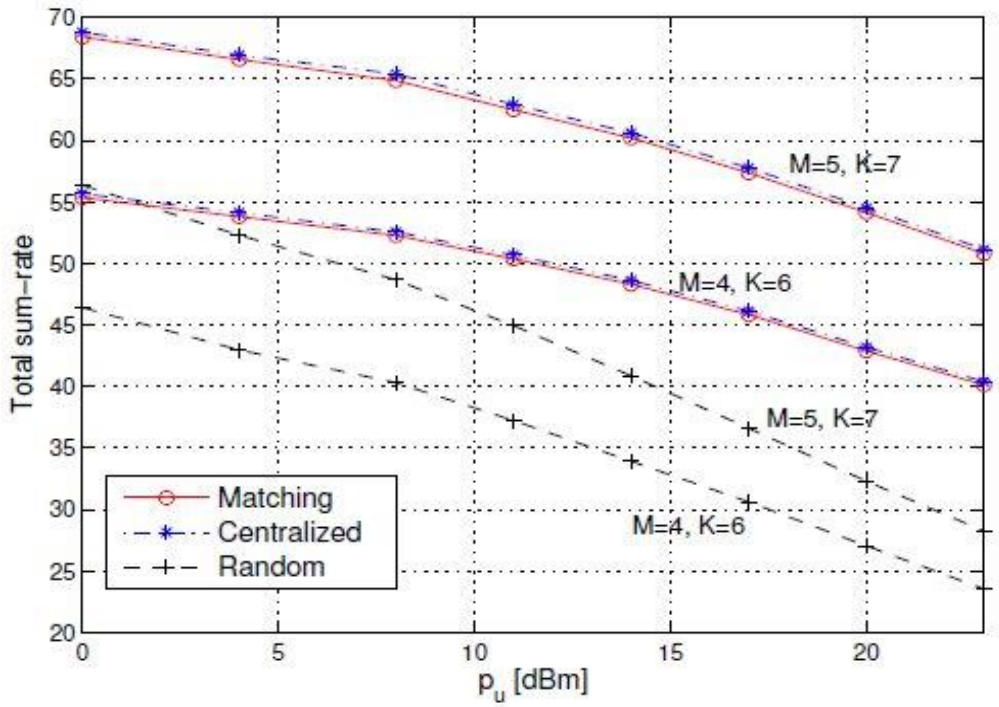
L. Song, Y. Li and Z. Han, "Resource allocation in full-duplex communications for future wireless networks," in *IEEE Wireless Communications*, vol. 22, no. 4, pp. 88-96, Aug. 2015.

Simulation Result

- Random matching algorithm: the TXs, the RXs, and subcarriers are randomly matched with each other

- Complexity level: the iteration number is much smaller than that of the centralized algorithm

- The performance is close to the centralized algorithm



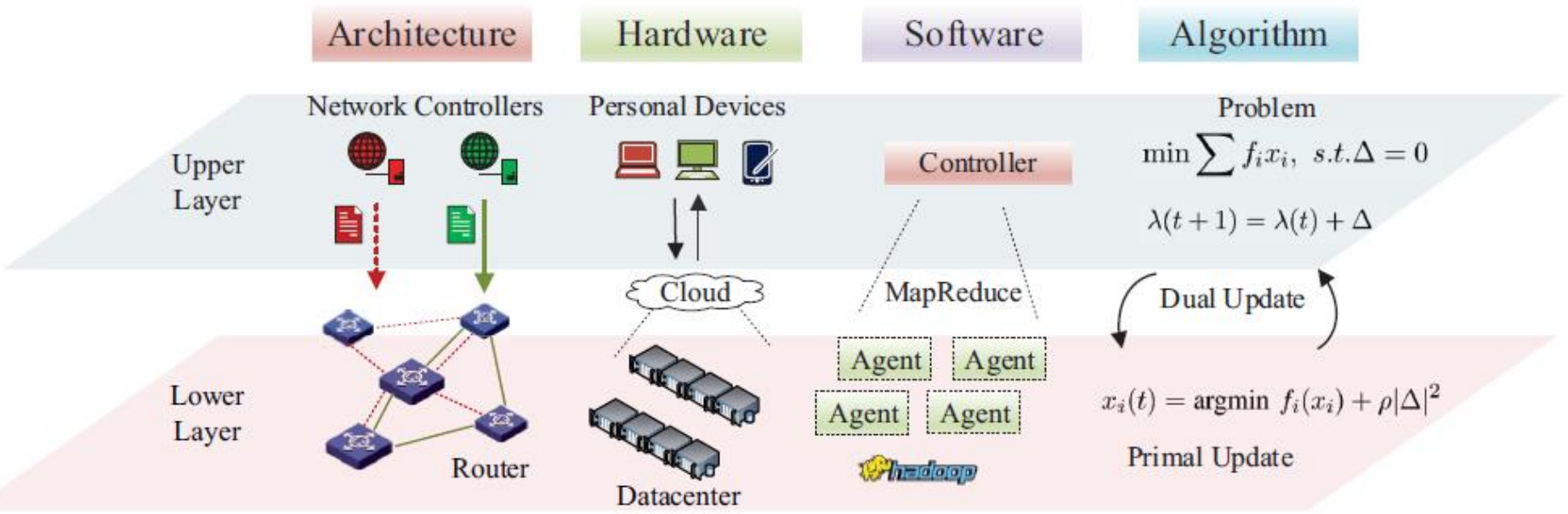
Two Specific Games

- Explain & Deal with the interactions between different entities
- For big data/ large-scale network: fast & scalable

- Matching Theory
- Hierarchical Game

Hierarchies in Large-Scale Networks

- Ubiquitous Hierarches in Large-Scale Networks



Z. Zheng, L. Song, Z. Han, G. Y. Li, H. V. Poor, “Game Theoretic Approaches to Massive Data Processing in Wireless Networks,” to appear in *IEEE Wireless Commun. Mag*.

Hierarchies in Large-Scale Networks

Challenge:

- Agents optimize selfish objectives rather than controller's
- Need incentive mechanisms

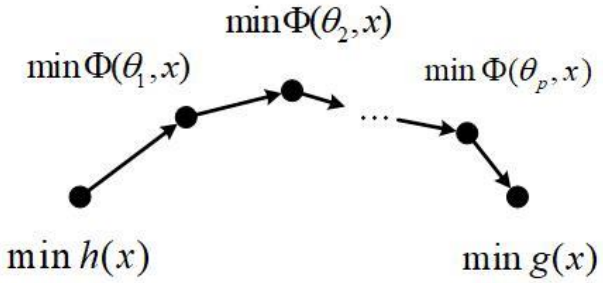
Hierarchical Game with One Leader & One Follower

- Controller's/ Leader 's Objective: $\min g(x)$
- Agent's/ Follower 's Objective: $\min h(x)$
- Incentive: Let follower achieves leader's optimum rather than its own
- Problem Formulation as a Game
  - Leader's game:

$$\arg\min_{\theta} g(x) + \theta x$$

  - Follower's game:

$$\arg\min_{x} h(x) - \theta x$$



x : variable controlled by follower

θ : price from leader to follower

Hierarchical Game with One Leader & One Follower

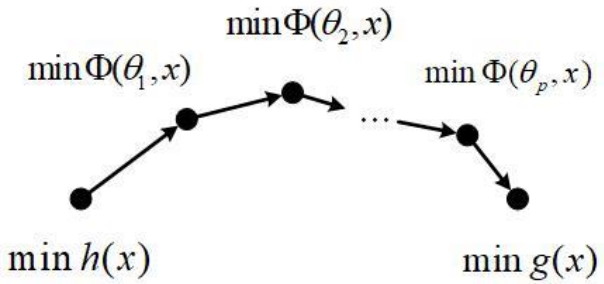
- Leader: give price & ask follower to minimize an incentive function
- Incentive Function Design

$$\min \Phi(\theta, x) = g(x) + h(x) - \theta x$$

Lear's utility Follower's utility Payment

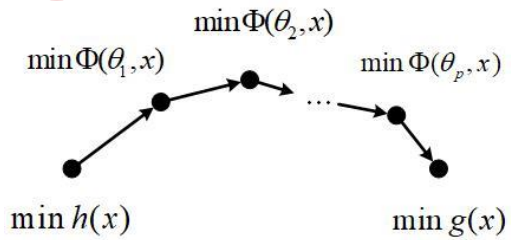
- Hierarchical Game Update
  - Leader's Update $\theta_{p+1} = \dfrac{\mathrm{d}h(x_p)}{\mathrm{d}x}$
  - Follower's Update $x_{p+1} = \arg\min_x \Phi(\theta_{p+1}, x)$

Marginal Cost



Optimum Properties

- Relaxed Stackelberg Equilibrium
  - Optimum of Leader's Original Objective: $\min_x g(x)$
  - Optimum of Incentive Function: $\min_x \Phi(\theta_p, x) = g(x) + h(x) - \theta_p x$
  - Optimum of Follower's Utility: $\min_x h(x) - \theta_p x$
  - Same Optimum: $\begin{cases} \min_x g(x) \\ \min_x h(x) - \theta_p x \end{cases}$
  - Not Optimum of Leader's Utility: $\min_\theta g\left(x(\theta)\right) + \theta \cdot \left(x(\theta)\right)$ Not idealized
- Conditions:
  - Strongly convex leader's utility function
  - Convex follower's utility function
  - Uniform Lipschitz gradient for follower's utility function



Convergence & Scalability Properties

- Just like Gradient Descent/Ascent :First-order numerical optimization
  - Linear Speed: $\varepsilon = o(1/p)$
  - Scalability

$$\arg\min_{\{\theta_i\}} \sum_{i=1}^{N} g(x_i)$$

$$\arg\min_{x_i} h_i(x_i) - \theta_i x_i$$

Hierarchical Game with One Leader & Multiple Followers

- Leader's game:

$$\arg\min_{\{\theta_i\}} \sum_{i=1}^{N} g_i(x_i) + \sum_{i=1}^{N} \theta_i x_i$$

- Followers' game:

$$\arg\min_{x_i} \; h_i(x_i) - \theta_i x_i$$

- Constraints:

$$\sum_{i=1}^{N} A_i x_i = B$$

Z. Zheng, L. Song, and Z. Han, "Bridging the gap between big data and game theory: a general hierarchical pricing framework", accepted by IEEE Int. Conf. Commun. (ICC17), Paris, France, May 2017.
Z. Zheng, L. Song, and Z. Han, "Bridge the gap between ADMM and stackelberg game: incentive mechanism design for big data networks," *IEEE Signal Process. Lett.*, vol. 24, no. 2, pp. 191-195, Feb. 2017.

Huaqing Zhang, Yong Xiao, Shengrong Bu, Dusit Niyato, F. Richard Yu, and Zhu Han, "Computing Resource Allocation in Three-Tier IoT Fog Networks: a Joint Optimization Approach Combining Stackelberg Game and Matching", IEEE Internet Of Things Journal, Special Issue on Fog Computing in the Internet of Things, vol. 4, no. 5, p.p. 1204-1215, October 2017.

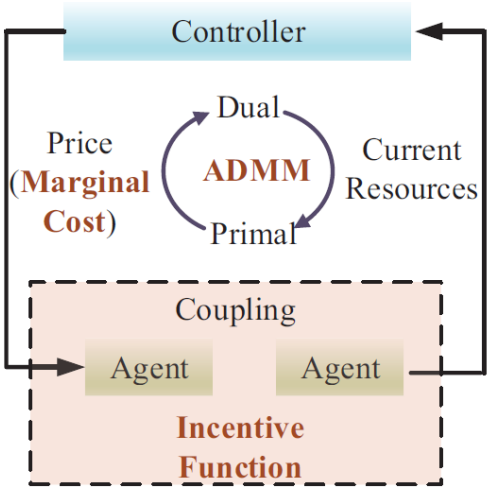
Deal with Constraints with ADMM

- Leader's Update $\theta_i^{p+1} = \dfrac{\mathrm{d}h_i(x_i^p)}{\mathrm{d}x_i}$
- Followers & Leader's Coordination
  - Sequential Follower' Update

$$x_i(t+1) = \arg\min_{x_i} \Phi_i(\theta_i^{p+1}, x_i) + \lambda A_i x_i + \frac{\rho}{2} \left| A_i x_i + \sum_{k=1}^{i-1} A_k x_k(t+1) + \sum_{i+1}^{N} A_k x_k(t) - B \right|^2$$

  - Leader's Dual Update

$$\lambda(t+1) = \lambda(t) - \rho\left(\sum_{i=1}^{N} A_i x_i(t+1) - B\right)$$



Optimal Properties

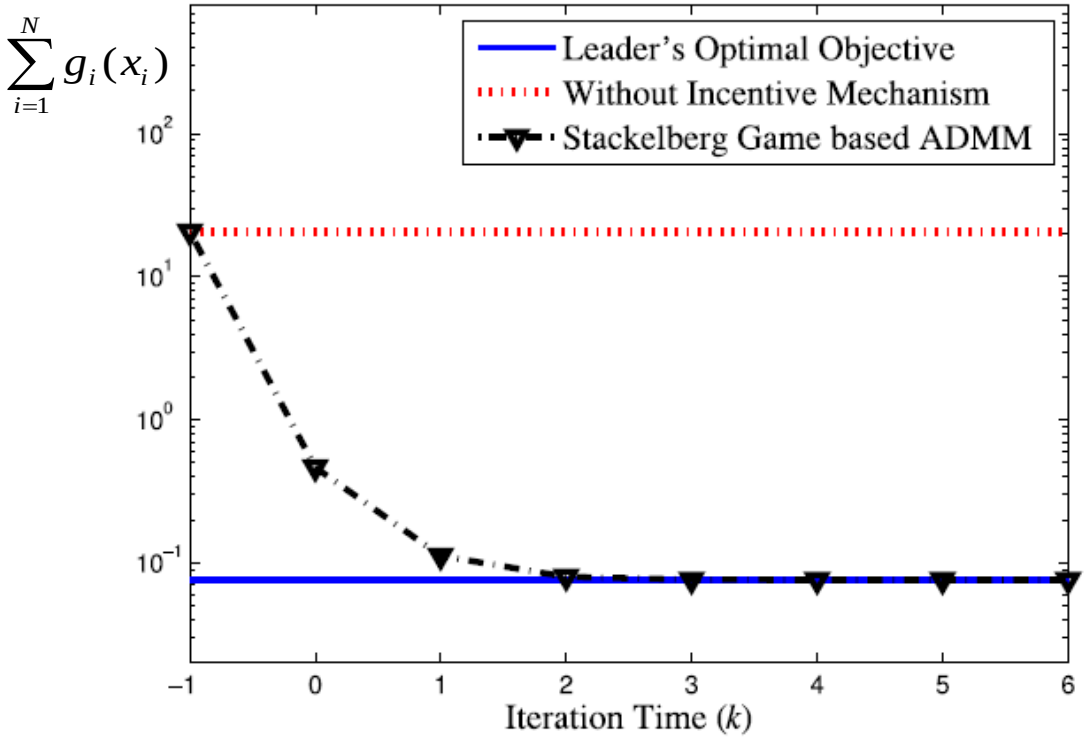
- Relaxed Stackelberg Equilibrium
  - Optimum of Leader's Original Objective $\arg\min_{\{\theta_i\}} \sum_{i=1}^{N} g_i(x_i)$
  - Optimum of Follower's Incentive Function
  - Optimum of Follower's Utility
  - Not Optimum of Leader's Utility $\arg\min_{\{\theta_i\}} \sum_{i=1}^{N} g_i(x_i) + \sum_{i=1}^{N} \theta_i x_i$
- Conditions:
  - Decomposed utility functions for the leader
  - Linear constraints (Convex can be right)
  - Strongly convex leader's utility function
  - Convex follower's utility function
  - Uniform Lipschitz gradient for follower's utility function

Convergence & Scalability

- Linear Speed on Outer Loop : $\varepsilon = o(1/\ p)$
- Scalability on Outer Loop

$$g_i(x_i) = |\ x_i\ |^2$$

$$h_i(x_i) = \exp(x_i)$$



Linear Convergence Speed

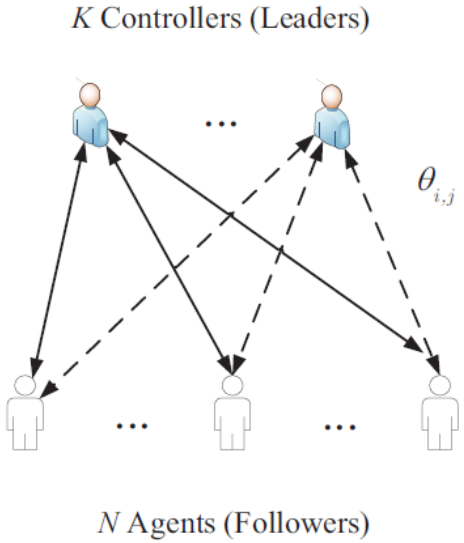
Hierarchical Game with Multiple Leaders & Multiple Followers

- Utility of Each Leader

$$\min_{\mathbf{x}_{i,*}} G_i(\mathbf{x}_{i,*}) = \sum_{j=1}^{N} g_{i,j}(x_{i,j})$$

- Utility of Each Follower

$$\min_{\mathbf{x}_{*,j}} H_j(\mathbf{x}_{*,j}) = \sum_{i=1}^{K} h_{i,j}(x_{i,j})$$



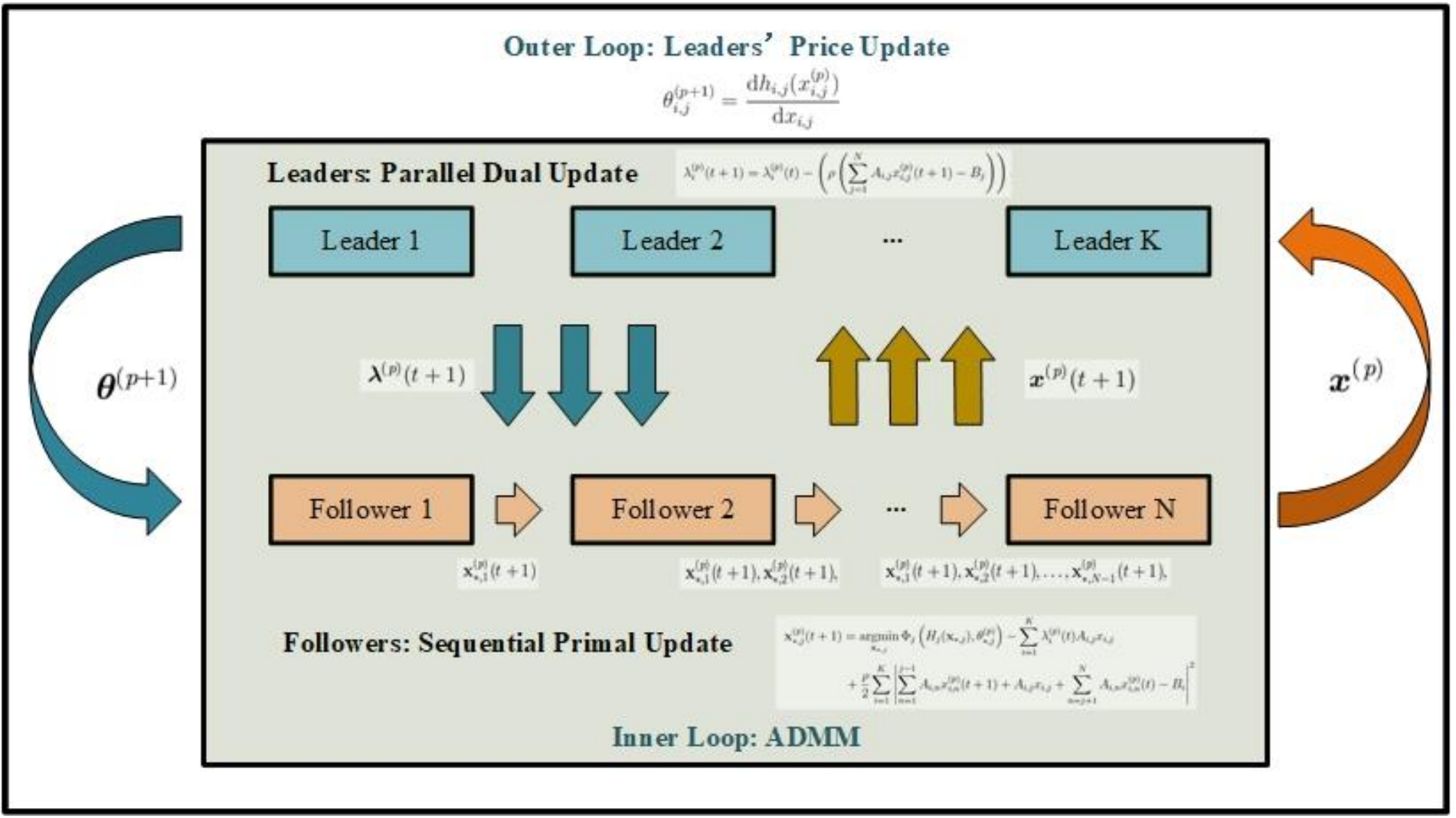
$x_{i,j}$: resources provided from agent j to controller i

Z. Zheng, L. Song, Z. Han, G. Li, and V. Poor, “Multi-leader multi-follower game-based ADMM for big data processing,” IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC17), Sapporo, Japan, Jul. 2017.

Z. Zheng, L. Song, Z. Han, G. Y. Li, H. V. Poor, Game Theory for Big Data Processing: Multi-Leader Multi-Follower Game-based ADMM, submitted to IEEE Trans. Signal Processing.

Huaqing Zhang, Yong Xiao, Lin X. Cai, Dusit Niyato, Lingyang Song, and Zhu Han, “A Multi-Leader Multi-Follower Stackelberg Game for Resource Management in LTE Unlicensed”, IEEE Transactions on Wireless Communications, vol. 16, no. 1, pp. 348-361, January 2017.

MLMF Game-based ADMM

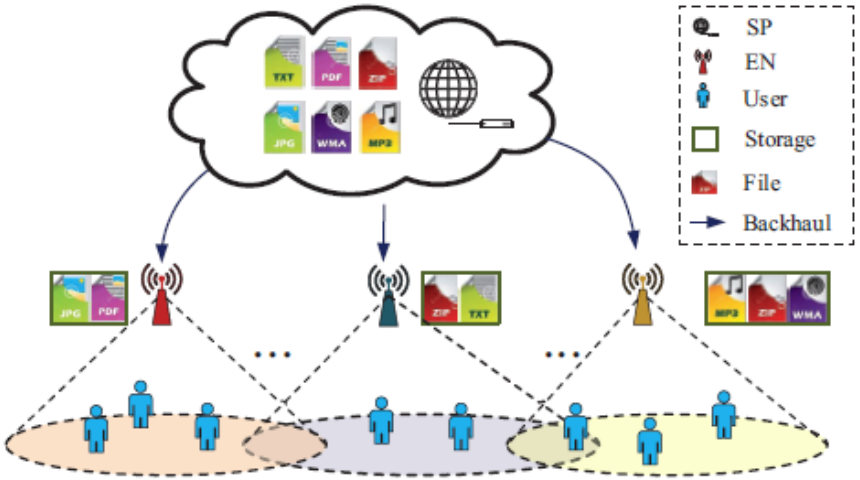


Game Theory for Big Data Processing

- Motivation
- Introduction of Game Theory
- A General Hierarchical Game
- Applications

Hierarchical Game: Application for Edge Caching

- Limited backhaul resources vs excessive streaming
- Anticipated file demand popularity by service provider (SP)
- SP proactively transmits popular files to a large number of edge nodes (ENs)
  - ENs - base stations, small-cell base stations, or WiFi access points



Z. Zheng, L. Song, Z. Han, G. Y. Li, H. V. Poor, A Stackelberg Game Approach to Proactive Caching in Large-Scale Mobile Edge Networks, submitted to IEEE Trans.Wireless Commun.

Huaqing Zhang, Yanru Zhang, Yunan Gu, Dusit Niyato, and Zhu Han, “A Hierarchical Game Framework for Resource Management in Fog Computing,” IEEE Communications Magazine, special issue on Fog Computing and Networking, vol. 55, no. 8, p.p. 52-57, August 2017.

Huaqing Zhang, Yong Xiao, Shengrong Bu, Richard Yu, Dusit Niyato, and Zhu Han, “Distributed Resource Allocation for Data Center Networks: A Hierarchical Game Approach,” accepted IEEE Transactions on Cloud Computing.

Stackelberg Game-based ADMM

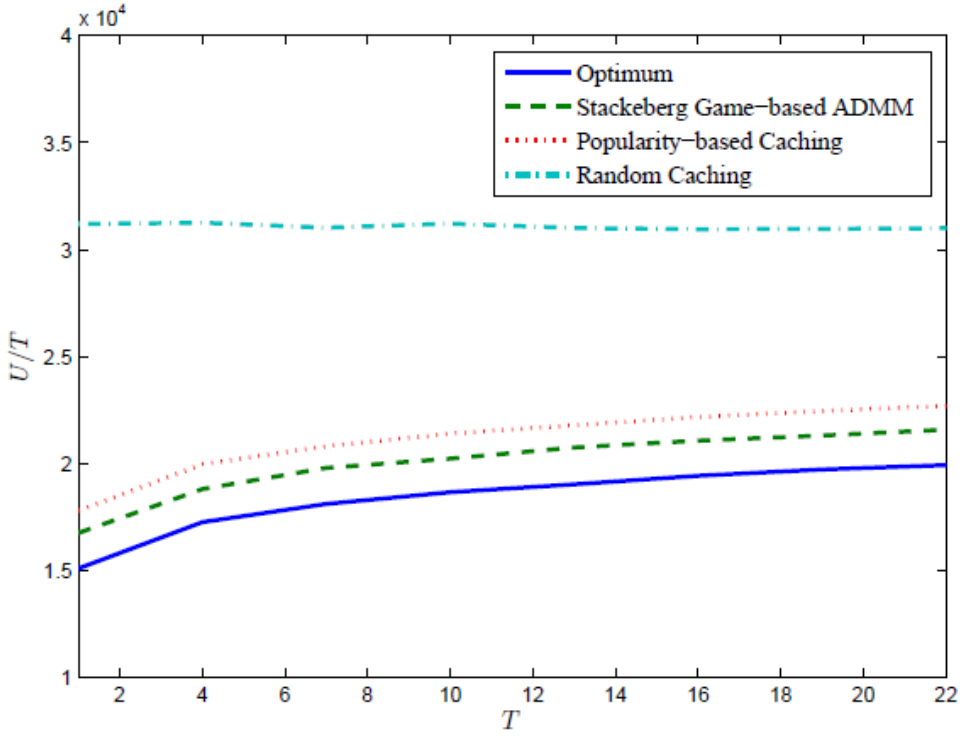
- Objective
  - Minimize total backhaul for SP
- Application of the Stackelberg game-based ADMM
  - One SP (controller) and multiple ENs (agents)

MLMF game-based ADMM reduces to Stackelberg game-based ADMM

  - SP pays for the ENs' backhaul & storage resources

Backhaul Utility

Average backhaul resources vs demands change times



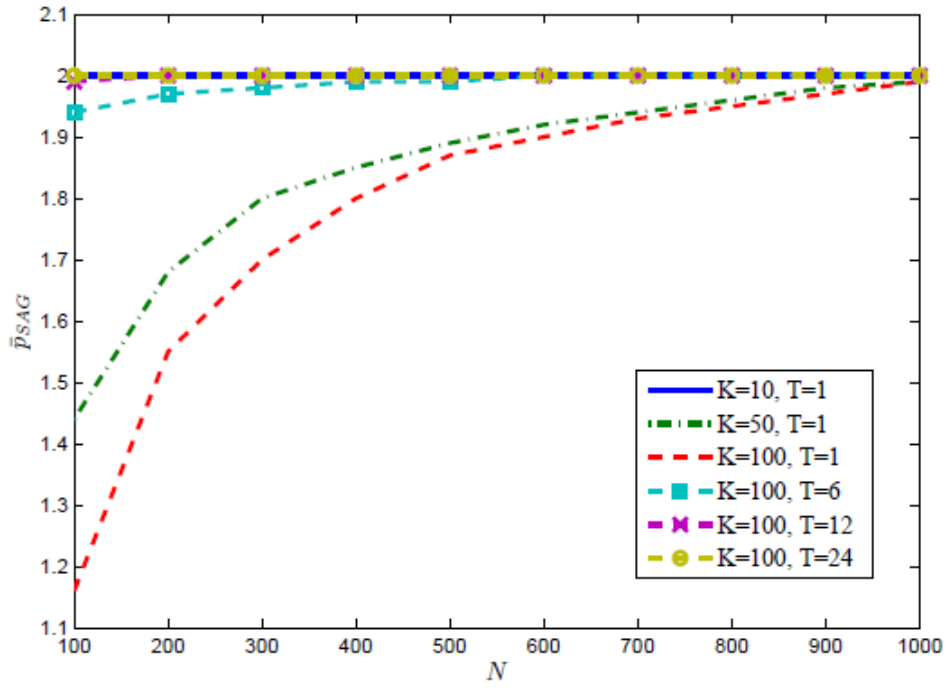
- 20 Files
- 100 ENs
- 1000 Users

- Popularity-based caching: cache the most popular files on ENs
- Random caching: cache files randomly

- Cost less backhaul resources than popularity-based caching & random caching
- Cost more backhaul resources when users change file demands more frequently

Convergence Speed

Average iteration time vs Number of users

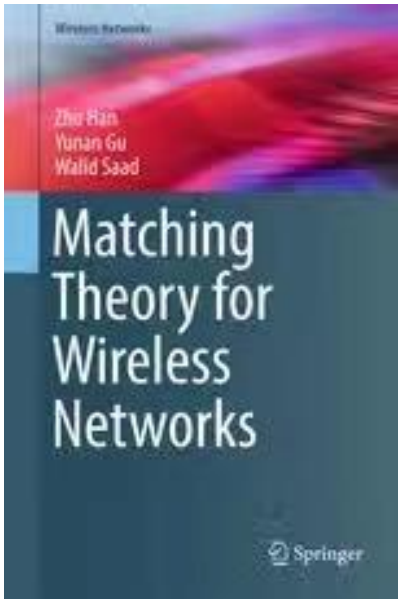


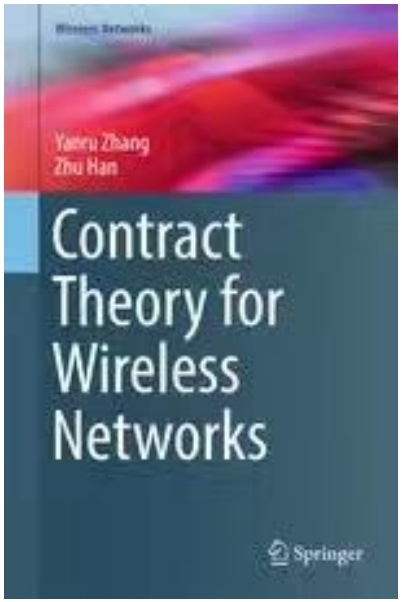
- 20 Files
- 10, 50, 100 ENs
- 1, 6, 12, 24 Demand Change
- Accuracy: 10^{-2}

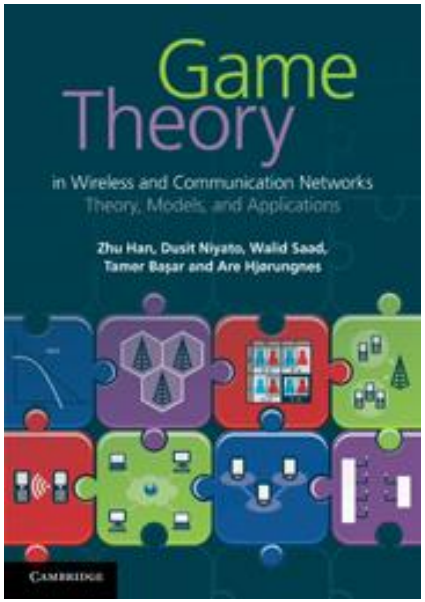
- Iteration times: change sub-linearly with the number of Users & T
- Iteration times: keep below 2.

Summary of Game Theory

- Provide more possible approaches in modeling and solving
- More specific algorithms are needed in the future





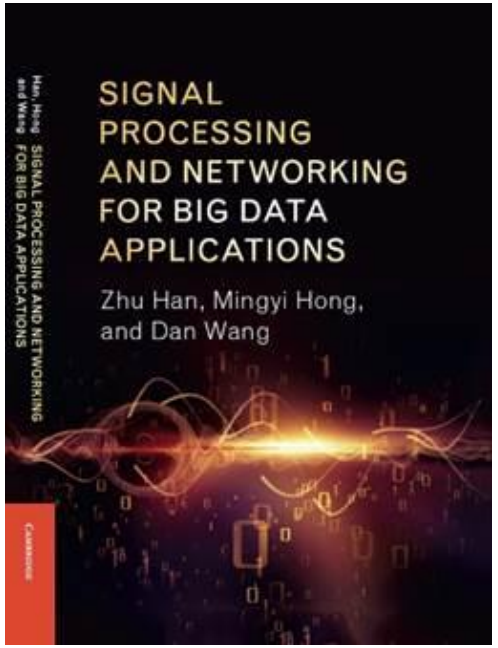


Summary of Tutorial

- Signal Processing Methods for Future Communication Works
  - Learning Methods
  - Commercial Systems
  - Large Scale Optimization
  - Game Theory based Approaches

www2.egr.uh.edu/~zhan2/big_data_course/

wireless.egr.uh.edu/research.html



Thanks