Convex Optimization

Spring 2019

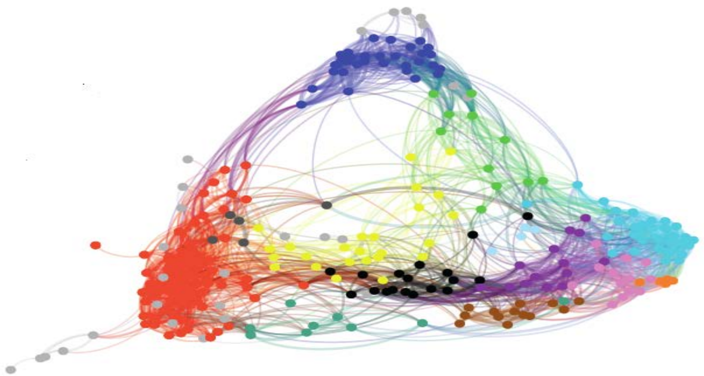
Yuanming Shi

信息科学与技术学院
School of Information Science and Technology

Outline

- **Data science models**
  - Linear, bilinear, quadratic, low-rank, and deep models
- **Large-scale optimization**
  - Constrained vs. unconstrained, convex vs. nonconvex, deterministic vs. stochastic, solvability vs. scalability
- **High-dimensional statistics**
  - Convex geometry, local geometry, global geometry
- **Topics and grading**
  - Theoretical foundations, first-order methods, second-order methods, stochastic methods, and applications.

Motivations: The Era of Big Data



Intelligent IoT applications

Autonomous vehicles

Smart home

Smart city

Smart health

Smart agriculture

Smart drones

Financial big data

- **Financial data & AI technologies:** set up analytic models to gain valuable insights for better business decisions

Large Volume

data generating at speed of 1TB/day in NYSE (2013)

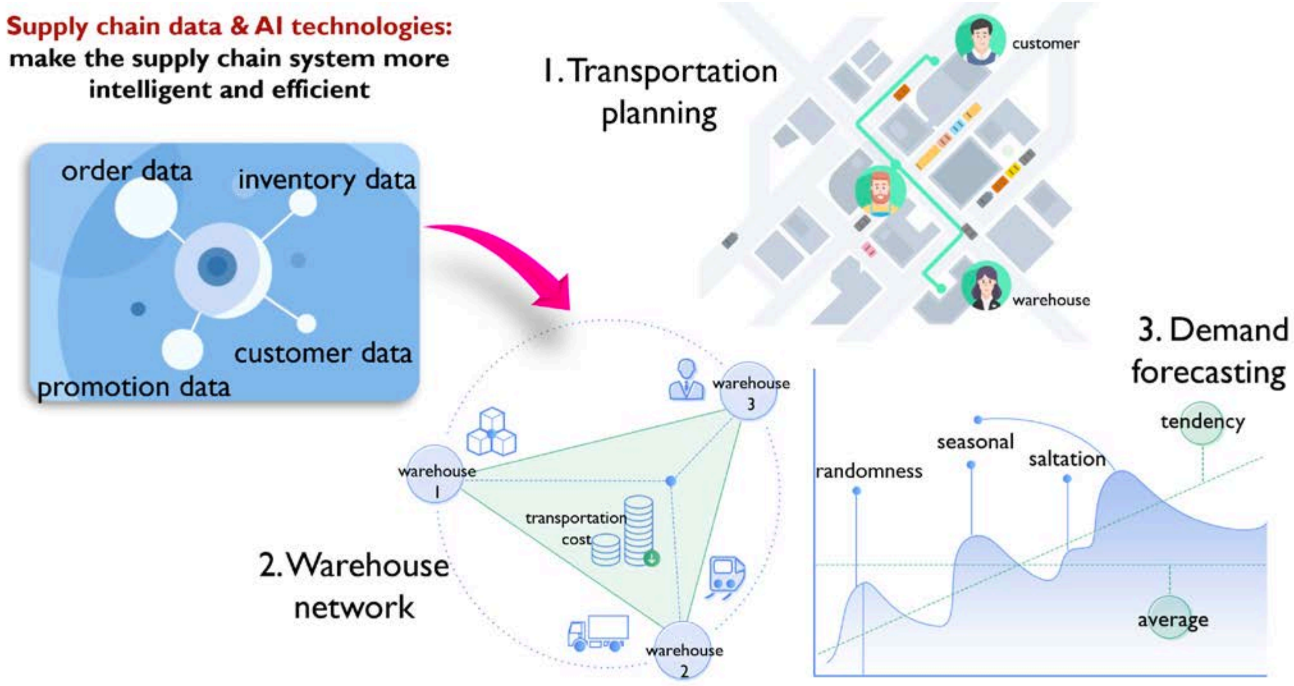
typically 100,000trans/s in High-frequency Trading

High Velocity

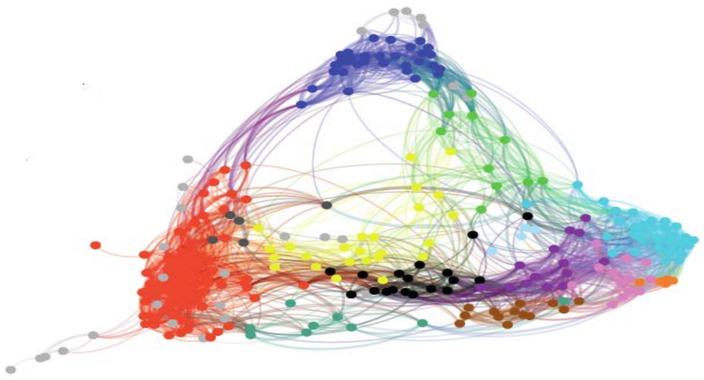
Wide Variety

various data sources and types

Intelligent supply chain system

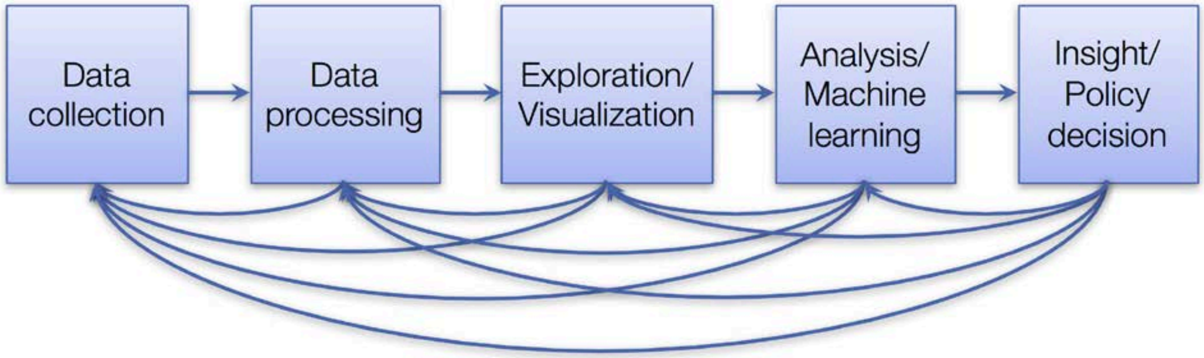


Vignettes A: ***Data Science Models***

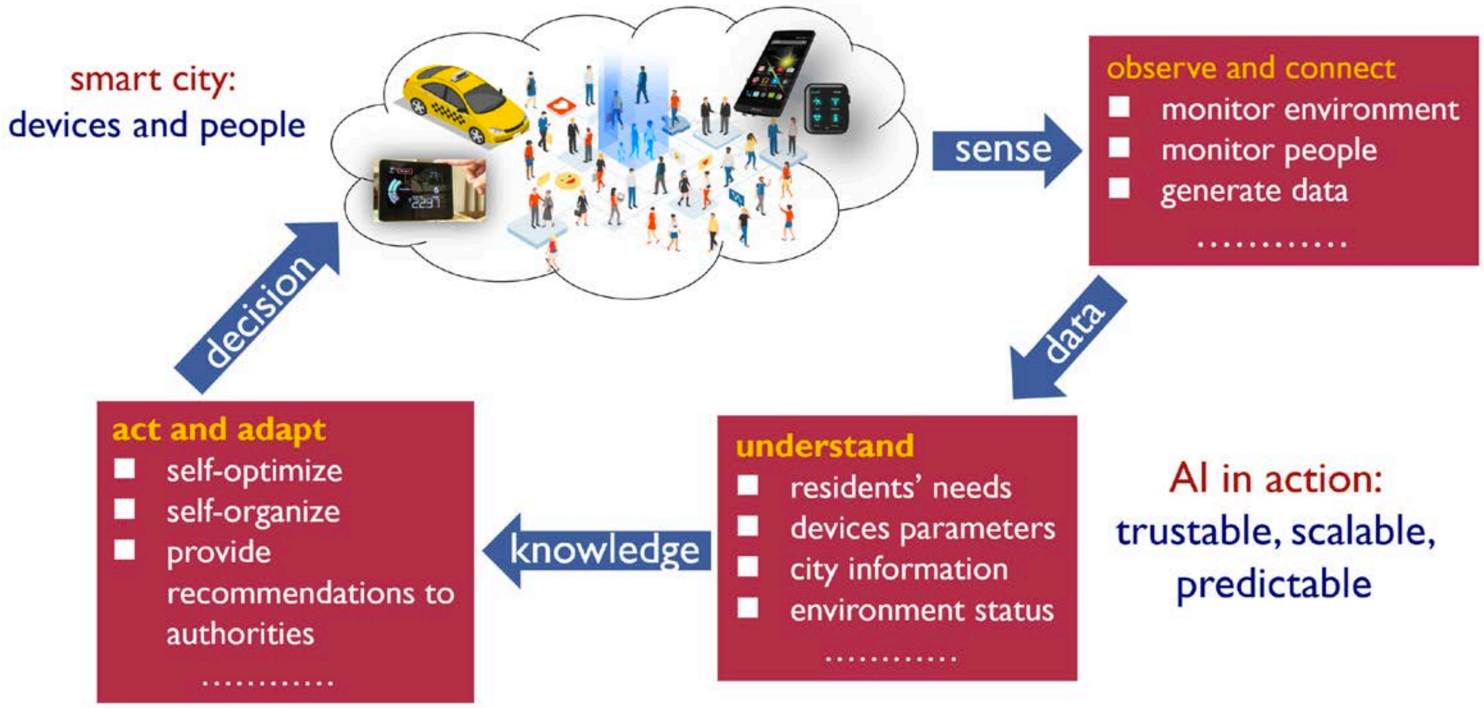


What is data science?

- **Some possible definitions**
  - Data science is the application of ***computational*** and ***statistical*** techniques to address or gain insight into some problem in the **real world**



Actionable intelligence



Challenges

- Retrieve or infer information from high-dimensional/large-scale data

2.5 exabytes of data
are generated every day (2012)

exabyte → zettabyte → yottabyte...??

We're interested in the ***information*** rather than the data

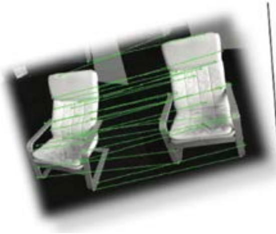
limited processing ability (computation, storage, ...)

Challenges:

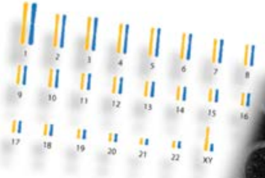
- High computational cost
- Only limited memory is available
- Do NOT want to compromise statistical accuracy

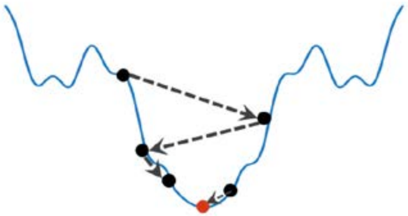
Optimization for data science

- Optimization has transformed algorithm design

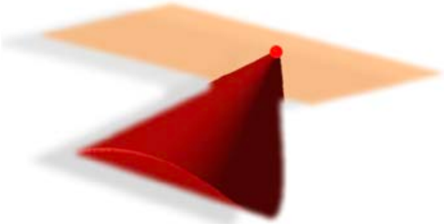


Learning/
Inference





Optimization Problems
(e.g., empirical risk minimization,
maximum likelihood estimation)



(Convex) optimization is *almost* a tool

Optimization problem

- General optimization problem in standard form:

$$\begin{aligned} \text{minimize} \quad & f_0(\boldsymbol{x}) \\ \text{subject to} \quad & f_i(\boldsymbol{x}) \leq 0, i = 1, \ldots, m \end{aligned}$$

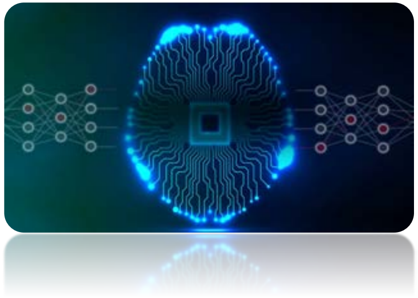
  - $\boldsymbol{x} = (x_1, \ldots, x_n)$: optimization variables
  - $f_0 : \mathbb{R}^n \to \mathbb{R}$: objective function
  - $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, m$: constraint functions

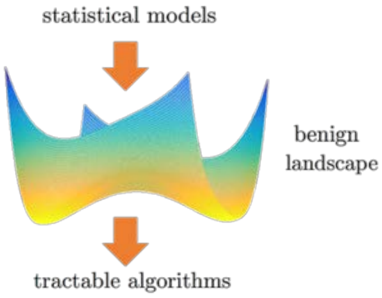
- **Goal:** find optimal solution $\boldsymbol{x}^\star$ minimizing f_0 while satisfying constraints
- **Three basic elements:** 1) variables, 2) constraints, and 3) objective

High-dimensional data analysis

(big) data

Models: (deep) machine learning





Methods:

1. Large-scale optimization
2. High-dimensional statistics
3. Device-edge-cloud computing

Linear model

- Let $\boldsymbol{x}^\natural \in \mathbb{R}^d$ be an unknown structured sparse signal
  - Individual sparsity for compressed sensing
- Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function that reflects structure, e.g., ℓ_1-norm
- Let $\boldsymbol{A} \in \mathbb{R}^{m \times d}$ be a measurement operator
- **Observe** $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{x}^\natural$
- Find estimate $\hat{\boldsymbol{x}}$ by solving **convex program**

$$\text{minimize} \quad f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$$

- **Hope:** $\hat{\boldsymbol{x}} = \boldsymbol{x}^\natural$

MR scanner

MR image

Bilinear model

image deblurring

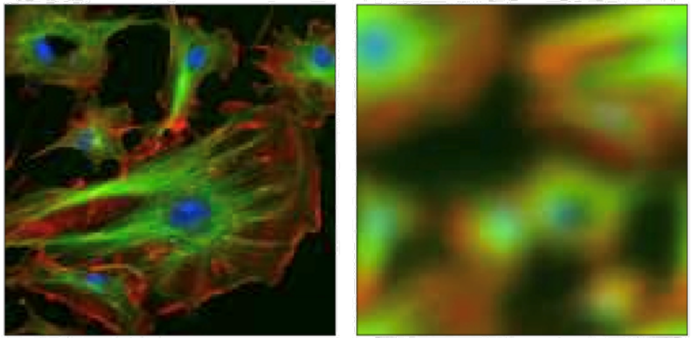


Fig. credit: Romberg

multipath in wireless comm

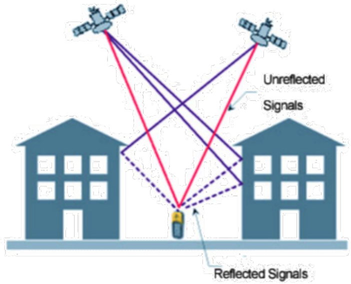


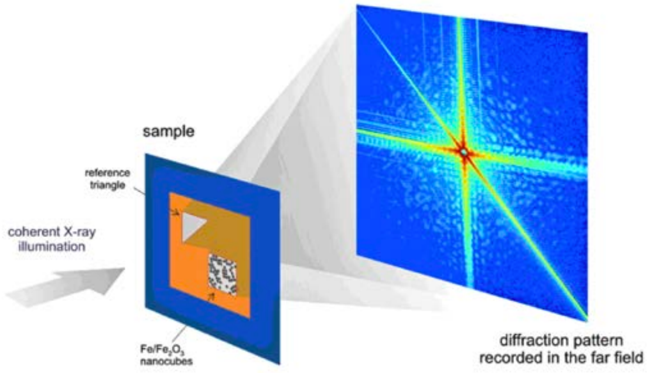
Fig. credit: EngineeringsALL

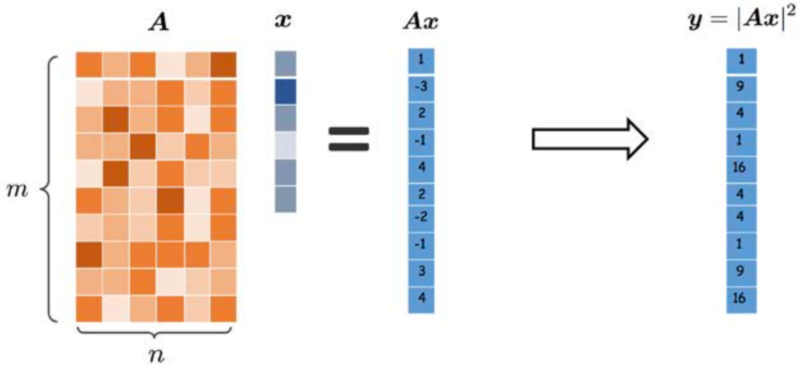
- **Blind deconvolution:** reconstruct two signals from their convolution

$$\text{find} \quad \boldsymbol{x}, \boldsymbol{h} \quad \text{subject to} \quad z_i = \boldsymbol{b}_i^* \boldsymbol{h} \boldsymbol{x}^* \boldsymbol{a}_i, \quad 1 \le i \le m$$

Quadratic model

- **Phase retrieval:** recover signal from intensity (missing phase)





- Recover $\boldsymbol{z}^\natural \in \mathbb{R}^n$ from m random quadratic measurements

$$\text{find} \quad \boldsymbol{z} \qquad \text{subject to} \;\; y_r = |\langle \boldsymbol{a}_r, \boldsymbol{z} \rangle|^2, \quad r = 1, 2, \ldots, m$$

Low-rank model

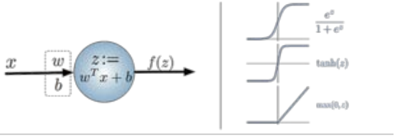
Fig. credit: Candès

- Given partial samples Ω of a low-rank matrix $M^\natural$, fill in missing entries

$$\underset{M \in \mathbb{C}^{m \times n}}{\text{minimize}} \quad \text{rank}(M) \quad \text{subject to} \quad Y_{i,k} = M_{i,k}, \quad (i,k) \in \Omega$$

Deep models

- **Data:** n observations $\{\boldsymbol{x}_i, y_i\}_{i=1}^n \in \mathcal{X} \times \mathcal{Y}$
- **Prediction function:** $h(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^d$
  - ➢ linear predictions: $h(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \Phi(\boldsymbol{x})$ using features $\Phi(\boldsymbol{x})$
  - ➢ neural networks: $h(\boldsymbol{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}_m^T \sigma(\boldsymbol{\theta}_{m-1}^T \sigma(\cdots \boldsymbol{\theta}_2^T \sigma(\boldsymbol{\theta}_1^T \boldsymbol{x})))$



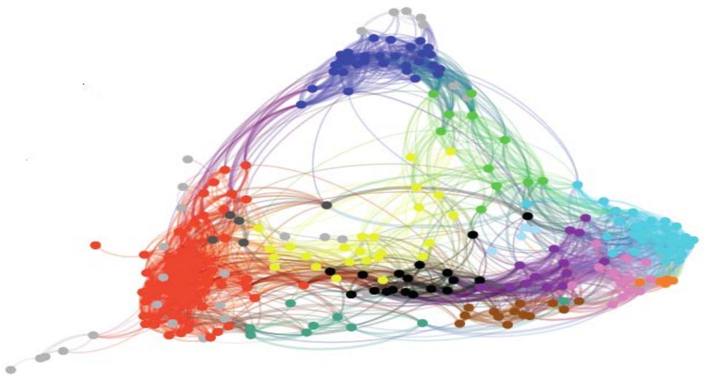
- Estimating $\boldsymbol{\theta}$ parameters is an optimization problem (ℓ: loss function)

$$\text{minimize} \quad f(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} \ell(h(\boldsymbol{x}_i, \boldsymbol{\theta}), y_i) \quad \text{subject to} \quad \mathcal{R}(\boldsymbol{\theta}) \leq \tau$$

➢ $\mathcal{R}$:regularization function encoding prior information (e.g., sparse) on $\boldsymbol{\theta}$

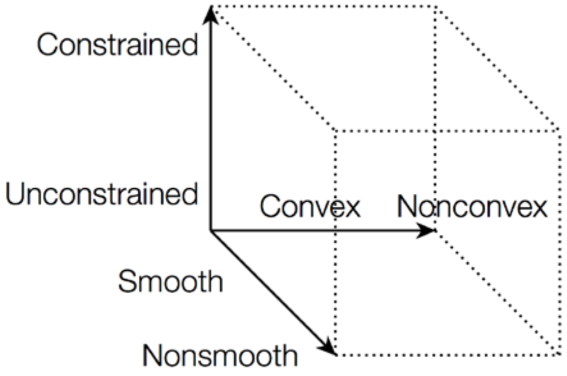
Key benefits of looking at problems in AI as optimization problems:
separate out the *definition* of the problem from the *method for solving it!*

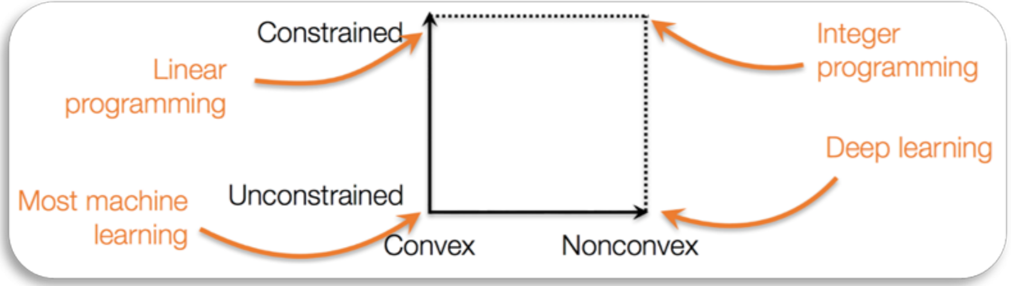
Vignettes B: ***Large-Scale Optimization***



Classes of optimization problems

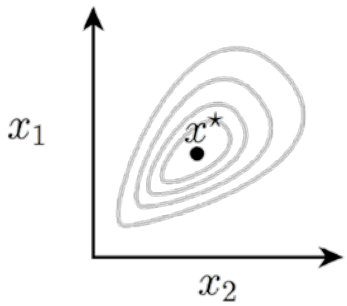
- Types of optimization problems: linear programming, nonlinear programming, integer programming, geometric programming, …

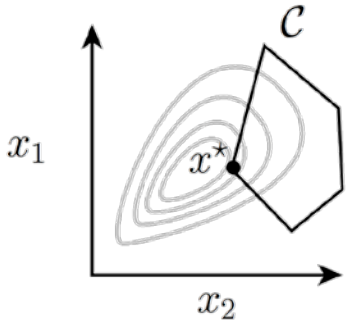




- We focus on three dimensions: ***unconstrained vs. constrained, convex vs. nonconvex, and smooth vs. nonsmooth***

Constrained vs. unconstrained optimization

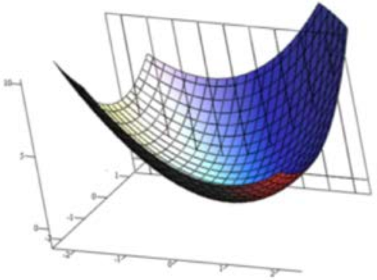


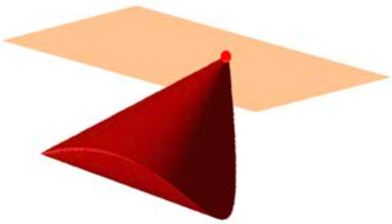


- **Unconstrained optimization:** every point $\boldsymbol{x} \in \mathbb{R}^n$ is feasible, so only focus is on minimizing $f(\boldsymbol{x})$
- **Constrained optimization:** it may be difficult to even *find* a feasible point $\boldsymbol{x} \in \mathcal{C}$

Typically leads to different classes of algorithms

Convex vs. nonconvex optimization





Convex optimization:

1) All local optima are global optima
2) Can be solved in polynomial-time

"... the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity"

— R. Rockafellar '1993

Deterministic vs. stochastic optimization

- **Stochastic optimization**

$$\text{minimize} \quad f(\boldsymbol{x}) := \mathbb{E}[F(\boldsymbol{x}, \boldsymbol{\xi})] \qquad \text{subject to} \quad \boldsymbol{x} \in \mathcal{X}$$

  - f: loss; $\boldsymbol{x}$: parameters; $\boldsymbol{\xi}$: data samples

- **Example:** supervised machine learning (finite-sum problems)

$$\text{minimize} \quad f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \ell(b_i - \boldsymbol{a}_i^T \boldsymbol{x})$$

  - Data observations: $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^d \times \mathbb{R}$; loss function: $\ell : \mathbb{R}^d \to \mathbb{R}$

- **Stochastic gradient:** $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f_{i(k)}(\boldsymbol{x}_k)$

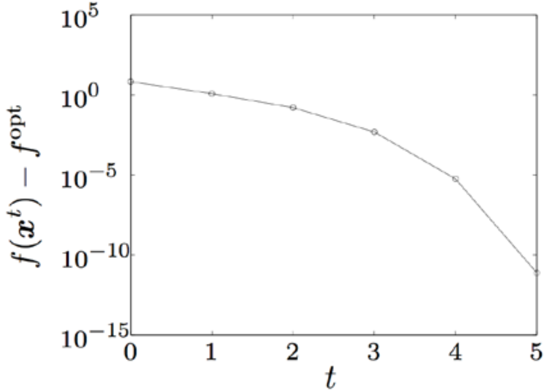
  - $i(k) \in \{1, 2, \ldots, n\}$ uniformly at random; unbiased estimate: $\mathbb{E}[\nabla f_{i(k)}] = \nabla f$

Scaling issues: solvability vs. scalability

- Polynomial-time algorithms might be *useless* in large-scale applications
- **Example:** Newton's method

$$\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x})$$

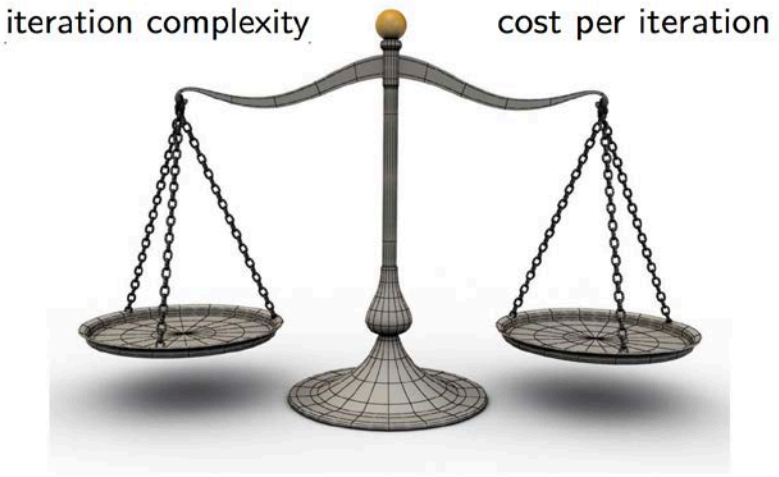
$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - (\nabla^2 f(\boldsymbol{x}^t))^{-1}\nabla f(\boldsymbol{x}^t)$$



- Attains ϵ accuracy within $\mathcal{O}(\log\log\frac{1}{\epsilon})$ iterations; requires $\nabla^2 f(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$
- A single iteration may last forever; prohibitive storage requirement

Iteration complexity vs. per-iteration cost

computational cost = iteration complexity (#iterations) x cost per iteration



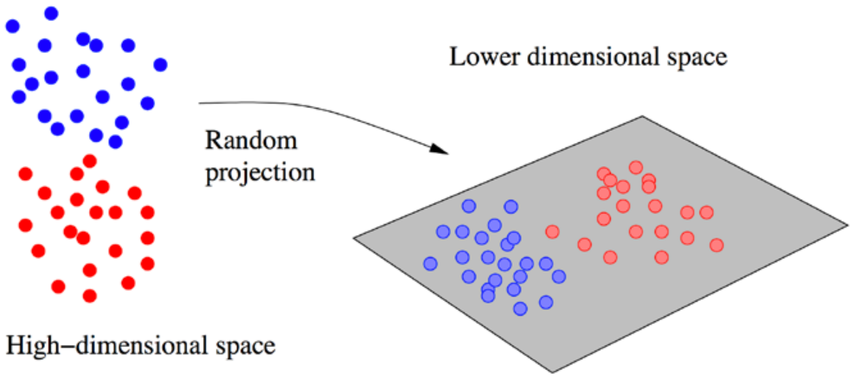
Large-scale problems call for methods with *cheap iterations*

First-order methods

- **First-order methods:** methods that exploit only information on function values and (sub)gradients without using Hessian information
    - cheap iterations
    - low memory requirements

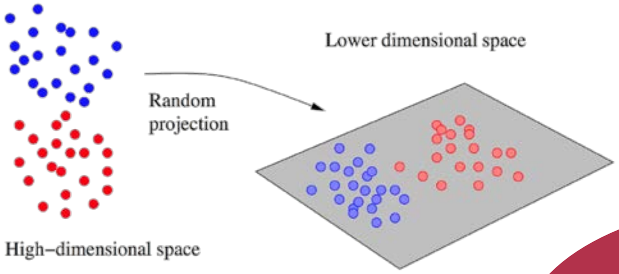
Randomized and approximation methods

- **Optimization for high-dimensional data analysis:** polynomial-time algorithms often not fast enough: further *approximations* are essential

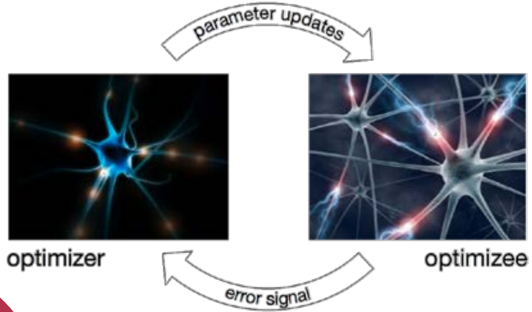


- **Randomized and stochastic methods:** project data into subspace, and solve reduced dimension problem

Advanced large-scale optimization

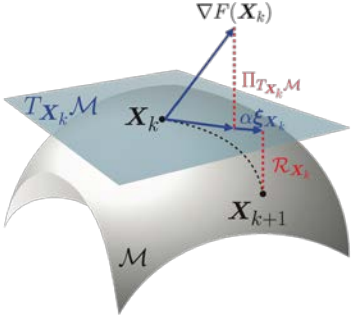


randomized methods

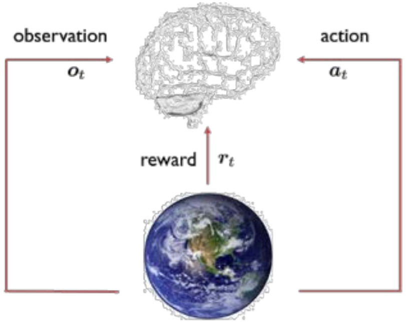


learning to optimize

Goal: scalable, real-time, parallel, distributed, automatic, etc.

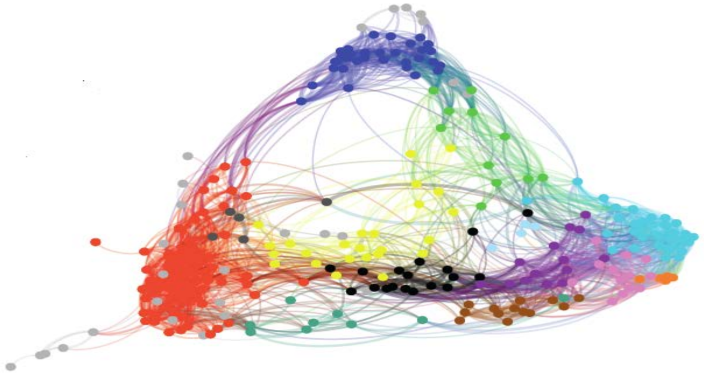


nonconvex optimization on manifold



deep reinforcement learning

Vignettes C: High-Dimensional Statistics

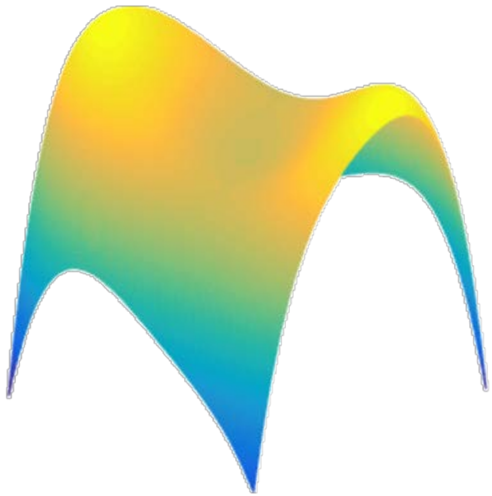


Nonconvex problems are everywhere

- Empirical risk minimization is usually nonconvex

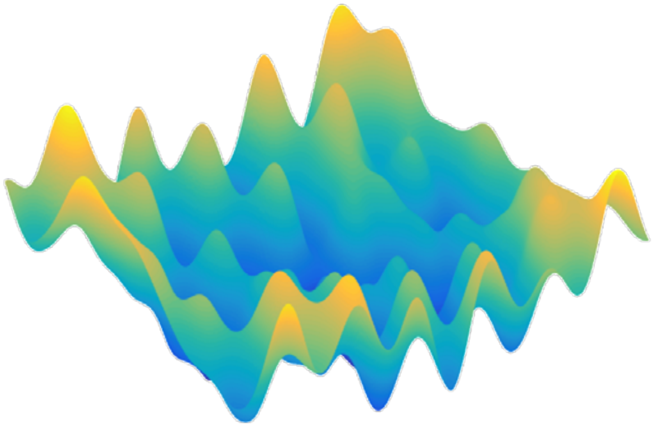
$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x}; \boldsymbol{\theta})$$

  - low-rank matrix completion
  - blind deconvolution/demixing
  - dictionary learning
  - phase retrieval
  - mixture models
  - deep learning
  - …



Nonconvex optimization may be super scary

- **Challenges:** saddle points, local optima, bumps,…



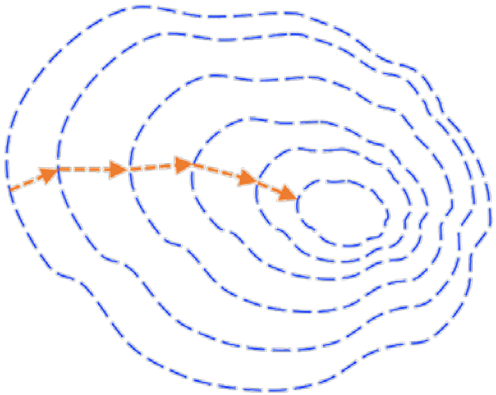


Fig. credit: Chen

- **Fact:** they are usually solved on a daily basis via simple algorithms like (stochastic) gradient descent

Statistical models come to rescue

- **Blessings:** when data are generated by certain statistical models, problems are often much nicer than worst-case instances

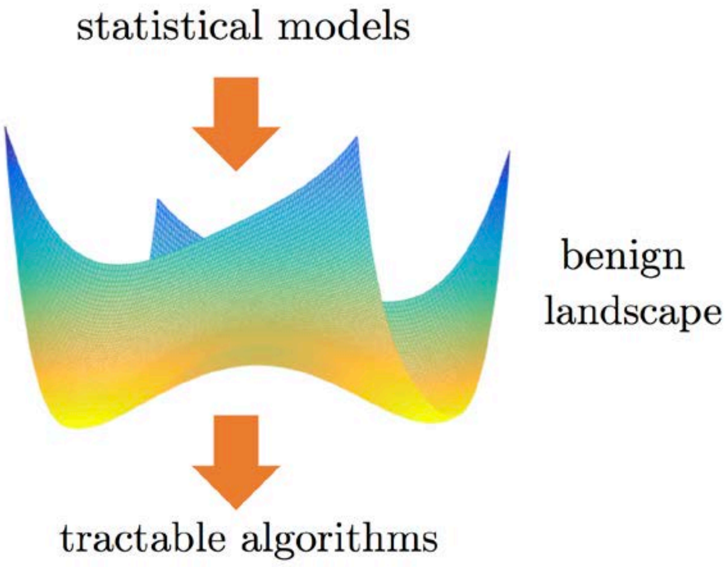


Fig. credit: Chen

Convex geometry

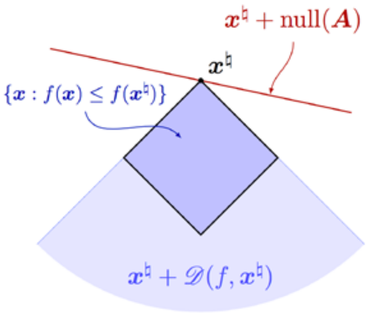
- Compressive sensing: find sparse estimate $\hat{\boldsymbol{x}}$ by solving

$$\text{minimize} \quad f(\boldsymbol{x}) \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{x} = \boldsymbol{z}$$

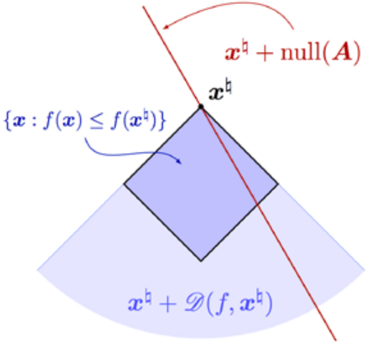
MR scanner

MR image

- Geometry of linear inverse problems



Success!

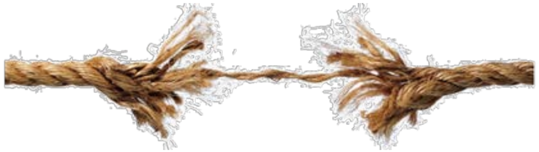


Failure!

Global geometry

- **Proposal:** separation of landscape analysis and generic algorithm design

landscape analysis (statistics)



generic algorithms (optimization)

all local minima are global minima

- dictionary learning (Sun et al. '15)
- phase retrieval (Sun et al. '16)
- matrix completion (Ge et al. '16)
- synchronization (Bandeira et al. '16)
- inverting deep neural nets (Hand et al. '17)
- ...

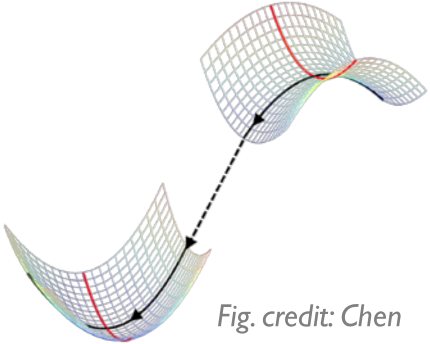


Fig. credit: Chen

all the saddle points can be escaped

- gradient descent (Lee et al. '16)
- trust region method (Sun et al. '16)
- perturbed GD (Jin et al. '17)
- cubic regularization (Agarwal et al. '17)
- Natasha (Allen-Zhu '17)
- ...

Local geometry

- **Initialize** within local basin sufficiently close to ground-truth (i.e., strongly convex, no saddle points/ local minima)
- **Iterative refinement** via some iterative optimization algorithms

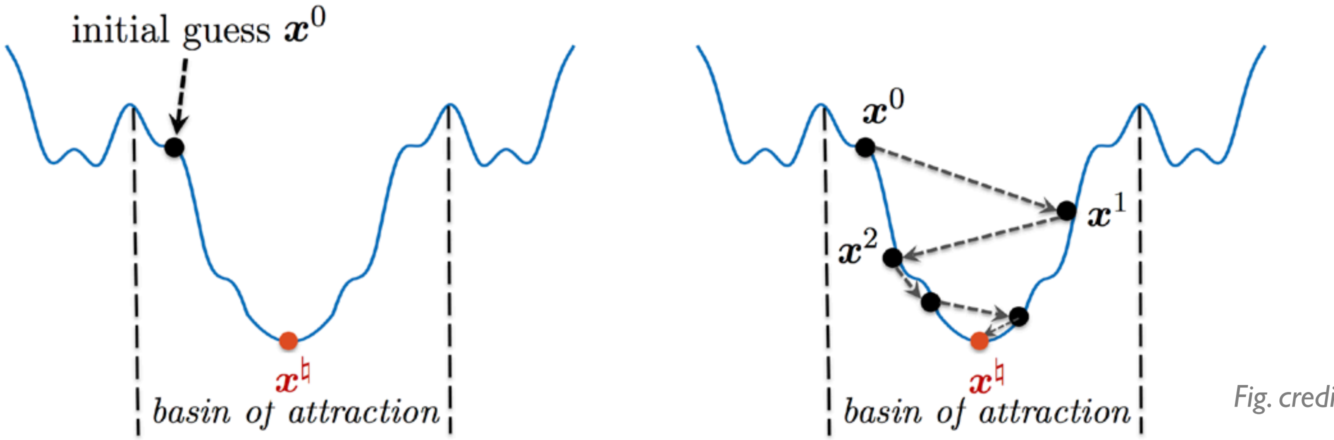
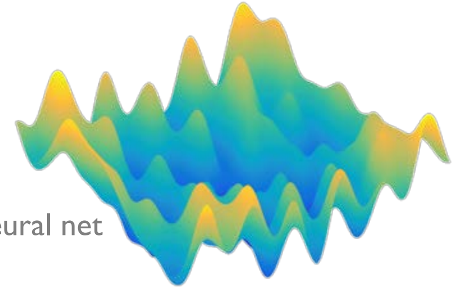


Fig. credit: Chen

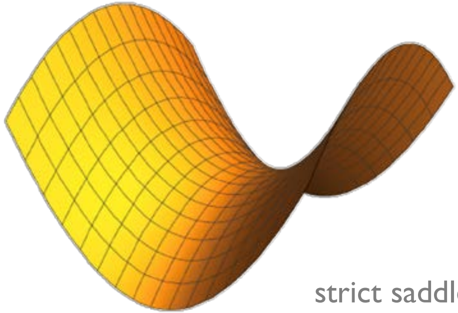
Optimization meets statistics



1-layer neural net

big data & deep learning

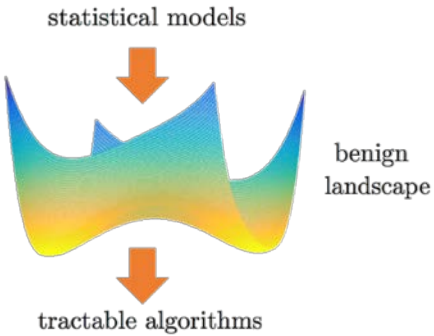
statistical models



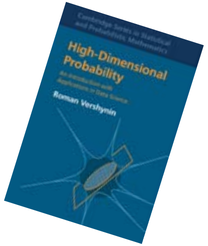
strict saddle points

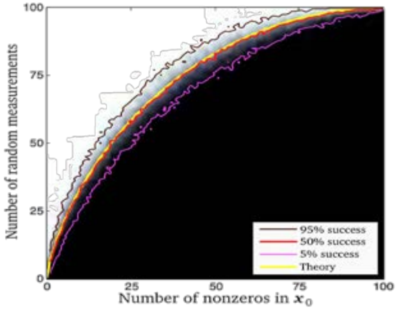
nonconvex optimization may be super scary

benign geometry: no spurious local optima



high-dimensional
probability & statistics





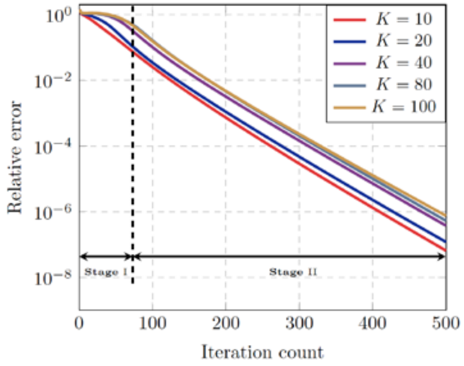
framework: high-dimensional data analysis

Goals: data sizes, expressivity, information propagations, etc.

Case study: bilinear model

- Demixing from bilinear measurements

$$\begin{array}{ll} \text{find} & \{\boldsymbol{x}_i\}, \{\boldsymbol{h}_i\} \\ \text{subject to} & z_j = \sum_{i=1}^{s} \boldsymbol{b}_j^* \boldsymbol{h}_i \boldsymbol{x}_i^* \boldsymbol{a}_{ij} \end{array}$$



- **Applications**

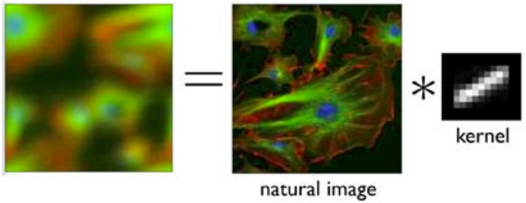
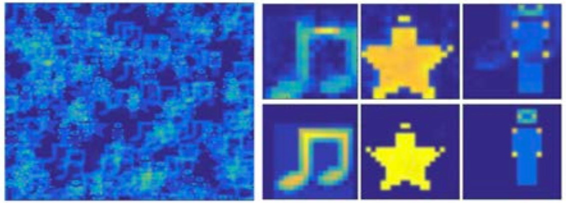
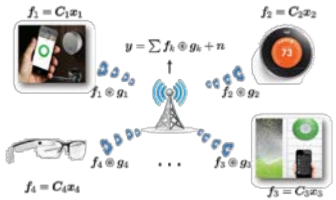


image deblurring



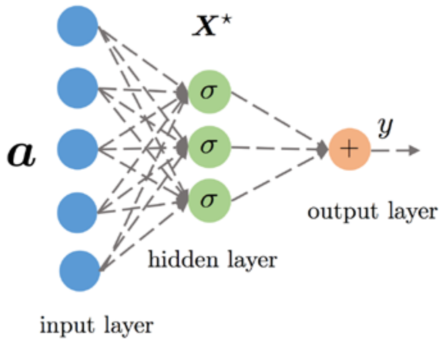
convolutional dictionary learning

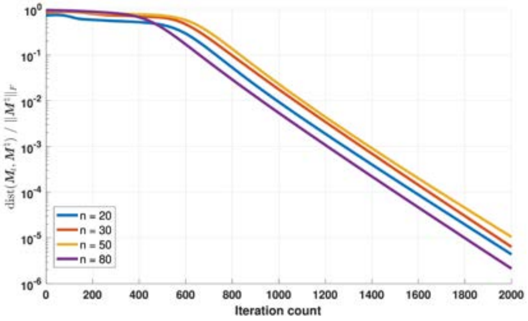


low-latency communication

Case study: deep learning model

- Learning neural networks with quadratic activation

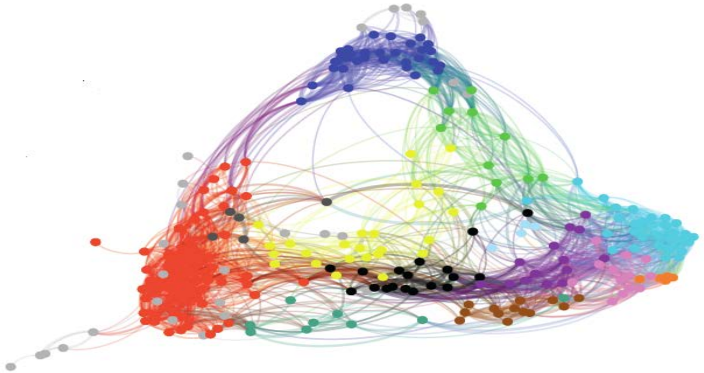




- input features: $\boldsymbol{a}$; weights: $\boldsymbol{X}^\star = [\boldsymbol{x}_1^\star, \cdots, \boldsymbol{x}_r^\star]$

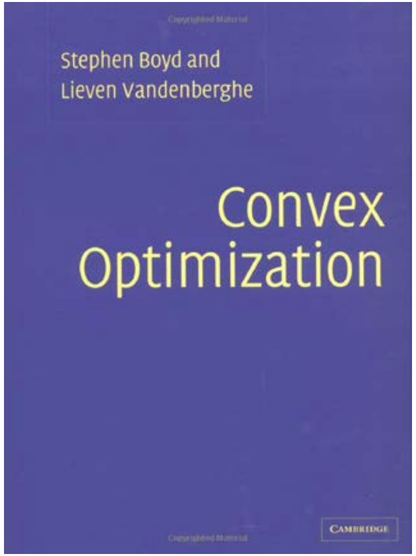
output: $$y = \sum_{i=1}^{r} \sigma(\boldsymbol{a}^\top \boldsymbol{x}_i^\star) \overset{\sigma(z)=z^2}{:=} \sum_{i=1}^{r} (\boldsymbol{a}^\top \boldsymbol{x}_i^\star)^2$$

Topics and Grading



Theoretical foundations

- **Main topics:** convex sets, convex functions, convex problems, Lagrange duality and KKT conditions, disciplined convex programming



Convex Optimization, by S. Boyd and L. Vandenberghe, Cambridge University Press, 2003.

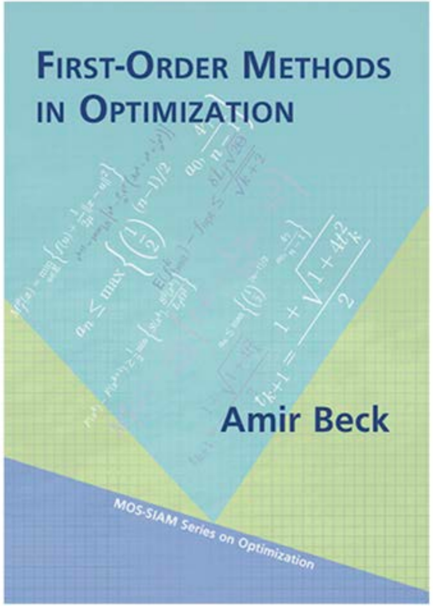
Applications in financial engineering

- **Main topics:** portfolio optimization, factor models, time series modeling, robust portfolio optimization, risk-parity portfolio, index tracking, pairs trading

Optimization Methods in Finance, by G. Cornuejols, J. Pena, and R. Tutuncu, Cambridge University Press, 2018.

First-order methods

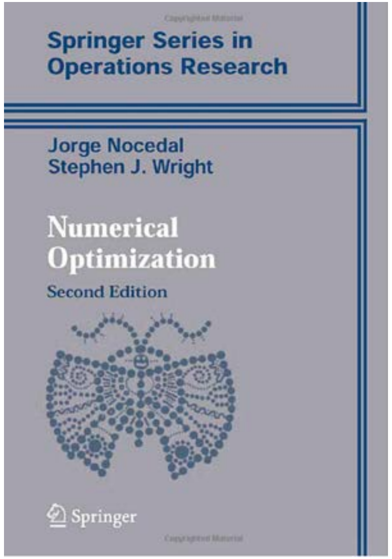
- **Main topics:** gradient methods, subgradient methods, proximal methods



First-order Methods in Optimization, by A. Beck, MOS-SIAM Series on Optimization, 2017.

Second-order methods

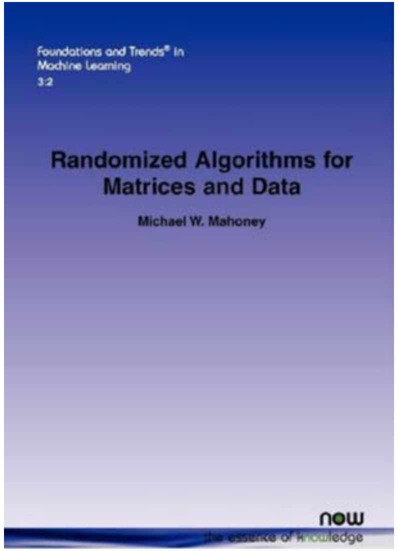
- **Main topics:** Newton method, interior-point methods, quasi-Newton methods



Numerical Optimization, by J. Nocedal and S. Wright, Springer-Verlag, 2006.

Stochastic and randomized methods

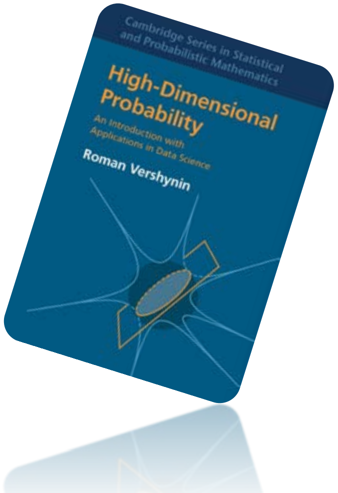
- **Main topics:** stochastic gradient methods, stochastic Newton methods, randomized sketching methods, randomized linear algebra



Lecture Notes on Randomized Linear Algebra
, by Mahoney, Michael, 2016.

Applications in machine learning

- **Main topics (*statistics meets optimization*):**
  - Convex geometry: phase transitions (compressive sensing, matrix sensing)
  - Local geometry: basin of attraction (phase retrieval, blind deconvolution)
  - Global geometry: escape saddle points (matrix completion, neural networks)



High-Dimensional Probability: An Introduction with Applications in Data Science, by Roman Vershynin, Cambridge University Press, 2018.

Prerequisites

- **Warning:** there will be quite a few THEOREMS and PROOFS …

- Basic linear algebra
- Basic probability
- A programming language (e.g. Matlab, Python, …)

Somewhat surprisingly, most proofs rely only on basic linear algebra and elementary recursive formula

Grading

- **Homeworks:** 6 homework sets
- **Final exam:** 3-hours open book exam
- **Course project:**
  - either individually or in groups of two/three
  - list of topics (before May 1); report & slides (end of 16-th week)

$$\text{grade} = 0.2H + 0.4E + 0.4P$$

  - H: homework; E: final exam; P: project

Course information

- **Instructor:** Yuanming Shi (http://shiyuanming.github.io)
  - Email: shiym@shanghaitech.edu.cn
  - Office location: Room 1C-403C, SIST Building
  - Office hours: by appointments

- **TAs:**
  - Chen Chen, Xiangyu Yang, Qiong Wu, Tao Jiang, Jialin Dong
  - Office hours: TBD

Course information

- Use WeChat as the main mode of electronic communication; please post (and answer) questions there!
- Post all the course materials on Piazza.

Thanks