Lecture: Introduction to Convex Optimization

Zaiwen Wen

Beijing International Center For Mathematical Research
Peking University

http://bicmr.pku.edu.cn/~wenzw/opt-2020-fall.html
wenzw@pku.edu.cn

Acknowledgement: some parts of this slides are based on Prof. Lieven Vandenberghe's lecture notes

课程信息

- 凸优化
- 课程代码：00102906 (研究生），00136660 (本科）
- 教师信息：文再文，wenzw@pku.edu.cn, 微信：wendoublewen
- 助教信息：杨明瀚，柳伊扬
- 上课地点：理教410
- 上课时间：每周周二1∼2节，双周周四1∼2节，8:00am - 9:50am
- 课程主页:
  http://bicmr.pku.edu.cn/~wenzw/opt-2020-fall.html

参考资料

class notes, and reference books or papers

- “Convex optimization”, Stephen Boyd and Lieven Vandenberghe
- “Numerical Optimization”, Jorge Nocedal and Stephen Wright, Springer
- “Optimization Theory and Methods”, Wenyu Sun, Ya-Xiang Yuan
- “Matrix Computations”, Gene H. Golub and Charles F. Van Loan. The Johns Hopkins University Press

教材：最优化：建模，算法与理论(coming soon)

http://bicmr.pku.edu.cn/~wenzw/optbook.html

大致课程计划

- 凸集，凸函数
- 数值代数基础
- 凸优化问题
- 凸优化模型语言和算法软件
- 对偶理论
- 梯度法和线搜索算法, 次梯度法
- 近似点梯度法
- Nesterov加速算法
- 坐标下降算法
- primal-dual 算法
- 交替方向乘子法及其变形
- 内点算法，半光滑牛顿法

课程信息

- 教学方式：课堂讲授
- 成绩评定办法：
  - 6-7次大作业，包括习题和程序：40%
  - 期中闭卷考试: 30%
  - 期末课程项目: 30%
  - 要求：作业要求：i) 计算题要求写出必要的推算步骤，证明题要写出关键推理和论证。数值试验题应该同时提交书面报告和程序，其中书面报告有详细的推导和数值结果及分析。ii) 可以同学间讨论或者找助教答疑，但不允许在讨论中直接抄袭，应该过后自己独立完成。iii) 严禁从其他学生，从互联网，从往年的答案，其它课程等等任何途径直接抄袭。iv) 如果有讨论或从其它任何途径取得帮助，请列出来源。
  - 请在书面报告声明：本项目文件的主要内容没有用在其它课程做为课程项目或作业提交。
  - 如果是两人组队，请明确说明每人负责的部分和内容。

Mathematical optimization

(mathematical) optimization problem

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- $x = (x_1, x_2, \ldots, x_n)$: 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

optimal solution x^* has smallest value of f_0 among all vectors that satisfy the constraints

Solving optimization problems

general optimization problem

- very difficult to solve
- methods involve some compromise, $e.g.$, very long computation time, or not always finding the solution

exceptions : certain problem classes can be solved efficiently and reliably

- least-squares problems
- linear programming problems
- convex optimization problems

Convex optimization problem

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

  if $\alpha + \beta = 1$, $\alpha \geq 0, \beta \geq 0$

- includes least-squares problems and linear programs as special cases

solving convex optimization problems

- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i's and their first and second derivatives
- almost a technology

using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization

Least squares problems

$$\min \quad \|Ax - b\|_2^2$$

solving least-squares problems

- analytical solution: $x^* = (A^T A)^{-1} A^T b$
- reliable and efficient algorithms and software
- computation time proportional to $n^2 k$ $(A \in \mathbb{R}^{k \times n})$; less if structured
- a mature technology

using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility ($e.g.$, including weights, adding regularization terms)

Variants of Least squares

- Ridge regression/Tikhonov regularization

$$\min_x \quad \frac{1}{2}\|Ax-b\|_2^2 + \mu\|x\|_2^2$$

- sparse regularization

$$\min_x \quad \frac{1}{2}\|Ax-b\|_2^2 + \mu\|x\|_1$$

- Lasso/Basis pursuit

$$\min_x \quad \|x\|_1, \text{ s.t. } \|Ax-b\|_2 \leq \epsilon$$

or

$$\min_x \quad \|Ax-b\|_2, \text{ s.t. } \|x\|_1 \leq \sigma$$

- or even under a different norm

$$\min_x \quad \|Ax-b\|_1, \text{ s.t. } \|x\|_1 \leq \sigma$$

Linear programming

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & a_i^T x \leq b_i, \quad i = 1, \ldots, m \end{array}$$

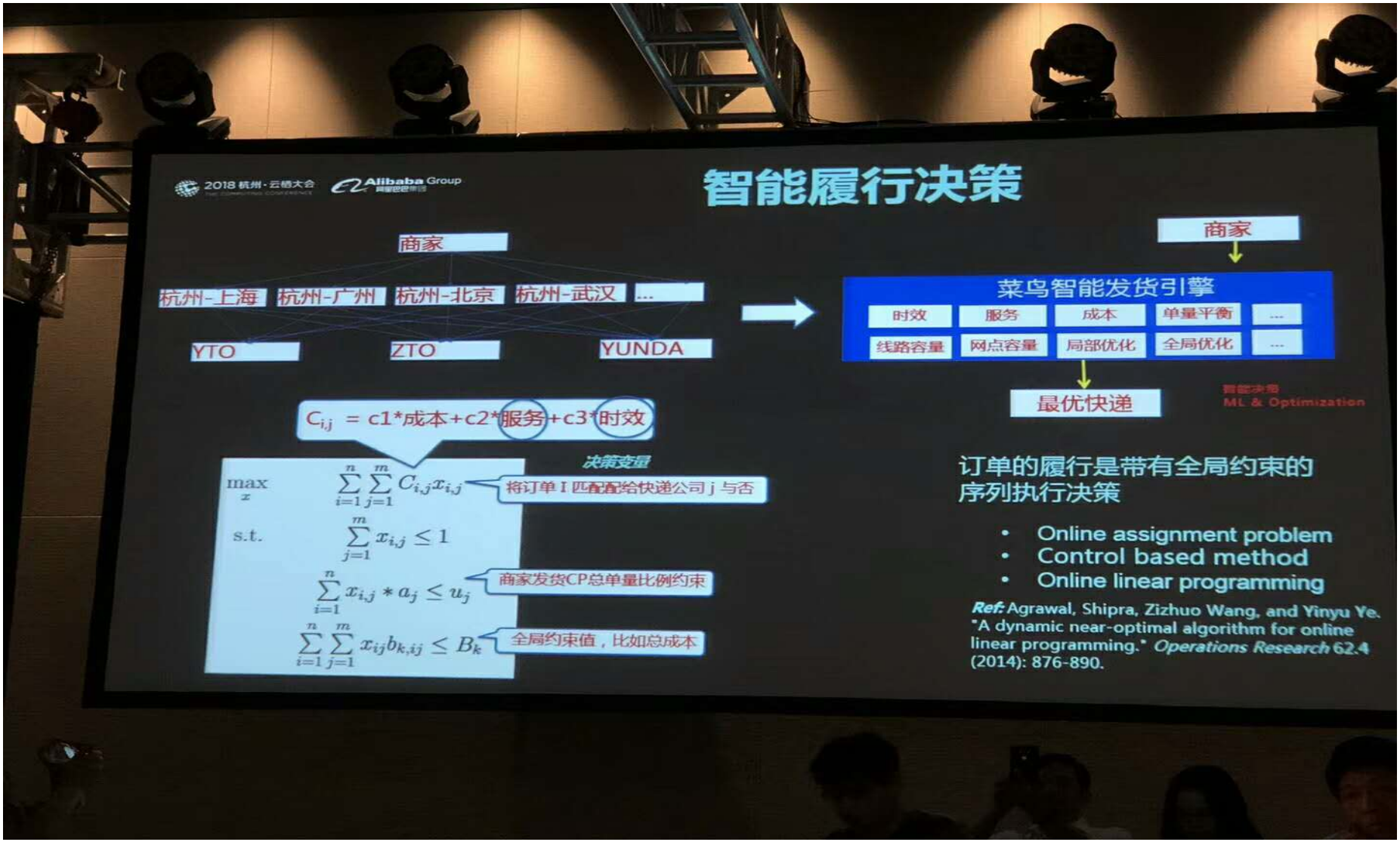
solving linear programs

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \geq n$; less with structure
- a mature technology

using linear programming

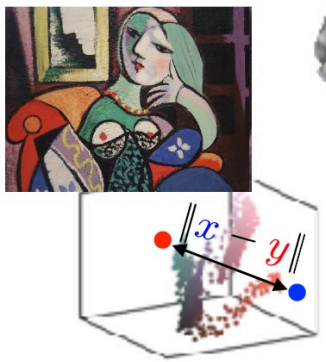
- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs (*e.g.*, problems involving ℓ_1- or ℓ_∞- norms, piecewise-linear functions)

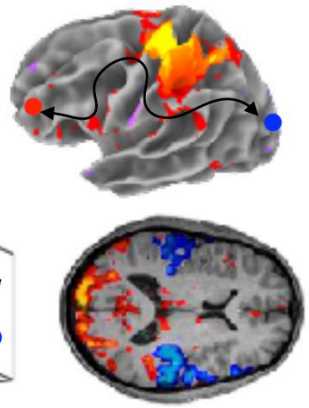
An example of linear programming: 菜鸟

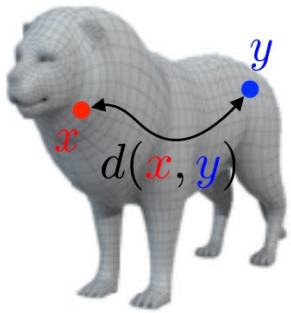


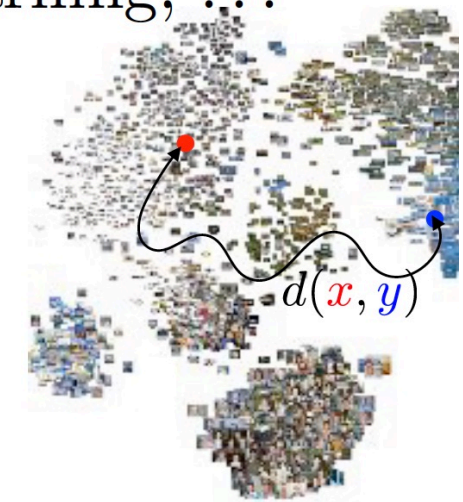
Optimal transport

$\rightarrow$ images, vision, graphics and machine learning, ...










Monge


Kantorovich


Koopmans

Nobel '75


Dantzig


Brenier


Otto


McCann


Villani

Fields '10


Figalli

Fields'18

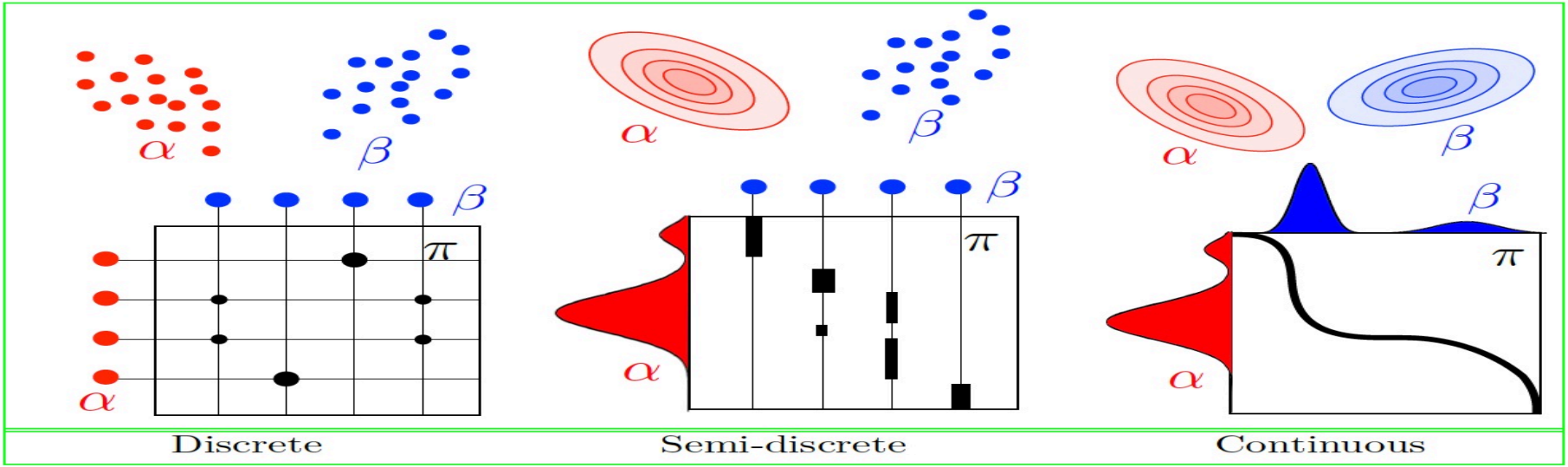
Optimal transport: LP

$$\min_{\pi \in \mathbb{R}^{m \times n}} \quad \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} \pi_{ij}$$

$$\text{s.t.} \quad \sum_{j=1}^{n} \pi_{ij} = \mu_i, \quad \forall i = 1, \ldots, m,$$

$$\sum_{i=1}^{m} \pi_{ij} = \nu_i, \quad \forall j = 1, \ldots, n$$

$$\pi \geq 0$$



Why Optimization in Machine Learning?

Many problems in ML can be written as

$$\min_{x \in \mathcal{W}} \quad \sum_{i=1}^{N} \frac{1}{2} \|a_i^\top x - b_i\|_2^2 + \mu \|w\|_1 \quad \text{linear regression}$$

$$\min_{x \in \mathcal{W}} \quad \frac{1}{N} \sum_{i=1}^{N} \log(1 + \exp(-b_i a_i^\top x)) + \mu \|x\|_1 \quad \text{logistic regression}$$

$$\min_{w \in \mathcal{W}} \quad \sum_{i=1}^{N} \ell(h(x, a_i), b_i) + \mu r(x) \quad \text{general formulation}$$

- The pairs (a_i, b_i) are given data, b_i is the label of the data point a_i
- $\ell(\cdot)$: measures how model fit for data points (avoids under-fitting)
- $r(x)$: regularization term (avoids over-fitting)
- $h(x, a)$: linear function or models constructed from deep neural networks

Loss functions in neural network

Lecture 3 from Fei-Fei Li & Andrej Karpathy & Justin Johnson

- We have some dataset of (x,y)
- We have a **score function:** $s = f(x; W) \overset{\text{e.g.}}{=} Wx$
- We have a **loss function**:

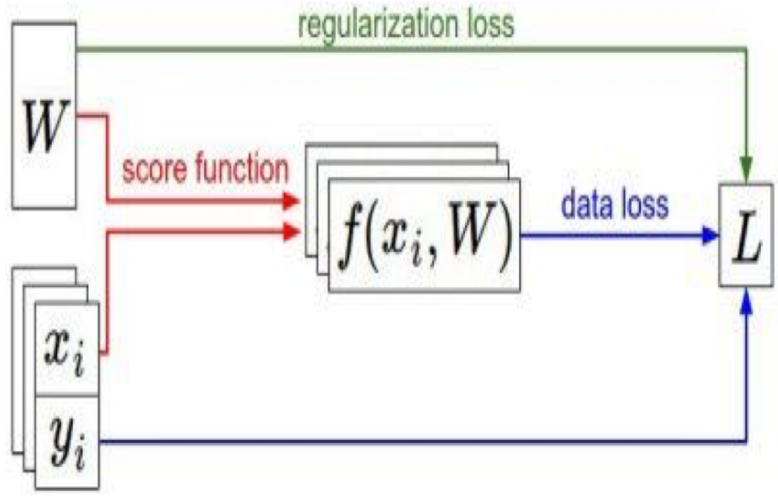
Softmax

$$L_i = -\log\left(\frac{e^{sy_i}}{\sum_j e^{s_j}}\right)$$

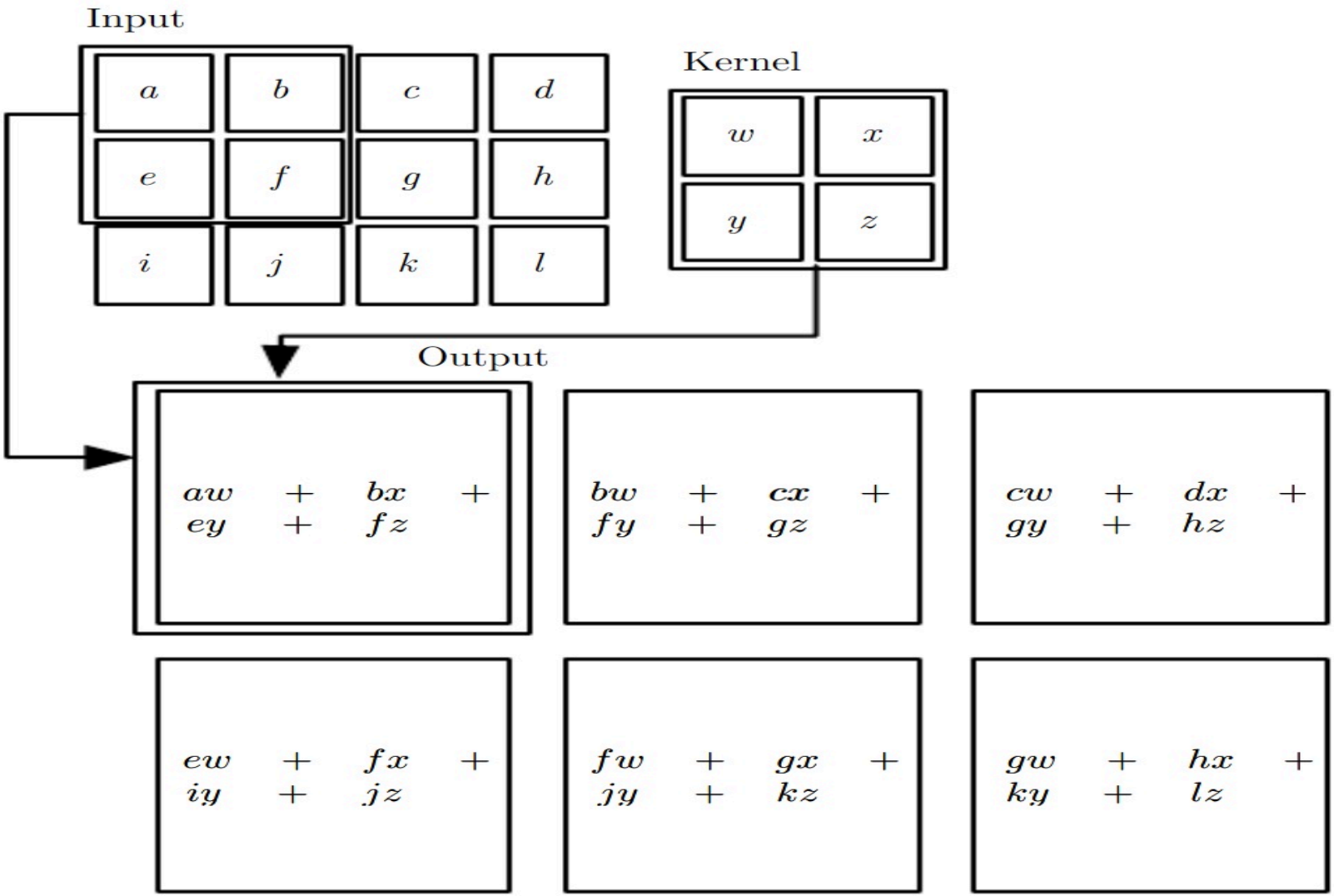
SVM

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i + R(W)$$ Full loss



convolution operator



Loss functions in neural network

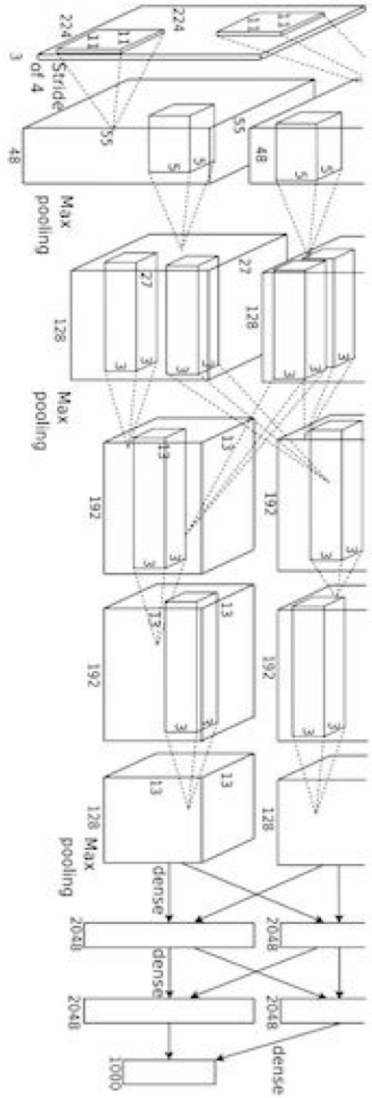
Lecture 4 from Fei-Fei Li & Andrej Karpathy & Justin Johnson

Convolutional Network
(AlexNet)

input image

weights

loss



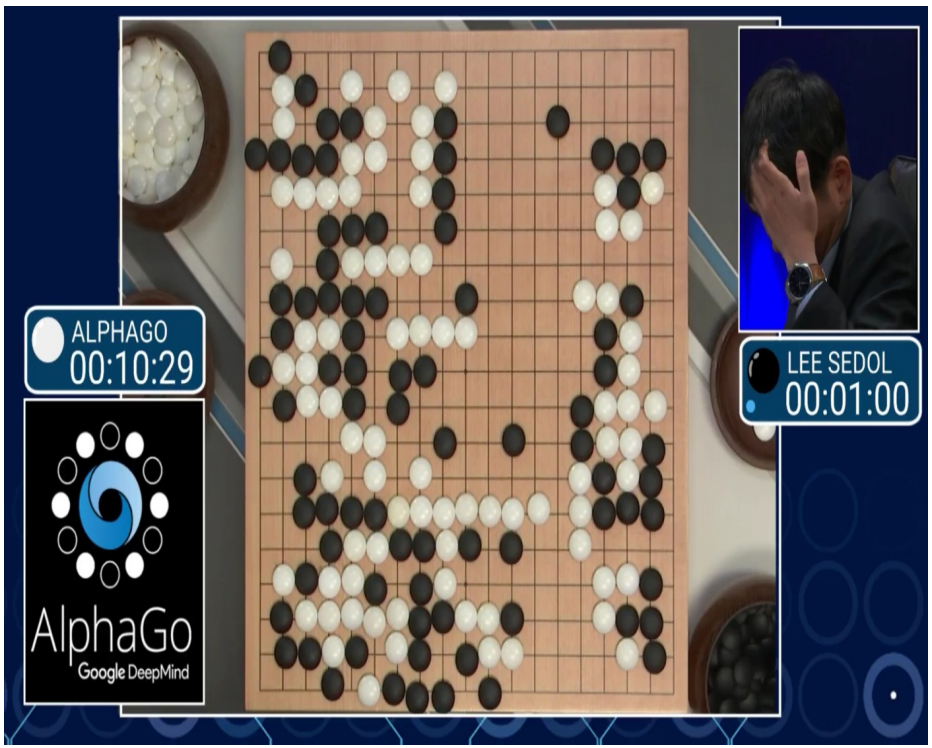
Optimization algorithms in Deep learning

随机梯度类算法

- pytorch/caffe2 里实现的算法有adadelta, adagrad, adam, nesterov, rmsprop, YellowFin
  `https://github.com/pytorch/pytorch/tree/master/caffe2/sgd`
- pytorch/torch 里有：sgd, asgd, adagrad, rmsprop, adadelta, adam, adamax
  `https://github.com/pytorch/pytorch/tree/master/torch/optim`
- tensorflow 实现的算法有：Adadelta, AdagradDA, Adagrad, ProximalAdagrad, Ftrl, Momentum, adam, Momentum, CenteredRMSProp
  具体实现:
  `https://github.com/tensorflow/tensorflow/blob/master/tensorflow/core/kernels/training_ops.cc`

Reinforcement Learning

- AlphaGo: supervised learning + policy gradients + value functions + Monte-Carlo tree search



Definition: MDP

A Markov Decision Process is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$:

- $\mathcal{S}$ is a finite set of states, $s \in \mathcal{S}$
- $\mathcal{A}$ is a finite set of actions, $a \in \mathcal{A}$
- $\mathcal{P}$ is the transition probability distribution.
  probability from state s with action a to state s': $P(s'|s,a)$
  also called the model or the dynamics
- r is a reward function, $r(s,a,s')$
  sometimes just $r(s)$ or r_s^a
  or r_t after time step t
- $\gamma \in [0,1]$ is a discount factor

Optimization model

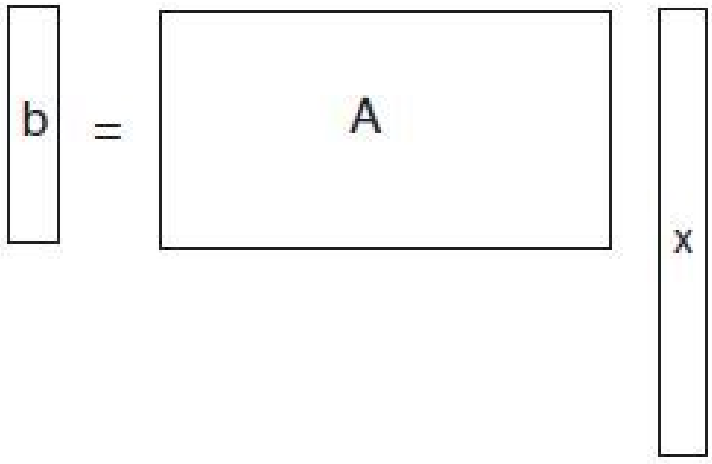
- Maximize the expected total discounted return of an episode

$$\max_{\pi} \quad E[\sum_{t=0}^{\infty} \gamma^t r_t | \pi]$$

$$\text{or}, \max_{\pi} \quad E[R(\tau) | \tau = \{s_0, a_0, s_1, a_1, ...\} \sim \pi]$$

- Policy $\pi(a|s)$ is a probability

压缩感知：从解线性方程组谈起



- $x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m$
- $m \ll n$ linear equations about x

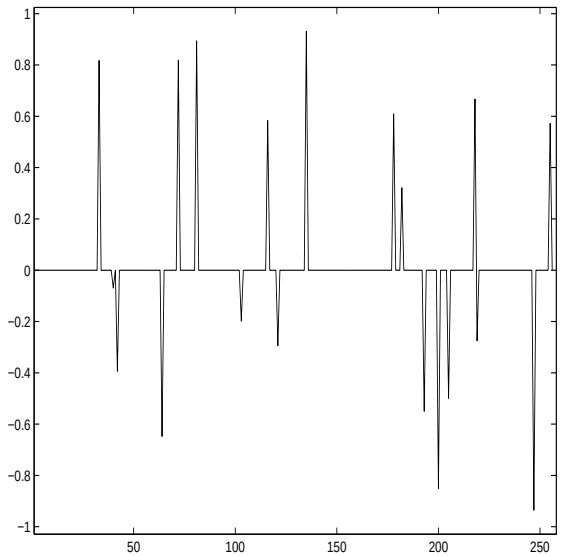
$$Ax = b$$

- want to recover x
- Arises in many fields of science and engineering

Compressive Sensing

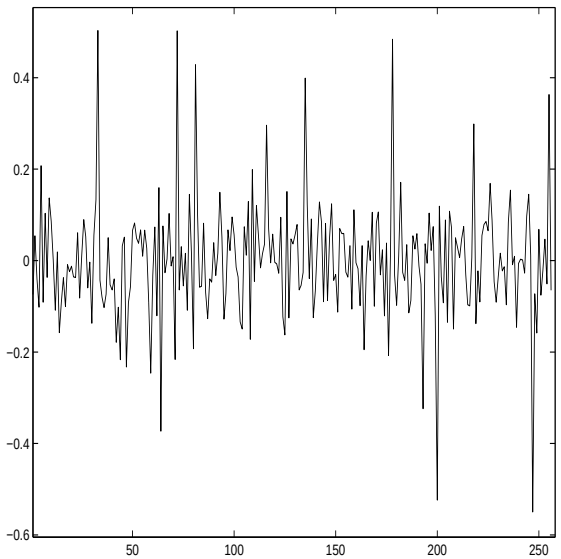
Find the sparest solution

- Given n=256, m=128.
- A = randn(m,n); u = sprandn(n, 1, 0.1); b = A*u;



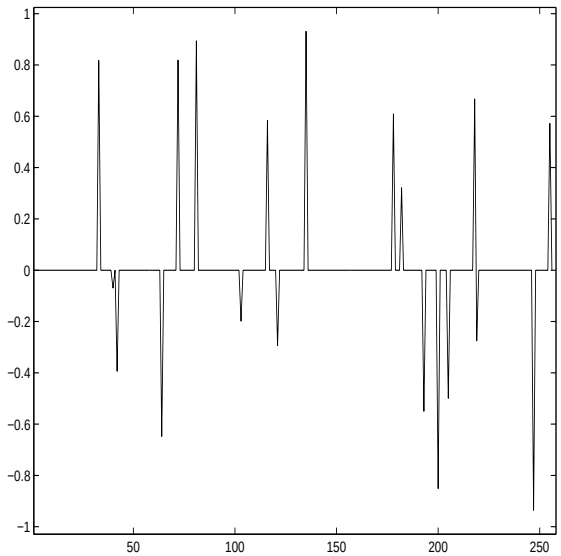
$$\begin{cases} \min\limits_{x} \|x\|_0 \\ \text{s.t. } Ax = b \end{cases}$$

(a) ℓ_0-minimization



$$\begin{cases} \min\limits_{x} \|x\|_2 \\ \text{s.t. } Ax = b \end{cases}$$

(b) ℓ_2-minimization



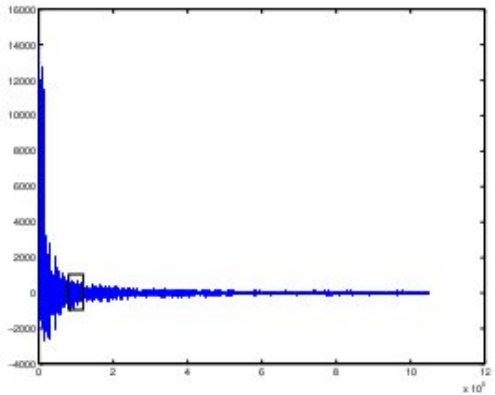
$$\begin{cases} \min\limits_{x} \|x\|_1 \\ \text{s.t. } Ax = b \end{cases}$$

(c) ℓ_1-minimization

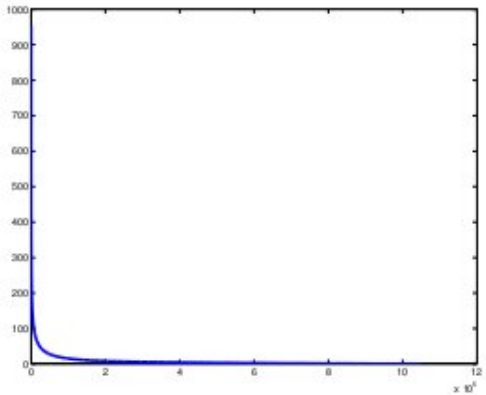
Wavelets and Images (Thanks: Richard Baraniuk)

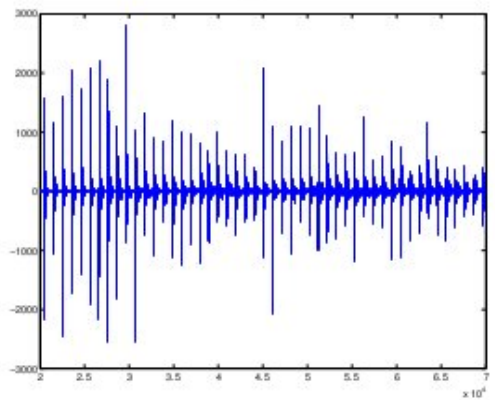
1 megapixel image

wavelet coeffs

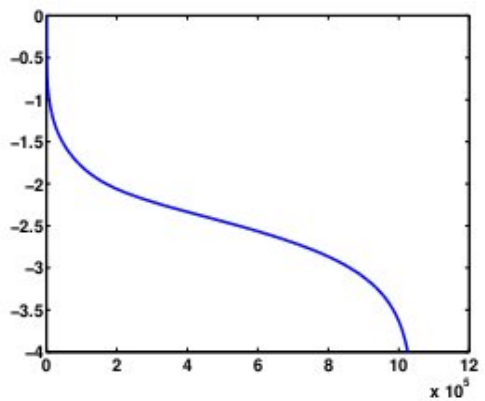


(sorted)





zoom in

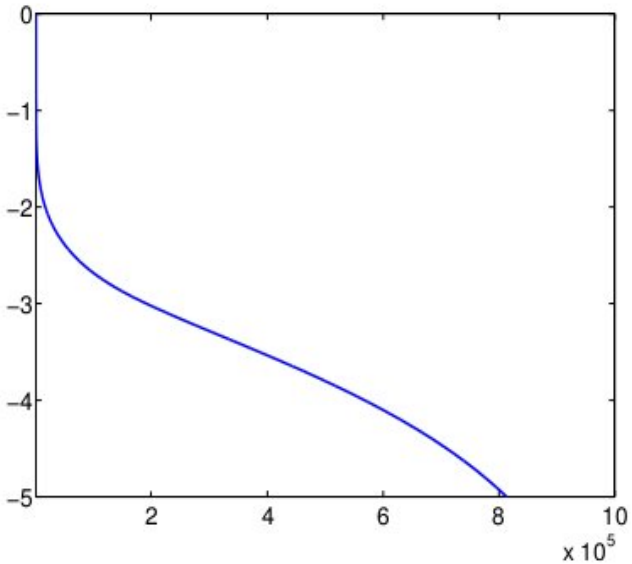


($\log_{10}$ sorted)

Wavelet Approximation (Thanks: Richard Baraniuk)

1 megapixel image

25k term approx



B-term approx error

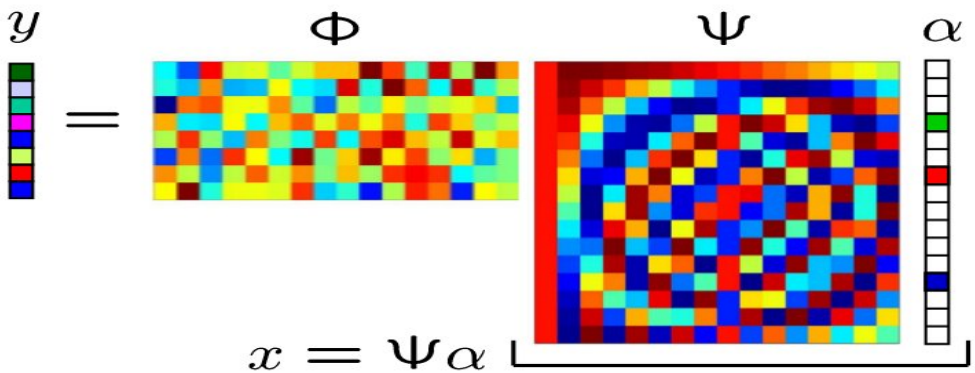
- Within 2 digits (in MSE) with $\approx 2.5\%$ of coeffs
- Original image = f, K-term approximation = f_K

$$\|f - f_K\|_2 \approx .01 \cdot \|f\|_2$$

Compressive sensing

- x is sparsely synthesized by atoms from Ψ, so vector α is sparse
- Random measurements can be used for signals sparse in any basis
- Dictionary Ψ: DCT, wavelets, curvelets, gabor, etc., also their combinations; they have analytic properties, often easy to compute (for example, multiplying a vector takes O(n log n) instead of O(n2))
- can also be numerically learned from training data or partial signal

$$y = \Phi x = \Phi \Psi \alpha$$



Compressive sensing

Given (A, b, Ψ), find the sparsest point:

$$x^* = \arg\min\{\|\Psi x\|_0 : Ax = b\}$$

From combinatorial to convex optimization:

$$\bar{x} = \arg\min\{\|\Psi x\|_1 : Ax = b\}$$

1-norm is sparsity promoting

- Basis pursuit (Donoho et al 98)
- Many variants: $\|Ax - b\|_2 \leq \sigma$ for noisy b
- Theoretical question: when is $\|\cdot\|_0 \leftrightarrow \|\cdot\|_1$?

Restricted Isometry Property (RIP)

Definition (Candes and Tao [2005])

Matrix A obeys the restricted isometry property (RIP) with constant δ_s if

$$(1 - \delta_s)\|c\|_2^2 \leq \|Ac\|_2^2 \leq (1 + \delta_s)\|c\|_2^2$$

for all s-sparse vectors c.

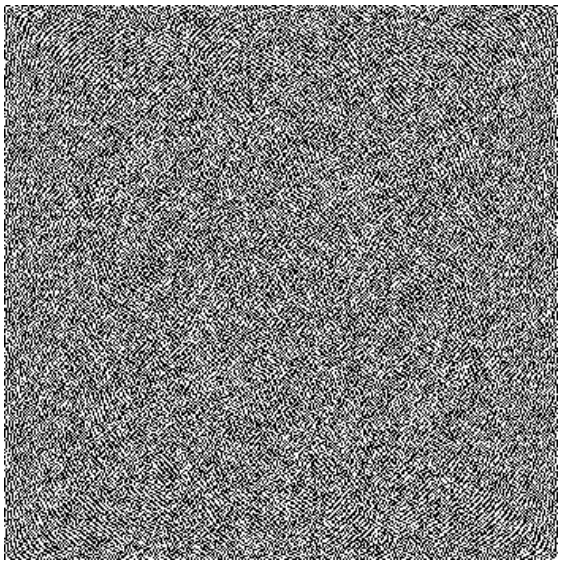
Theorem (Candes and Tao [2006])

If x is a k-sparse and A satisfies $\delta_{2k} + \delta_{3k} < 1$, then x is the unique ℓ_1 minimizer.

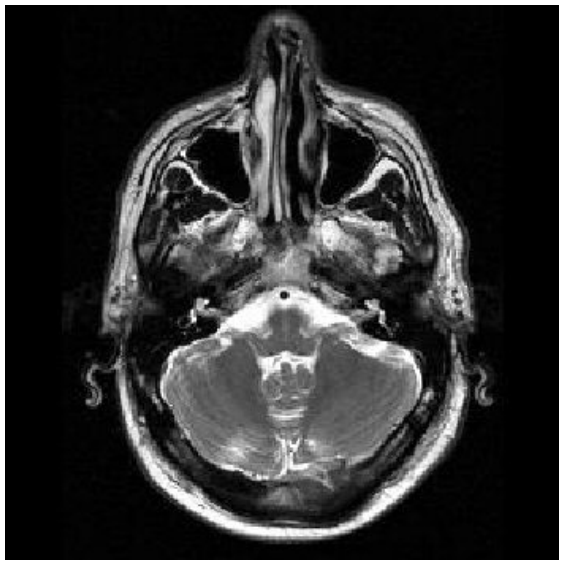
- RIP essentially requires that every set of columns with cardinality less than or equal to s behaves like an orthonormal system.

MRI: Magnetic Resonance Imaging

(a) MRI Scan



(b) Fourier Coefficients



(c) Image

Is it possible to cut the scan time into half?

MRI (Thanks: Wotao Yin)

- MR images often have sparse sparse representations under some wavelet transform Φ
- Solve

$$\min_u \|\Phi u\|_1 + \frac{\mu}{2}\|Ru - b\|^2$$

  R: partial discrete Fourier transform
- The higher the SNR (signal-noise ratio) is, the better the image quality is.

(a) full sampling

(b) 39% sampling, SNR=32.2

MRI: Magnetic Resonance Imaging

(a) full sampling

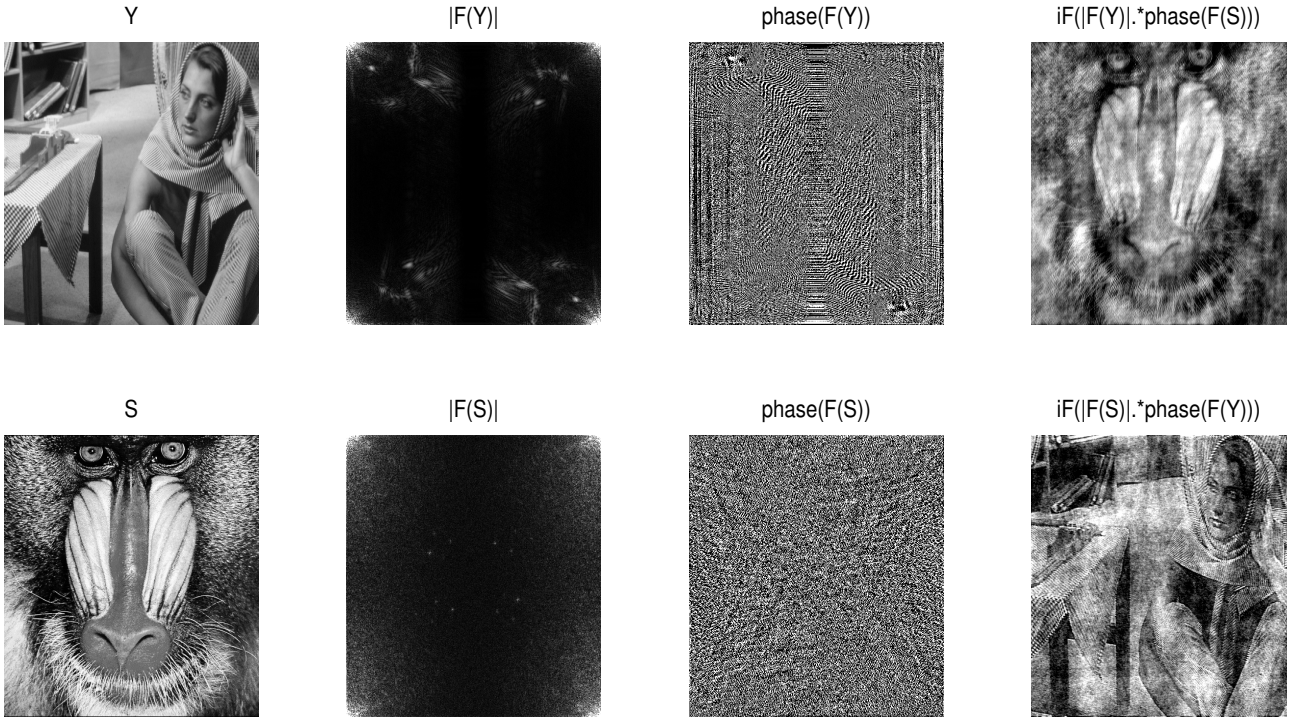
(b) 39% sampling, SNR=32.2

(c) 22% sampling, SNR=21.4

(d) 14% sampling, SNR=15.8

Phase Retrieval

Phase carries more information than magnitude



Question: recover signal without knowing phase?

Classical Phase Retrieval

Feasibility problem

$$\text{find } x \in S \cap \mathcal{M} \text{ or } \text{ find } x \in S_+ \cap \mathcal{M}$$

- given Fourier magnitudes:

$$\mathcal{M} := \{x(r) \mid |\hat{x}(\omega)| = b(\omega)\}$$

  where $\hat{x}(\omega) = \mathcal{F}(x(r))$, $\mathcal{F}$: Fourier transform

- given support estimate:

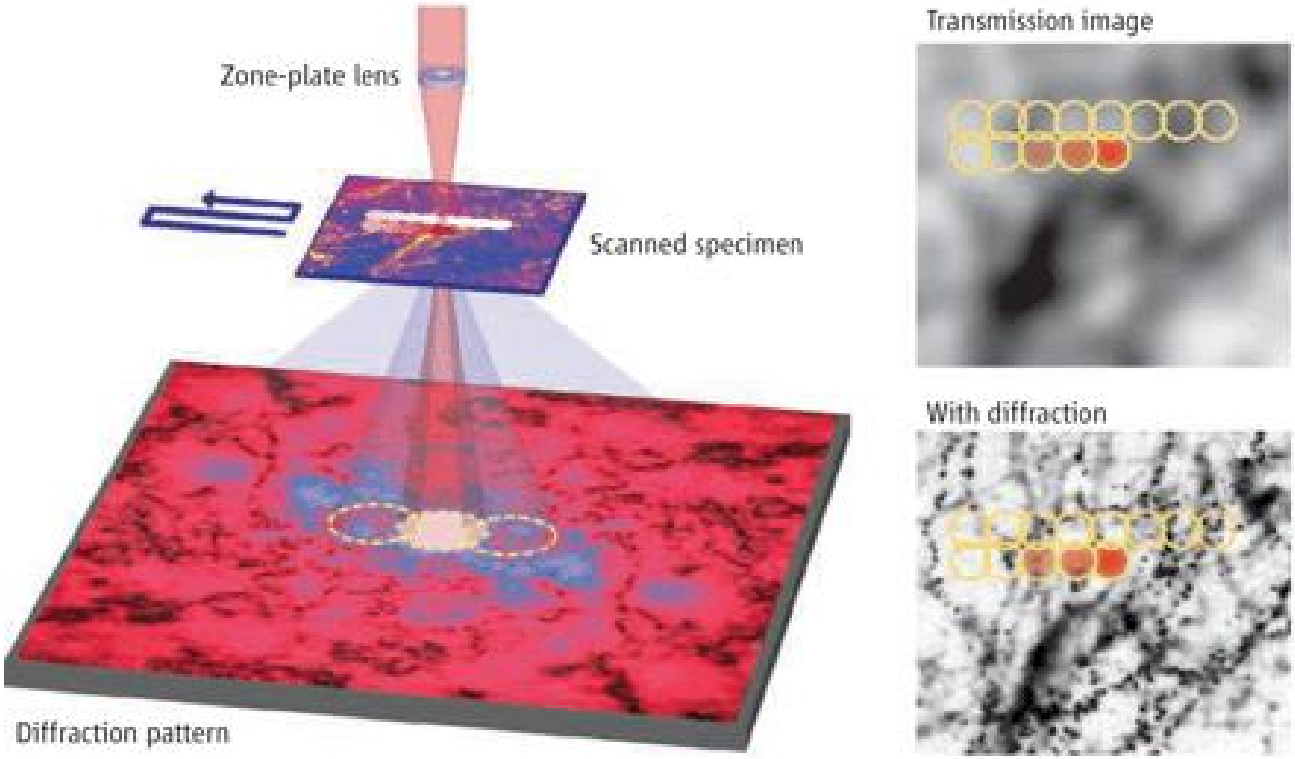
$$S := \{x(r) \mid x(r) = 0 \text{ for } r \notin D\}$$

  or

$$S_+ := \{x(r) \mid x(r) \geq 0 \text{ and } x(r) = 0 \text{ if } r \notin D\}$$

Ptychographic Phase Retrieval (Thanks: Chao Yang)

Given $b_i = |\mathcal{F}(Q_i\psi)|$ for $i = 1, \ldots, k$, can we recover ψ?



Ptychographic imaging along with advances in detectors and computing have resulted in X-ray microscopes with increased spatial resolution without the need for lenses

Recent Phase Retrieval Model Problems

- Given $A \in \mathbf{C}^{m \times n}$ and $b \in \mathbb{R}^m$

$$\text{find } x, \text{ s.t. } |Ax| = b.$$

(Candes et al. 2011b, Alexandre d'Aspremont 2013)

- SDP Relaxation: $|Ax|^2$ is a linear function of $X = xx^*$

$$\begin{aligned} \min_{X \in S_n} \quad & Tr(X) \\ s.t. \quad & Tr(a_i a_i^* X) = b_i^2, i = 1, \ldots, m, \\ & X \succeq 0 \end{aligned}$$

- Exact recovery conditions

The Netflix problem

- Netflix database: about a million users, $25,000$ movies
- People rate movies
- Sparsely sampled entries

Challenge: million dollar award

Complete the ”Netflix matrix"

collaborative filtering, Partially filled out surveys...

Matrix Rank Minimization

Given $X \in \mathbb{R}^{m\times n}$, $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^p$, $b \in \mathbb{R}^p$, we consider

- matrix completion problem:

$$\min \ \operatorname{rank}(X), \ \text{s.t. } X_{ij} = M_{ij}, (i,j) \in \Omega$$

- the matrix rank minimization problem:

$$\min \ \operatorname{rank}(X), \ \text{s.t. } \mathcal{A}(X) = b$$

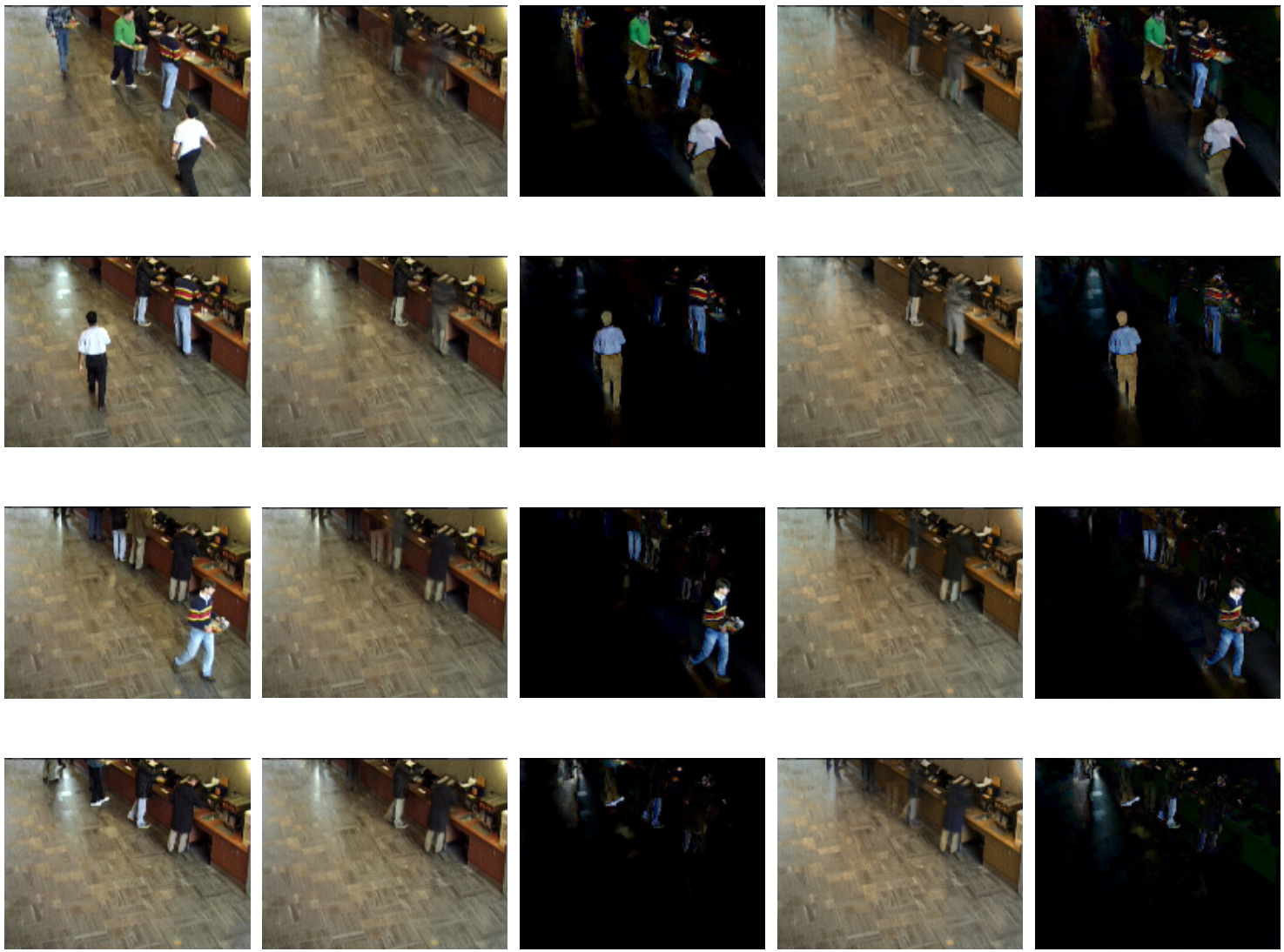
- nuclear norm minimization:

$$\min \ \|X\|_* \ \text{ s.t. } \mathcal{A}(X) = b$$

where $\|X\|_* = \sum_i \sigma_i$ and $\sigma_i = i$th singular value of matrix X.

Video separation

- Partition the video into moving and static parts



Sparse and low-rank matrix separation

- Given a matrix M, we want to find a low rank matrix W and a sparse matrix E, so that $W + E = M$.
- Convex approximation:

$$\min_{W,E} \ \|W\|_* + \mu\|E\|_1, \ \text{s.t. } W + E = M$$

- Robust PCA

Portfolio optimization

- r_i, random variable, the rate of return for stock i
- x_i, the relative amount invested in stock i
- Return: $r = r_1x_1 + r_2x_2 + \ldots + r_nx_n$
- expected return: $R = E(r) = \sum E(r_i)x_i = \sum \mu_i x_i$
- Risk: $V = Var(r) = \sum_{i,j} \sigma_{ij}x_ix_j = x^\top \Sigma x$

$$\begin{aligned} \min \ & \frac{1}{2}x^\top \Sigma x, \\ \text{s.t.} \ & \sum \mu_i x_i \geq r_0 \\ & \sum x_i = 1, \\ & x_i \geq 0 \end{aligned} \qquad \begin{aligned} \min \ & \text{risk measure}, \\ \text{s.t.} \ & \sum \mu_i x_i \geq r_0 \\ & \sum x_i = 1, \\ & x_i \geq 0 \end{aligned}$$

Review of Risk Measures

- Variance
- Value-at-Risk (VaR): Let $F(\cdot)$ be the distribution function of the random loss X. For a given $\alpha \in (0,1)$, VaR of X at level α is defined as

$$\text{VaR}_\alpha(X) := \inf\{x \mid F(x) \geq \alpha\} = F^{-1}(\alpha).$$

- Conditional Value-at-Risk (CVaR): The α-tail distribution function of X is defined as

$$F_{\alpha,X}(x) := \begin{cases} 0, & \text{for } x < \text{VaR}_\alpha(X), \\ \frac{F_X(x)-\alpha}{1-\alpha}, & \text{for } x \geq \text{VaR}_\alpha(X). \end{cases}$$

$$\begin{aligned} \text{CVaR}_\alpha(X) &:= \text{mean of the } \alpha\text{-tail distribution of } X \\ &= \int_{-\infty}^{\infty} x dF_{\alpha,X}(x). \end{aligned}$$

Basel 2.5 Accord, July 2009

Financial Crisis of 2007/2008

- capital requirements for market risk

$$
\begin{aligned}
c_t &= \max\left\{ \mathrm{VaR}_{\alpha,t-1}, \frac{k}{60}\sum_{s=1}^{60} \mathrm{VaR}_{\alpha,t-s} \right\} \\
&\quad + \max\left\{ \mathrm{sVaR}_{\alpha,t-1}, \frac{\ell}{60}\sum_{s=1}^{60} \mathrm{sVaR}_{\alpha,t-s} \right\}
\end{aligned}
$$

- $\mathrm{sVaR}_{\alpha,t-s}$ is called the *stressed* VaR on day $t-s$ at confidence level $\alpha = 99\%$, which is calculated under the scenario that the financial market is under significant stress such as the one that happened during the period from 2007 to 2008.

Basel 3 Accord, May 2012

- Uses CVaR (or, equivalently, ES) instead of VaR
- The capital requirement for a group of trading desks that share similar major risk factors, such as equity, credit, interest rate, and currency, is defined as the CVaR of the loss that may be incurred by the group of trading desks;
- The CVaR should be calculated under stressed scenarios rather than under current market conditions.
- capital requirements for market risk

$$c_t = \max\left\{ \text{sCVaR}_{\alpha,t-1}, \frac{\ell}{60}\sum_{s=1}^{60} \text{sCVaR}_{\alpha,t-s} \right\},$$

- $\text{sCVaR}_{\alpha,t-s}$ is the *stressed* CVaR at level α calculated on day $t-s$.

Correlation Matrices

A correlation matrix satisfies

$$X = X^\top,\ X_{ii} = 1,\ i = 1, \ldots, n,\ X \succeq 0.$$

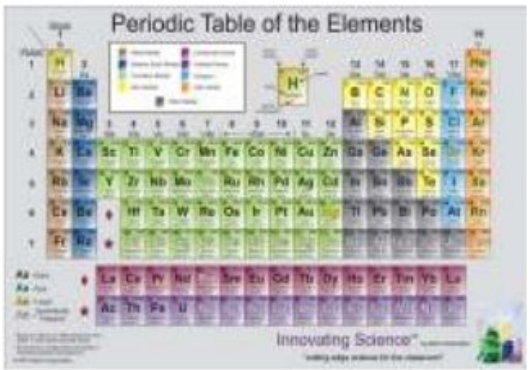
Example: (low-rank) nearest correlation matrix estimation

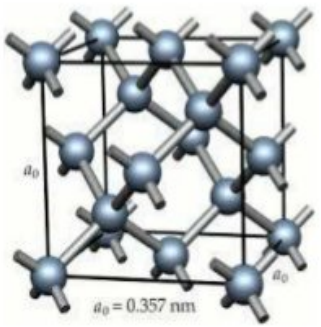
$$\begin{aligned} &\min \frac{1}{2} \|X - C\|_F^2, \\ &\text{s.t. } X = X^\top,\ X_{ii} = 1,\ i = 1, \ldots, n,\ X \succeq 0 \end{aligned}$$

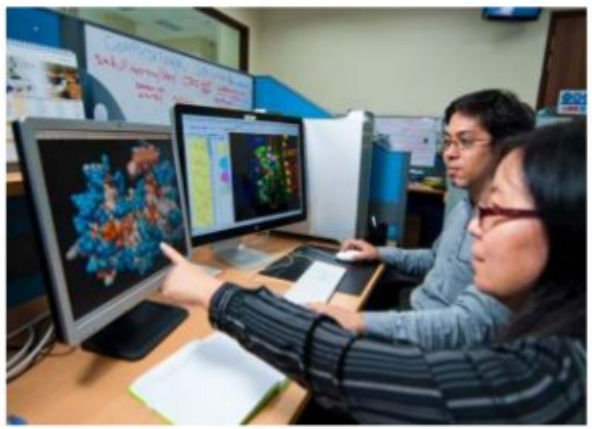
- objective fun.: $\|W \odot (X - C)\|_F^2$
- lower and upper bounds
- rank constraints $\operatorname{rank}(X) \leq r$

Computational Materials Simulation

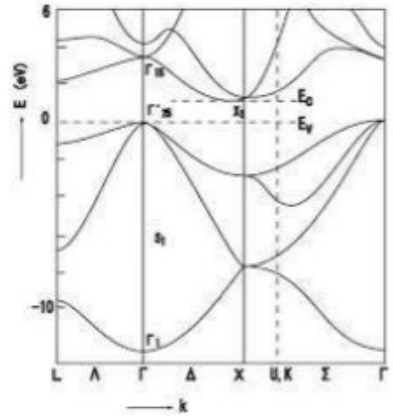
Numerical simulation of material on atomic and molecular scale (Thanks: Chao Yang)



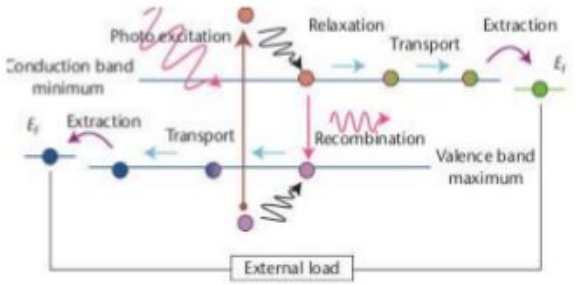


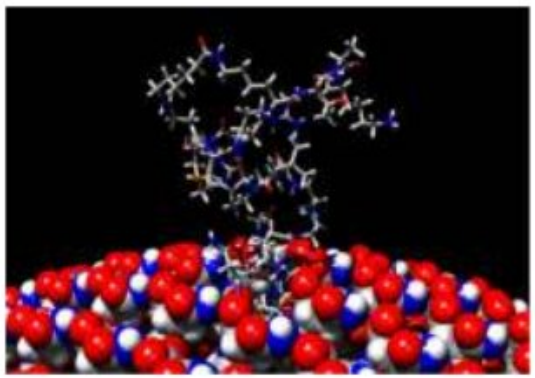


What to simulate



- Energy
  - Ground state vs excited states
  - Binding energy (cohesive, adsorption etc.)
  - Band structure
- Geometry
  - Most stable configuration
- Dynamics
- Reaction rates
- Charge transfer and electron transport
- Mechanical properties (e.g. elasticity)
- Optical properties





Electronic Structure Calculation

- N particle Schrodinger equation: Physics of material systems — atomic and molecular properties, almost correct (nonrelativistic) phyiscs is quantum mechanics
- Main goal: Given atomic positions $\{R_\alpha\}_{\alpha=1}^{M}$, compute the ground state electron energy $E_e(\{R_\alpha\})$.
- Ground state electron wavefunction $\Psi_e(r_1, \ldots, r_N; \{R_\alpha\})$:

$$H\Psi_e = \left(-\frac{1}{2}\sum_{i=1}^{N} \Delta_i - \sum_{\alpha=1}^{M}\sum_{j=1}^{N} \frac{Z_\alpha}{|r_i - R_\alpha|} + \frac{1}{2}\sum_{i,j=1, i\neq j}^{N} \frac{1}{|r_i - r_j|} \right) \Psi_e$$
$$= E_e(\{R_\alpha\})\Psi_e$$

- Curse of dimensionality: Computational work goes as 10^{3N}, where N is the number of electrons

Kohn-Sham Formulation

- Replace many-particle wavefunctions, Ψ_i, with single particle wavefunction, ψ_i
- Minimize Kohn-Sham total energy

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{i=1}^{n_e}\int_\Omega |\nabla\psi_i|^2 + \int_\Omega V_{ion}(\rho) + \frac{1}{2}\int_\Omega \frac{\rho(r)\rho(r')}{|r-r'|}drdr' + E_{xc}(\rho) \\ \text{s.t.} \quad & \rho(r) = \sum_{i=1}^{n_e}|\psi_i(r)|^2, \int_\Omega \psi_i\psi_j = \delta_{i,j} \end{aligned}$$

  Exchange-correlation term: E_{xc} contains quantum mechanical contribution, plus, part of K.E. not converged by first term when using single-particle wavefunctions
- Discretized Kohn-Sham Formulation

$$\min_{X^*X=I} E_{KS}(X)$$

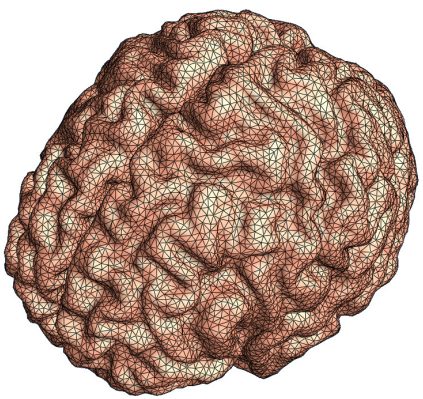
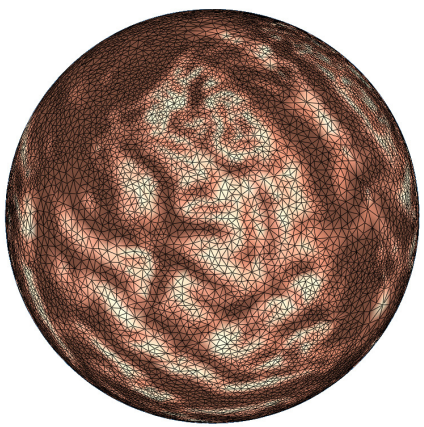
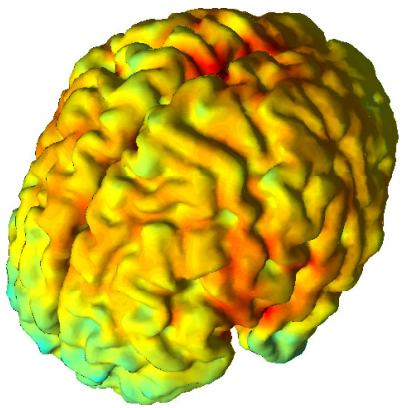
Optimization on Manifold

Consider $X \in \mathbb{C}^{n \times p}$ and

$$\min F(X), \quad \text{subject to} \quad X^\top X = I$$

Why is the problem interesting?

- Stiefel Manifold, Grassmanian Manifold
- it is expensive to keep constraints feasible;
- non-convex, leading to possibly local minima.
- it has many applications;

Course goals and topics

goals

1. recognize/formulate problems as convex optimization problems
2. develop code for problems of moderate size
3. characterize optimal solution, give limits of performance, etc.

topics

1. convex sets, functions, optimization problems
2. examples and applications
3. algorithms

Nonlinear optimization

traditional techniques for general nonconvex problems involve compromises

local optimization methods (nonlinear programming)

- find a point that minimizes f_0 among feasible points near it
- fast, can handle large problems
- require initial guess
- provide no information about distance to (global) optimum

global optimization methods

- find the (global) solution
- worst-case complexity grows exponentially with problem size

these algorithms are often based on solving convex subproblems

Brief history of convex optimization

theory (convex analysis): ca1900-1970

algorithms

- 1947: simplex algorithm for linear programming (Dantzig)
- 1960s: early interior-point methods (Fiacco & McCormick, Dikin, ...)
- 1970s: ellipsoid method and other subgradient methods
- 1980s: polynomial-time interior-point methods for linear programming (Karmarkar 1984)
- late 1980s-now: polynomial-time interior-point methods for nonlinear convex optimization (Nesterov & Nemirovski 1994)

applications

- before 1990: mostly in operations research; few in engineering
- since 1990: many new applications in engineering (control, signal processing, communications, circuit design, ...); new problem classes (semidefinite and second-order cone programming, robust optimization)