Introduction: Examples on Convex Optimization Problems

Instructor: Jin Zhang

Department of Mathematics
Southern University of Science and Technology

Fall 2023

Contents

Part I: Nonsmooth Setting

Consider that if $\bar{x} \in \mathbb{R}^n$ is a minimizer of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$, that is to say,

$$f(\bar{x}) = \min_{x \in \mathbb{R}^n} f(x),$$

then we have

$$\nabla f(\bar{x}) = 0.$$

Moreover if f is convex, every solution of the equation $\nabla f(x) = 0$ with respect to $x \in \mathbb{R}^n$ is a minimizer of f.

But what if f is nonsmooth?
For example consider the absolute value function $f(x) = |x|, (x \in \mathbb{R})$.
Notice that $\bar{x} = 0$ is a minimizer of the absolute value function but at the same time $\nabla f(x)$ does not exist.

Part II: Examples

- Support Vector Machines
- Least Squares Problems
- Linear Programming
- Optimal transport
- Regression and Formulation
- ...

Example 1 : SVM (Support Vector Machines)

Consider the following instance of empirical risk minimization

$$\min_{x \in \mathbb{R}^m} \frac{\lambda}{2} \|x\|_2^2 + \frac{1}{n} \sum_{i=1}^{n} \max\left\{0, 1 - b_i(a_i^T x)\right\}$$

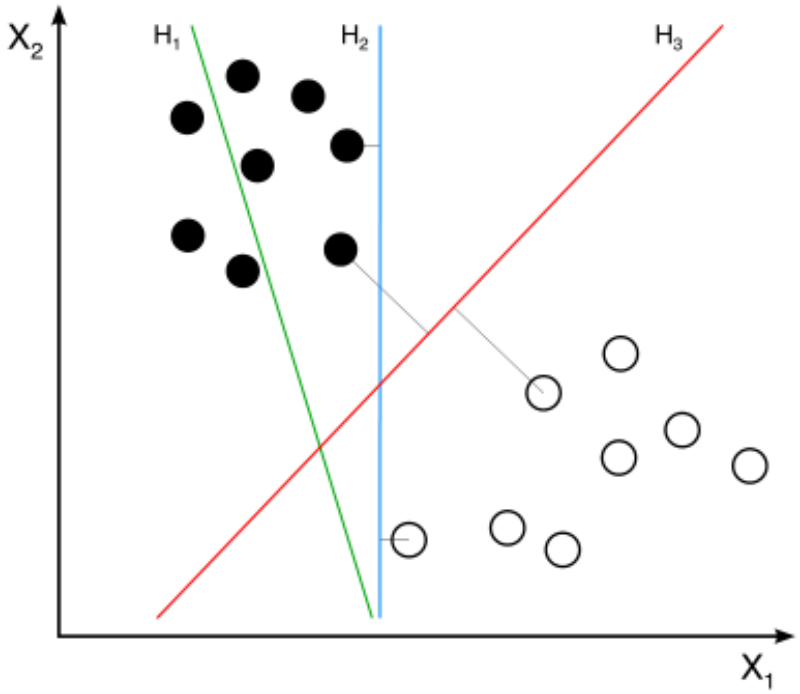
where

- the first term $\frac{\lambda}{2} \|\cdot\|_2^2$ for a parameter $\lambda > 0$ is a regularizer.
- the second term is a hinge loss.
- a_i is a feature vector associated to a label $b_i = \pm 1$.
- $a_i^T x$ is a linear predictor.

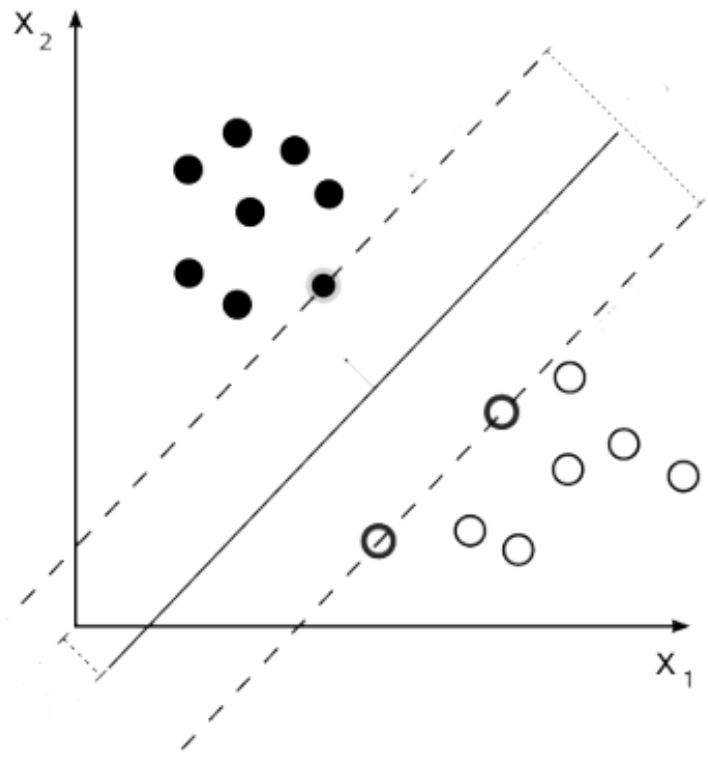
Interpretation:

- For fixed $x \in \mathbb{R}^m$, the sign of b_i determines the side of the hyperplane $H_x = \left\{ y \in \mathbb{R}^m \,\middle|\, \langle x, y \rangle = 0 \right\}$.
- Right side: $a_i \in H_x + b_i \frac{x}{\|x\|^2}$.
- Wrong side: $a_i \notin H_x + b_i \frac{x}{\|x\|^2}$.
- The hinge loss does not penalise x if a_i is on the right side of H_x.
- If a_i is too far on the wrong side of H_x, then it is penalised by the amount $1 - b_i(a_i^T x)$.
- The regularizer controls the length of x.
- The length of x controls the margin of the SVM.

Linear support vector machines can thus be used to find a separating hyperplane to samples a_i spurning from two different clusters characterised by $b_i = \pm 1$.



(a) The hyperplane H_1 does not separate the two classes. H_2 does, but only with a small margin. The hyperplane H_3 separates them with the optimum margin.



(b) The margin of the SVM is the distance $2/\|x\|$ between the dashed lines.

Example 2: 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

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

Example 3: Linear programming

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

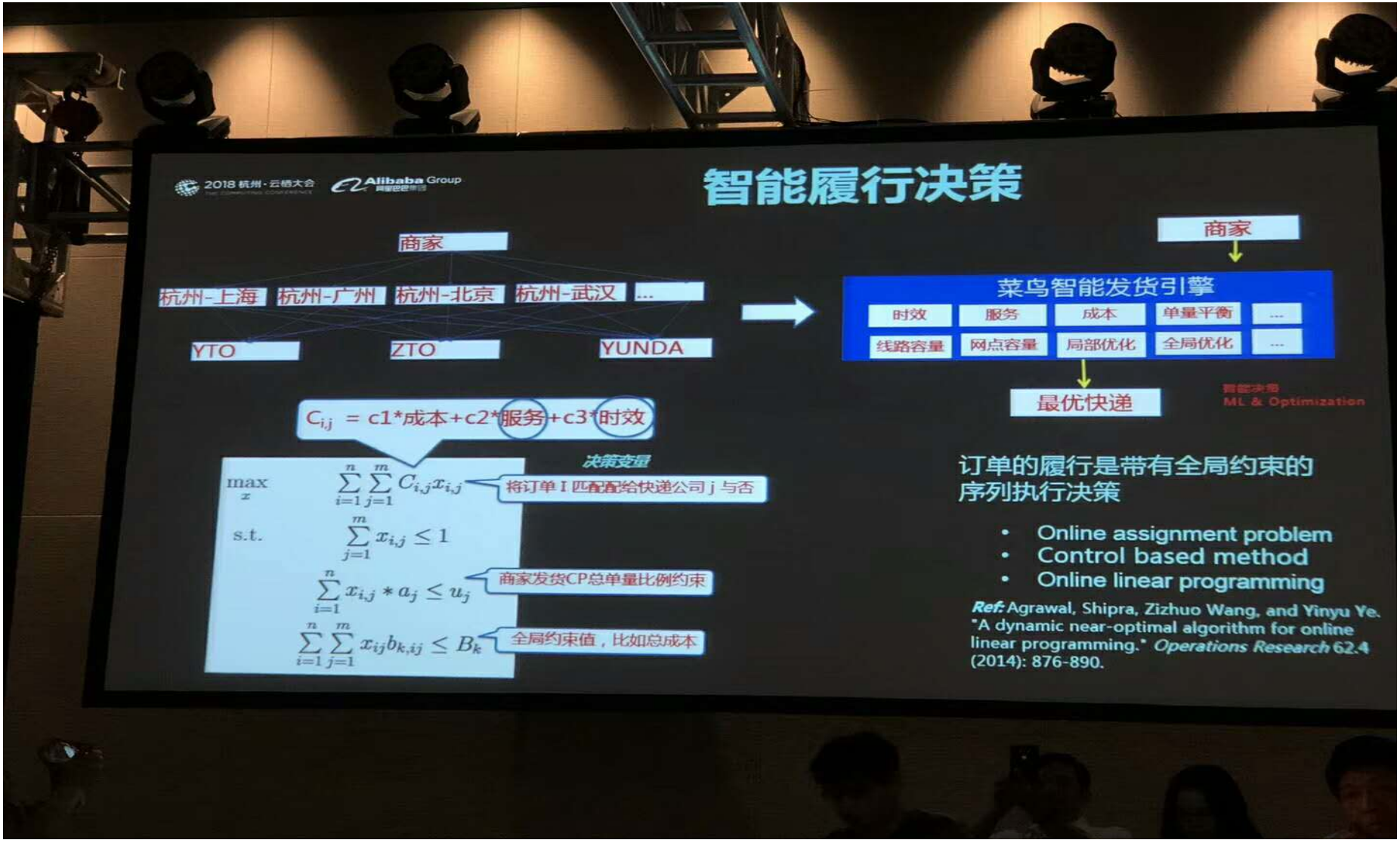
solving linear programs

- 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

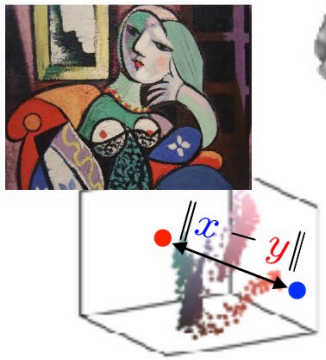
- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving ℓ_1- or ℓ_∞- norms, piecewise-linear functions)

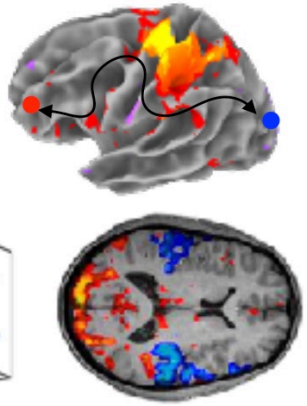
An example of linear programming: 菜鸟

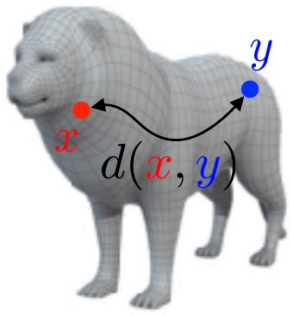


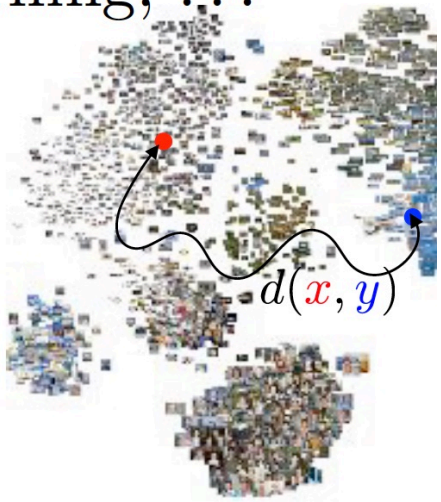
Example 4: Optimal transport

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










Monge


Kantorovich


Koopmans

Nobel '75


Dantzig


Brenier


Otto


McCann


Villani

Fields '10


Figalli

Fields'18

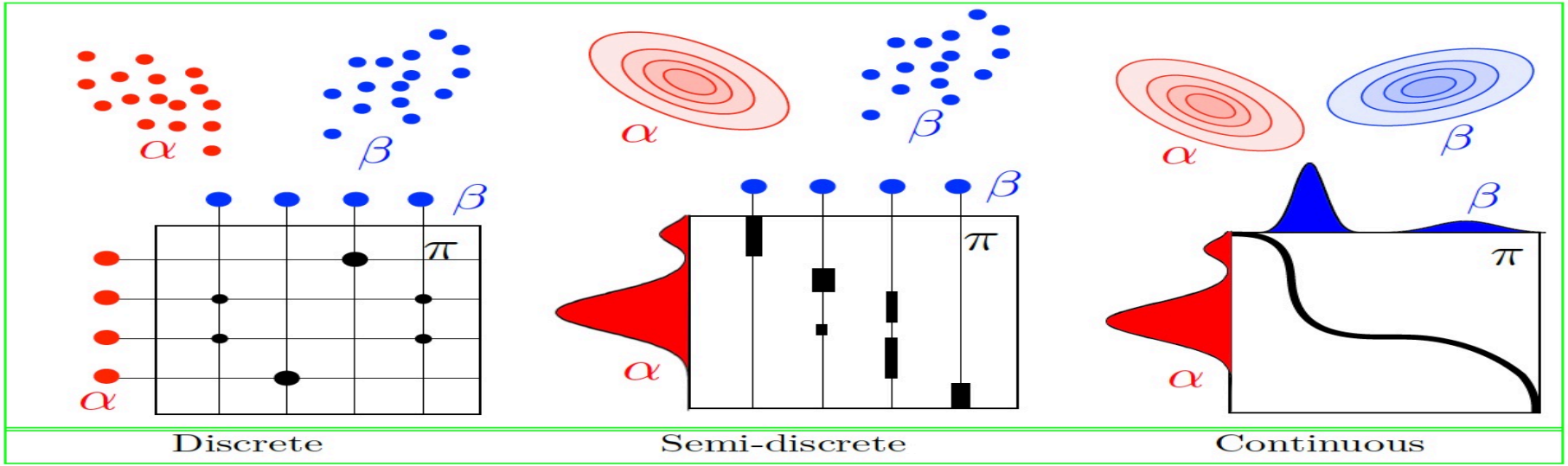
Optimal transport: LP

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

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

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

$$\pi \geq 0$$



Part III: Optimization in Machine Learning

Why Optimization in Machine Learning?

Many problems in ML can be written as

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

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

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

- 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

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

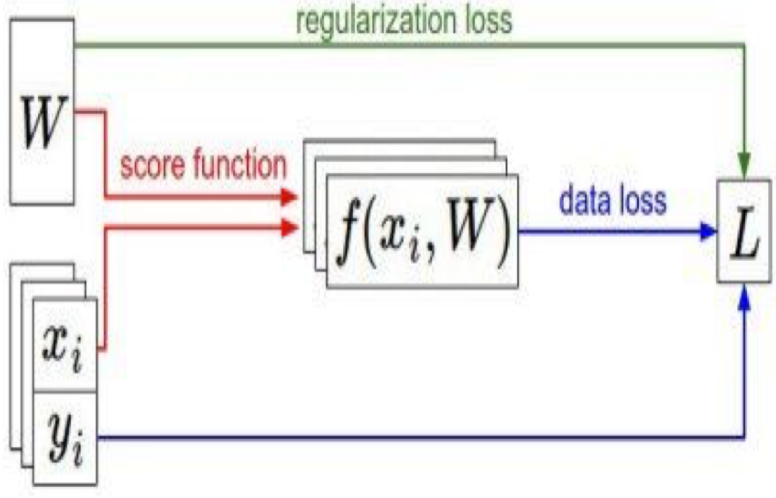
Softmax

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

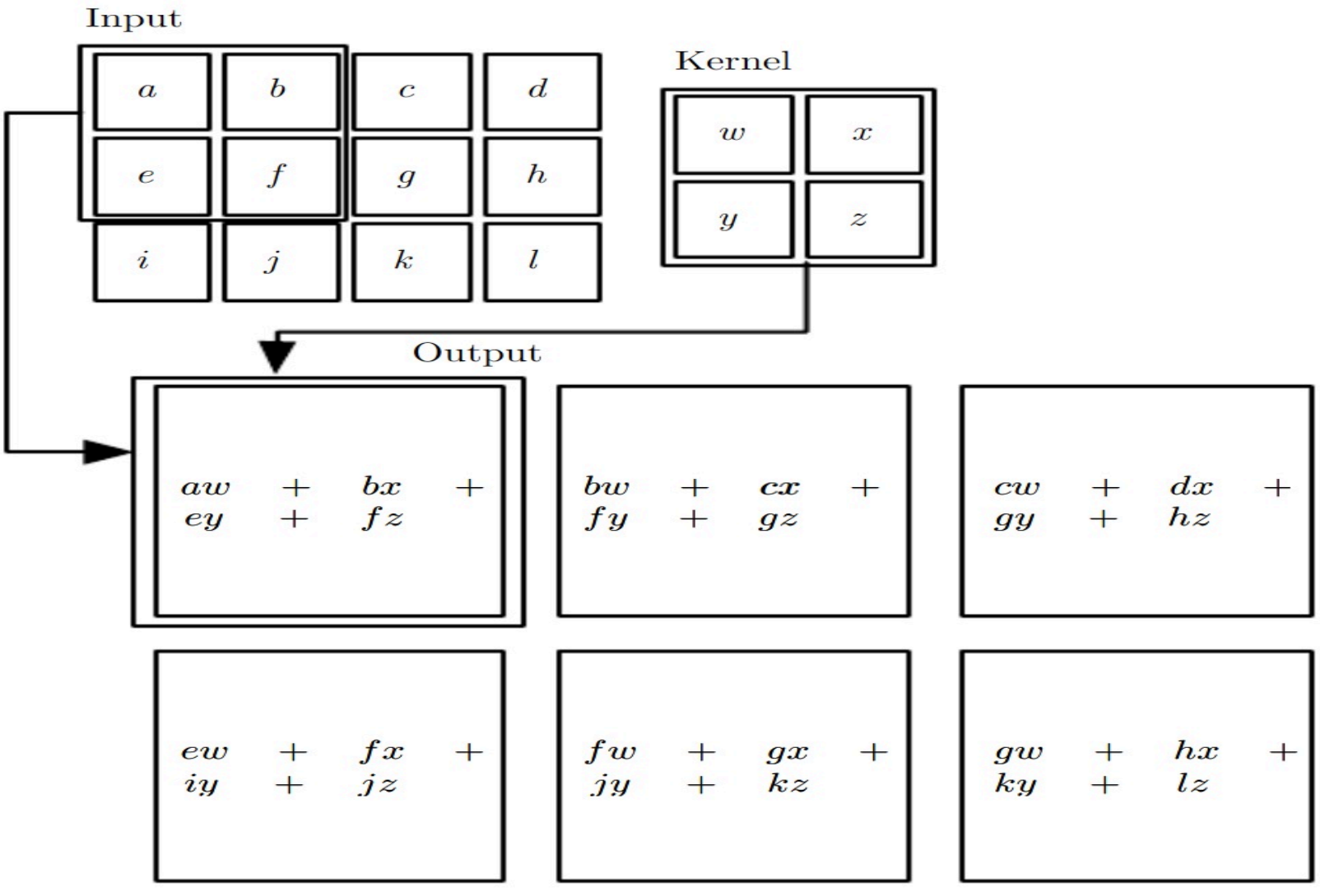
SVM

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

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



convolution operator



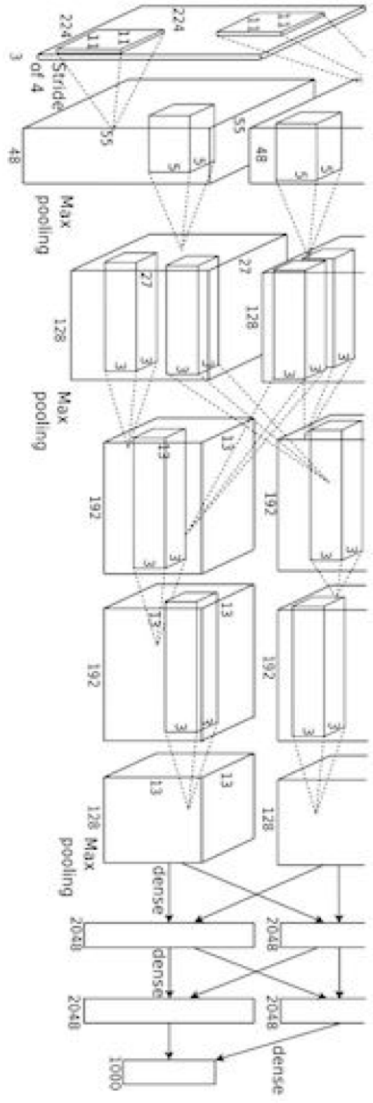
Loss functions in neural network

Convolutional Network
(AlexNet)

input image

weights

loss



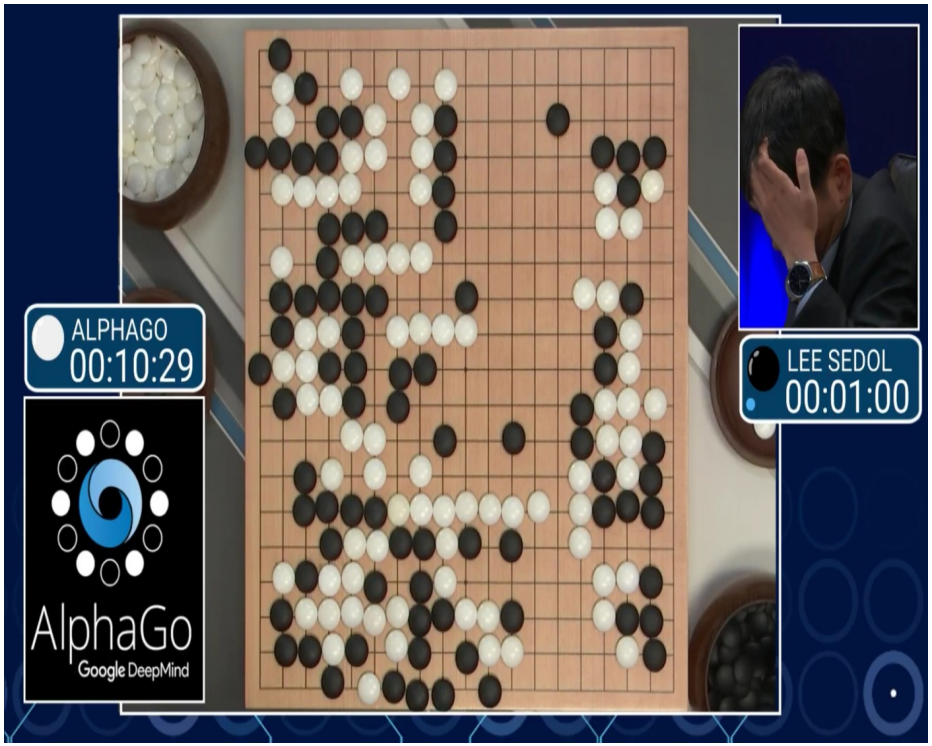
Optimization algorithms in Deep learning

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`

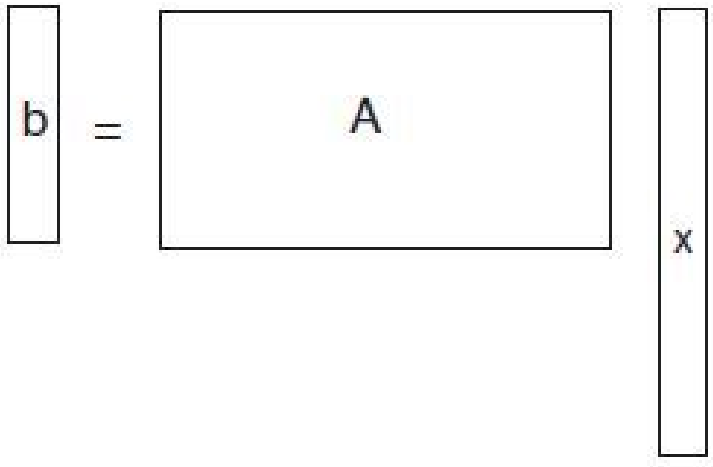
Reinforcement Learning

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



压缩感知(Compressive Sensing)

压缩感知(Compressive Sensing)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

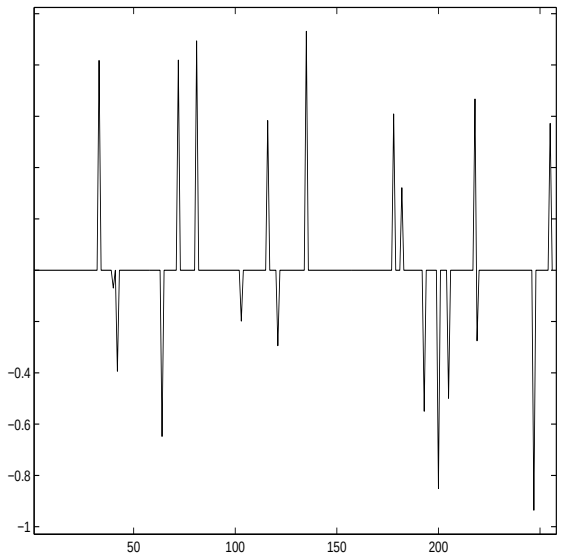
$$Ax = b$$

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

Compressive Sensing

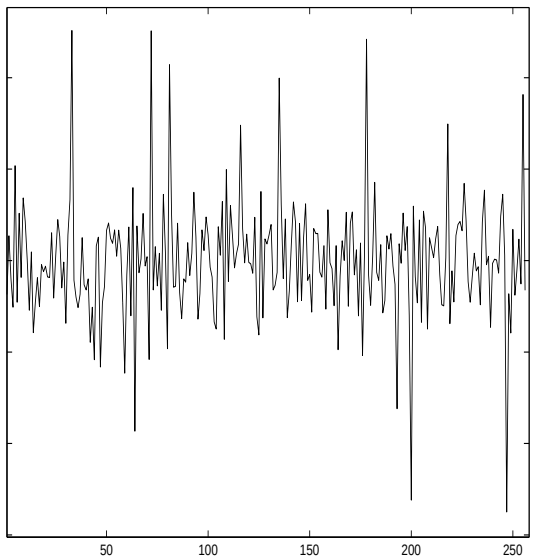
Find the sparest solution

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



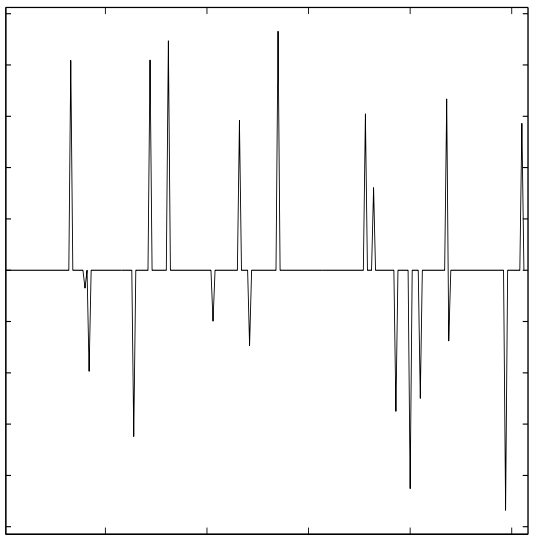
$$\begin{cases} \min\limits_{x} \|x\|_0 \\ \text{s.t. } Ax = b \end{cases}$$

(a) ℓ_0-minimization



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

(b) ℓ_2-minimization



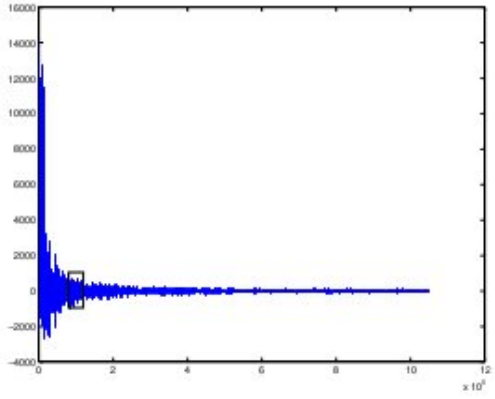
$$\begin{cases} \min\limits_{x} \|x\|_1 \\ \text{s.t. } Ax = b \end{cases}$$

(c) ℓ_1-minimization

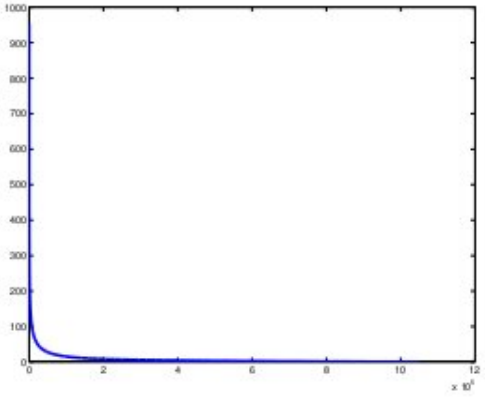
Wavelets and Images

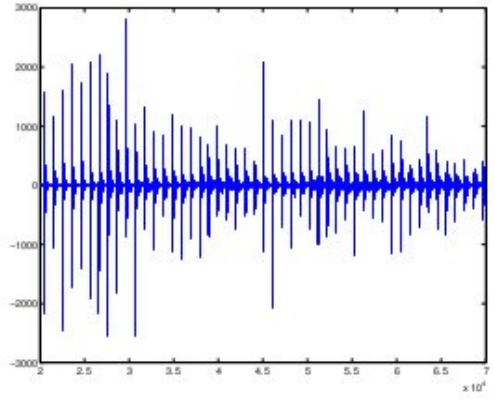
1 megapixel image

wavelet coeffs

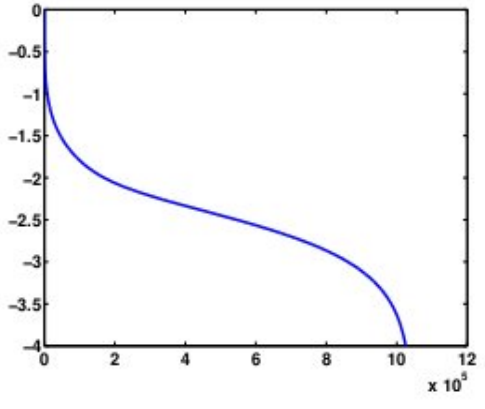


(sorted)





zoom in

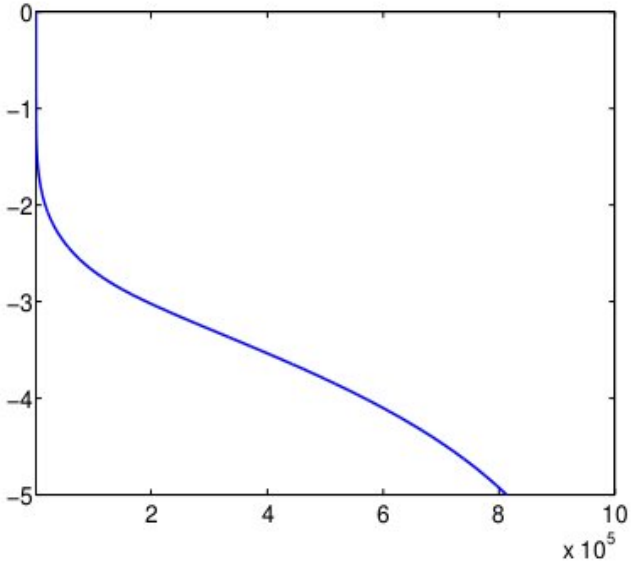


($\log_{10}$ sorted)

Wavelet Approximation

1 megapixel image

25k term approx



B-term approx error

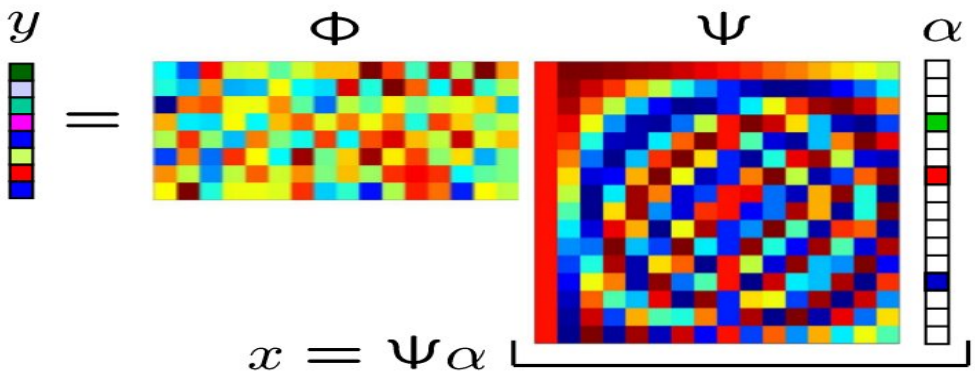
- Within 2 digits (in MSE) with $\approx 2.5\%$ of coeffs
- Original image = f, K-term approximation = f_K

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

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])

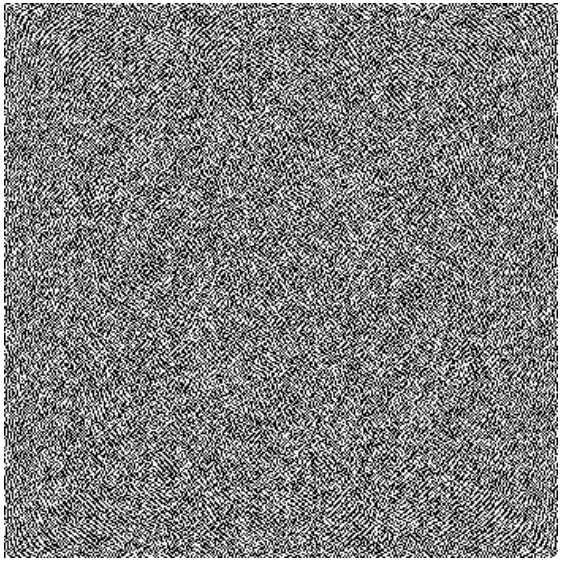
If x is a k-sparse and A satisfies $\delta_{2k} + \delta_{3k} < 1$, then x is the unique ℓ_1 minimizer.

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

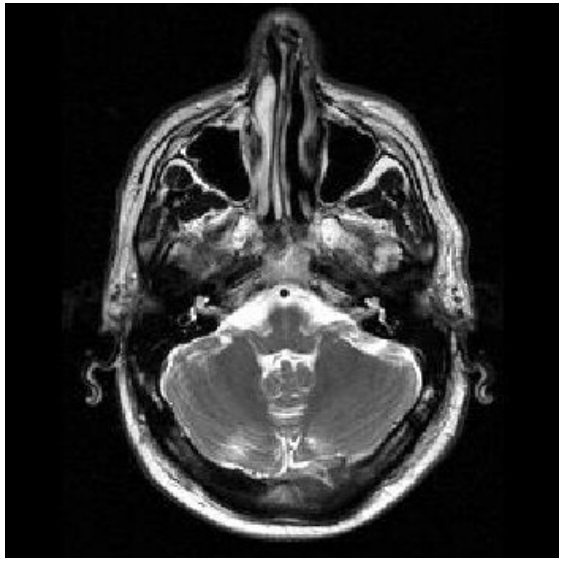
Magnetic Resonance Imaging and Phase Retrieval

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.

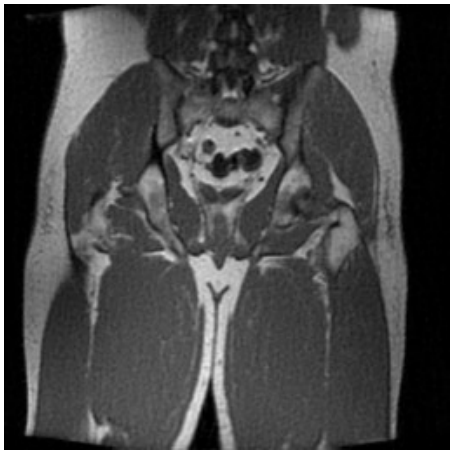
(a) full sampling

(b) 39% sampling, SNR=32.2

MRI: Magnetic Resonance Imaging

(a) full sampling

(b) 39% sampling, SNR=32.2

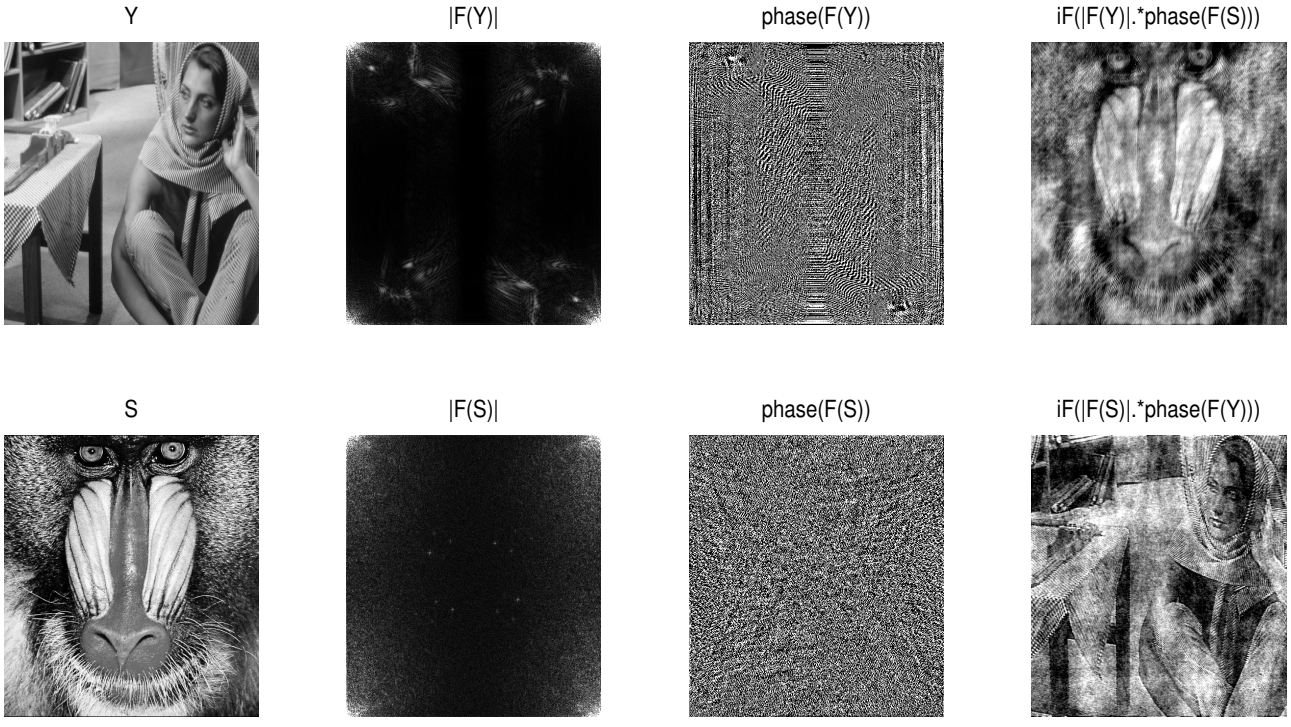


(c) 22% sampling, SNR=21.4

(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:

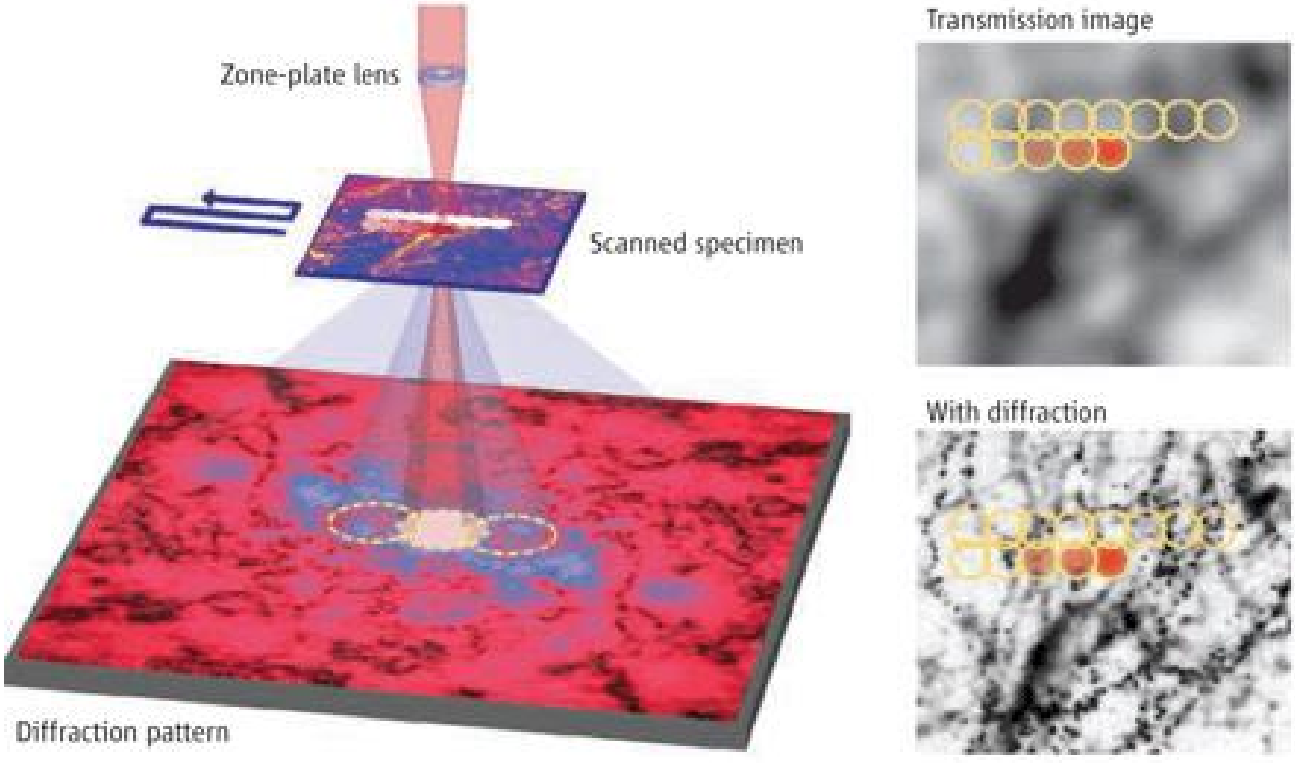
$$S := \{x(r) \mid x(r) = 0 \text{ for } r \notin D\}$$

  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 $X = xx^*$

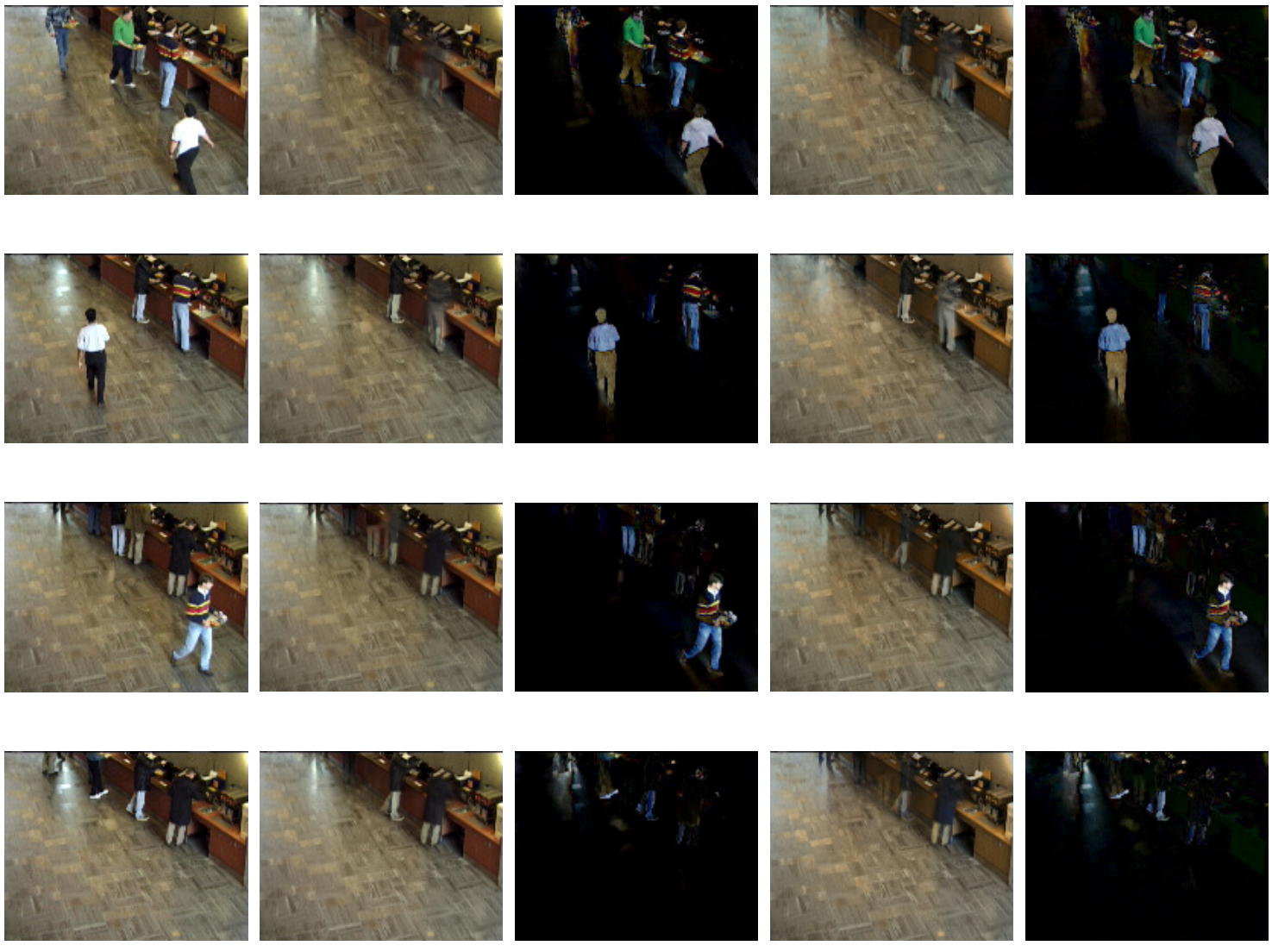
$$\begin{aligned} \min_{X \in S_n} \quad & Tr(X) \\ s.t. \quad & Tr(a_i a_i^* X) = b_i^2, i = 1, \ldots, m, \\ & X \succeq 0 \end{aligned}$$

- Exact recovery conditions

Video separation

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

Part IV: Netflix (Recommendation System)

Movie Rating

	Alice	Bob	Chris	David
X-Men	3	5	4	4
Yellow Rock	2	3	?	3
Zoolander	4	4	4	?

- 用户：Alice, Bob, Chris, David
- 电影：X-Men, Yellow Rock, Zoolander
- 数字表示某位用户对某部电影的评分。
  例如 Alice 对 X-Men 这部电影的打分是 3。
- “?” 表示出于某些原因这个人没有对某部电影进行打分。
  例如 David 对 Zoolander 这部电影没有打分。
- 猜出 “?” 所对应的数值，就是推荐系统 (Recommendation Systems) 要做的事情。

如果生产产品 (电影，歌曲，视频，服装等) 的公司猜到了某用户对某产品的打分，公司就可以知道此用户对该产品的喜好程度 (虽然他并没有进行购买)。如果评分好，推荐系统就可以把该产品推荐给他，他就很有可能购买该产品，从而提供公司的盈利。

杉数科技 Cardinal Operations

Recommendation systems

Search

Recommendations

Items

Products, web sites, blogs, news items, …

Examples:

amazon.com.

PANDORA

StumbleUpon

del.icio.us

NETFLIX

movielens
helping you find the *right* movies

last·fm
the social music revolution

Google News

YouTube

XBOX LIVE

The Netflix Prize

- Competition
  - 2700+ teams
  - $1 million prize for 10% improvement on Cinematch

Netflix 公司在 2008 年发起了一个竞赛，获奖者可以得到 100 万美金的奖励。获奖的条件是：

- 在 2700 多个参赛队里拿到第一名;
- 比 Netflix 公司原有系统 Cinematch 跑出的结果准确率高 10%.

Training data:

- 6 years (2000-2005) of data:
  100 million ratings, 480,000users, 17,770 movies.

即现在有一个 48 万行，1 万 7 千列的矩阵，在这个矩阵中，我们知道了 1 亿个元素的值 (仅占矩阵元素个数的百分之一)。参赛者的任务是把矩阵中未知的元素数值 (用户对电影的打分值) 猜测出来。

如何衡量打分结果?

Test data

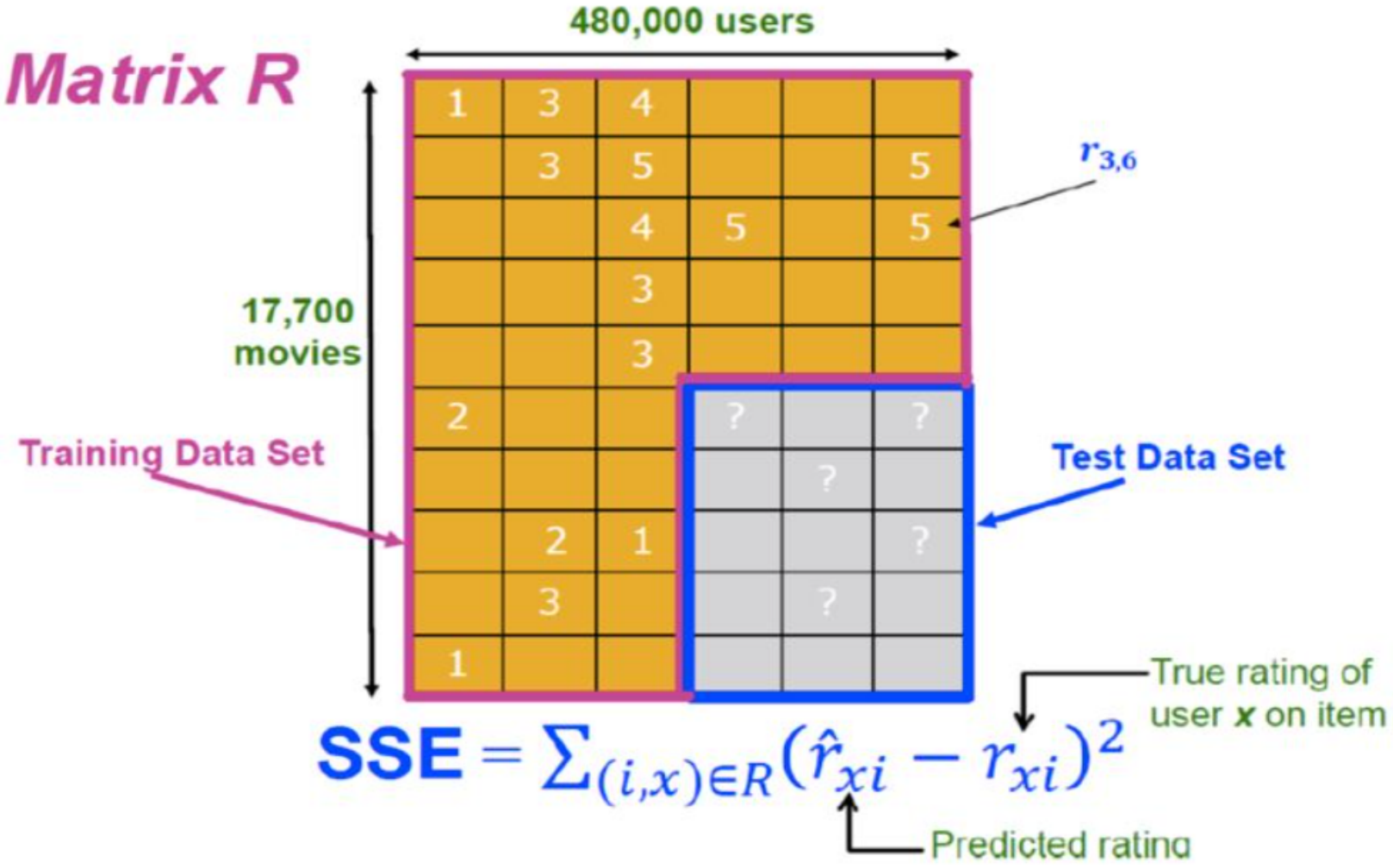
- Last few ratings of each user (2.8 million);
- Evaluation criterion:

$$error = \sqrt{\sum_{xi} (r_{xi} - r^*_{xi})^2},$$

  where r_{xi} and r^*_{xi} are the predicted and true rating of the x-th user on the i-th movie.
- Netflix Cinematch: $error = 0.9514$

参赛者的 $error$ 越小表示参赛者猜的越准，表现越好。

Netflix: evaluation

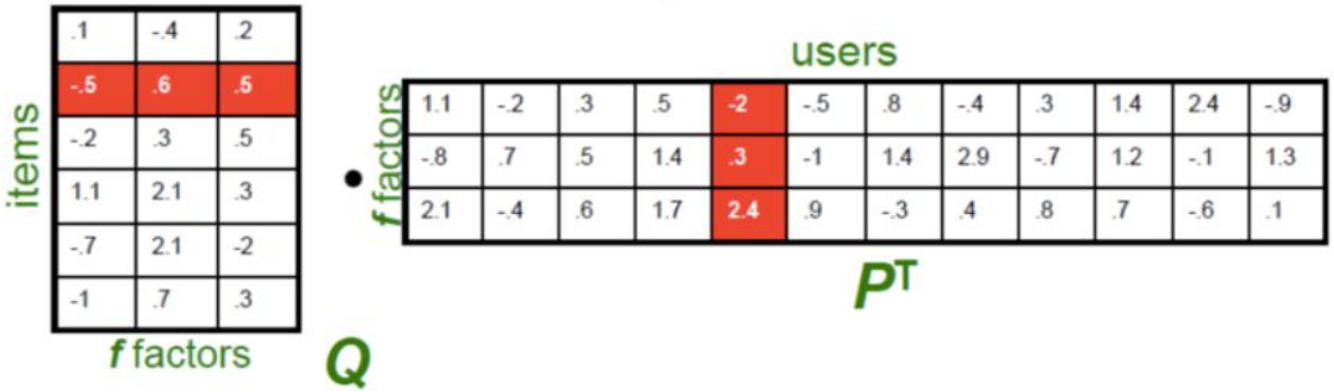


问题建模

Latent factor models: ratings as products of factors

- How to estimate the missing rating of user x for item i?

$$\hat{r}_{xi} = q_i \cdot p_x^T = \sum_f q_{if} p_{xf},$$



- 电影有恐怖, 科幻, 爱情等分类; 例如成龙电影一般既是武打片, 又是喜剧.
- 左图的中 items 每一行表示每个类型所占的权重。例如标红的 0.5,0.6,0.5 可以表示一部电影在喜剧，爱情，故事片三个元素上所占因子。
- 不同用户对电影类型的偏好也不一样。例如年轻人可能喜欢科幻的、浪漫的电影，对这类电影打分比较高。老一辈人可能更喜欢革命电影，故事片。在右图中，标红的这一列，表示了不同的用户对不同类型电影的喜好程度。
- 左图第 i 行表示第 i 部电影在不同类型的比重，右图第列 x 对应第 x 位用户对不同类型电影的喜好程度。二者的内积是该用户对该电影的打分情况

Matrix Rank Minimization

- matrix completion problem:

$$\min \quad \text{rank}\,(R)\,, \quad \text{s.t. } r_{xi} = r^*_{xi},\ (i,x) \in \text{ training data}$$

- matrix rank minimization problem:

$$\min \quad \text{rank}\,(R)\,, \quad \text{s.t. } \mathcal{A}\,(R) = r = (r_{xi})_{(i,x)\in \text{ training data}}$$

- nuclear norm minimization:

$$\min \quad \|R\|_*\,, \quad \text{s.t. } \mathcal{A}\,(R) = r$$

where $\|R\|_* = \sum_i \sigma_i$ and σ_i is the i-th singular value of matrix R

故事继续

The last 30 Days

- BellKor
  - Lead all teams
  - Continue to get small improvements in their scores
  - Realize they are in direct competition with team Ensemble
- Ensemble
  - Group of other teams on leaderboard forms a new team
  - Relies on combining their models
  - Quickly also get a qualifying score over 10%.

在比赛还剩 30 天结束时，一个叫 BellKor 的队伍在所有队伍中排第一。但比赛规则是只有第一名可以得到 100 万奖金，所以第二名和最后一名没有任何差别。排名靠前的其他一些队伍联合了起来，把各自的模型整合在一起，组成了一个新的队伍:Ensemble。Ensemble 将模型整合后，确实发现分数提高了 10%。

24 hours from the deadline

- Submission limited to 1 a day: Only 1 final submission could be made in the last 24 hours.
- 24 hours before deadline...
  - BellKor team member in Austria notices (by chance) that Ensemble posts a score that is slightly better than BellKor's
- Final submissions:
  - BellKor submits a little early (on purpose), 40 mins before deadline.
  - Ensemble submits their final entry 20 mins later.

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update

Leaderboard

Showing Test Score. Click here to show quiz score

Display top 20 **leaders.**

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace_	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11