Optimization in Machine Learning: Lecture 1

Tutorial

by Xiaolin Huang xiaolinhuang@sjtu.edu.cn SEIEE 2-429

Institute of Image Processing and Pattern Recognition

http://www.pami.sjtu.edu.cn/

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录 Contents

1 What is Optimization

2 History of Optimization

3 Optimization in Machine Learning

4 Course Information

目录 Contents

1 What is Optimization

2 History of Optimization

3 Optimization in Machine Learning

4 Course Information

Definition of Optimization

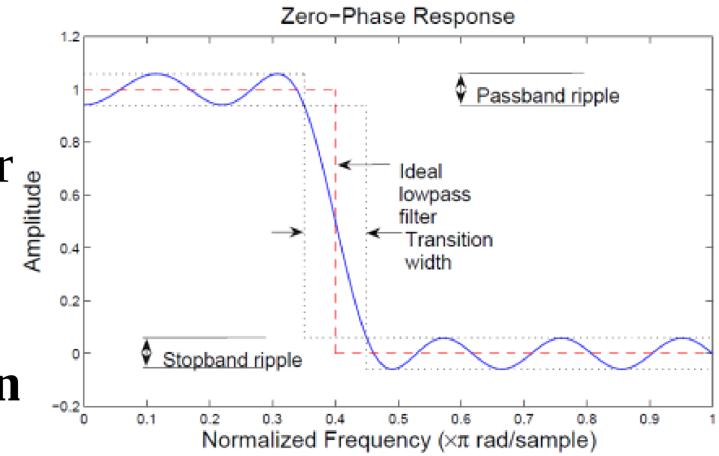
- **Optimization** includes finding "best available" values of some objective function given a defined domain (or input), including a variety of different types of objective functions and different types of domains[wikipedia]

- *"Since 1990 many applications have been discovered in areas such as* ***automatic control systems****,* ***estimation and signal processing****,* ***communications and networks****,* ***electronic circuit design****,* ***data analysis and modeling****,* ***statistics****, and* ***finance****." —— S. Boyd and L. Vendenberghe*

Applications of Optimization

Automatic Control Systems

- **Optimal Control** deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved.
- **Stability condition** of continuous-time linear systems could be modeled as an linear matrix inequality, a convex constraint.
- **Filter Design** is to find a discrete FIR filter by minimizing the integral of the square of the error to an ideal frequency function of the filter
- **Model Predictive Control, Network Allocation**



https://www.mathworks.com/

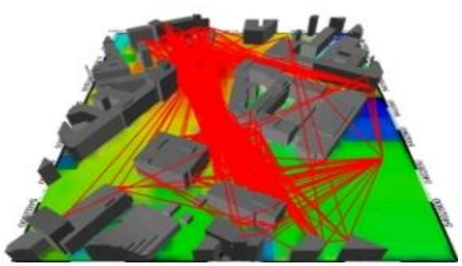
Applications of Optimization

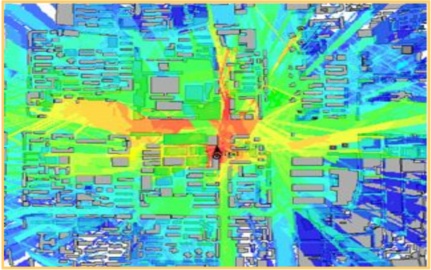
Signal Processing

- **Compressive/Compressed Sensing**

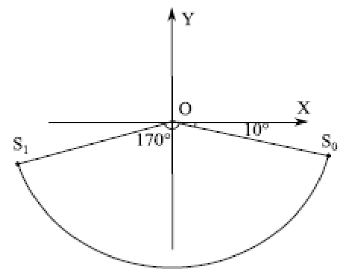
  By minimizing the l1 regularization term, one could recover sparse signals with relatively fewer measurements

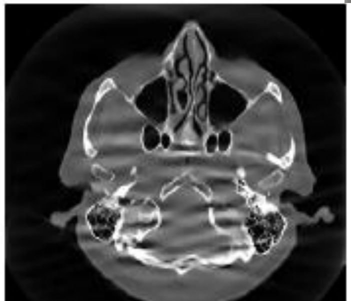
- Parameters Setting
  - Antenna/Cell/District/City

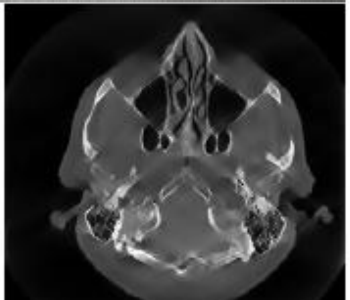




J. Mattingley and S. Boyd, Real-time Convex Optimization in Signal Processing



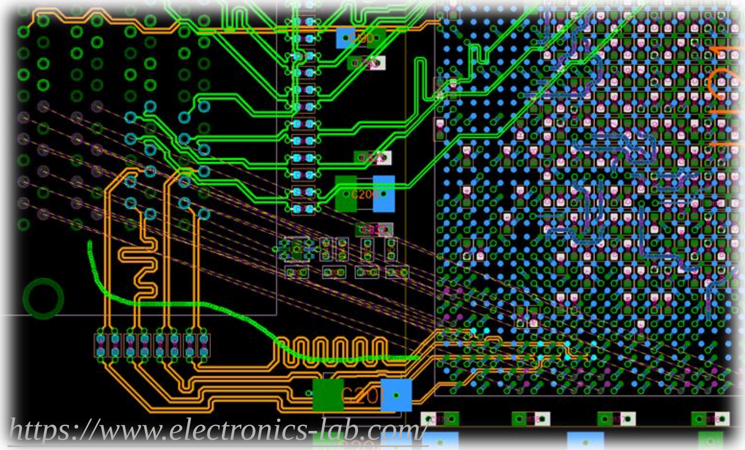




Y. Huang, O. Taubmann, X. Huang, et al. Scale-Space Anisotropic Total Variation for Limited Angle Tomography

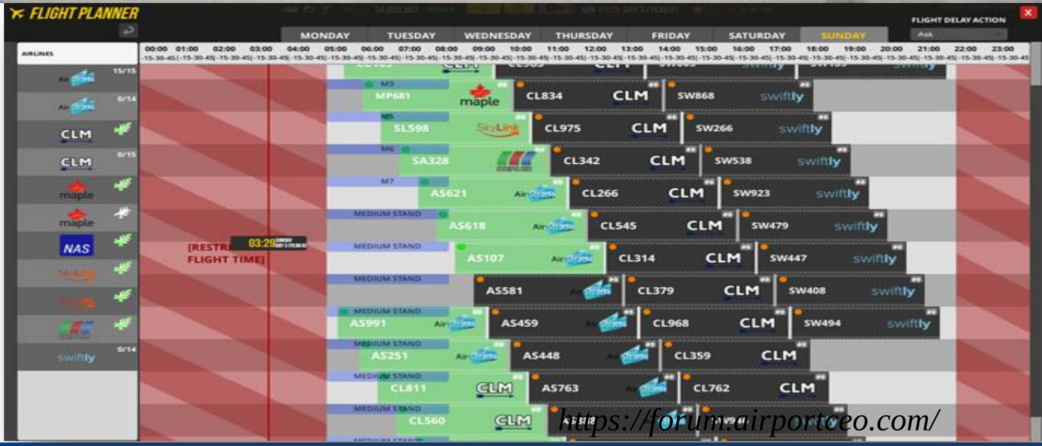
Applications of Optimization

- **Electronic Circuit Design**
  - **Device sizing (EDA)**
- **Finance**
  - **Prediction**
  - **Portfolio optimization**
  - **Risk control**
- **Route Planning**
- **Scheduling**



Applications of Optimization

- **Route Planning**
- **Scheduling**



https://forum.airportceo.com/

Other Words

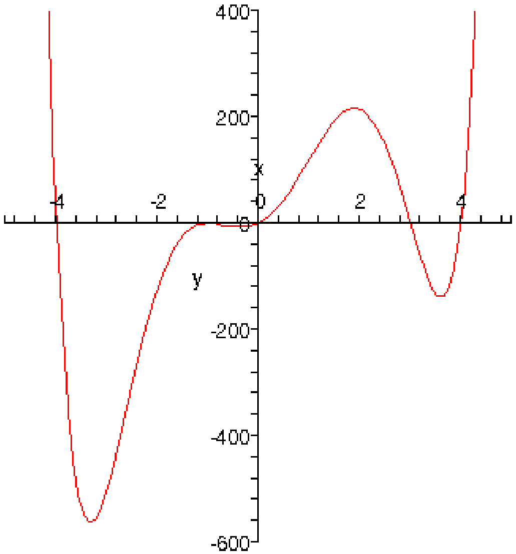
- **Optimization** includes finding "best available" values of some objective function given a defined domain (or input), including a variety of different types of objective functions and different types of domains[wikipedia]
- **Operations Research/Operational Research**
- **Mathematical Programming**

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\
\text{s.t.} \quad & f_i(x) \le 0, i = 1, \ldots, m, \\
& h_i(\mathbf{x}) = 0, i = 1, \ldots, p.
\end{aligned}
$$

- 优化/运筹学/数学规划
- 优选法/统筹法

Optimization Problem

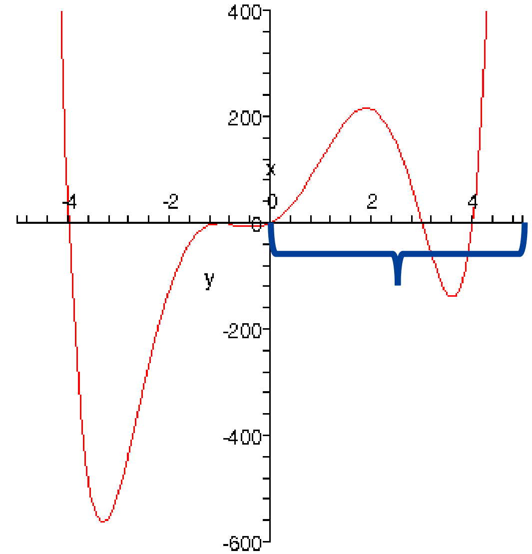
$$\min_{x} \quad f_0(x)$$
$$\text{s.t.} \quad f_i(x) \le 0, i = 1, \ldots, m$$
$$h_i(x) = 0, i = 1, \ldots, p$$



- $f_0(x)$: objective function
  - loss/cost function
  - (minus) utility/fitness/energy function
  - $f_0(x): R^n \to R$
  - $f_0(x)$ could be a vector function → multiple-criterion problem

Optimization Problem

$$
\begin{aligned}
\min_{x} \quad & f_0(x) \\
\text{s.t.} \quad & f_i(x) \le 0, i = 1, \dots, m \\
& h_i(x) = 0, i = 1, \dots, p
\end{aligned}
$$

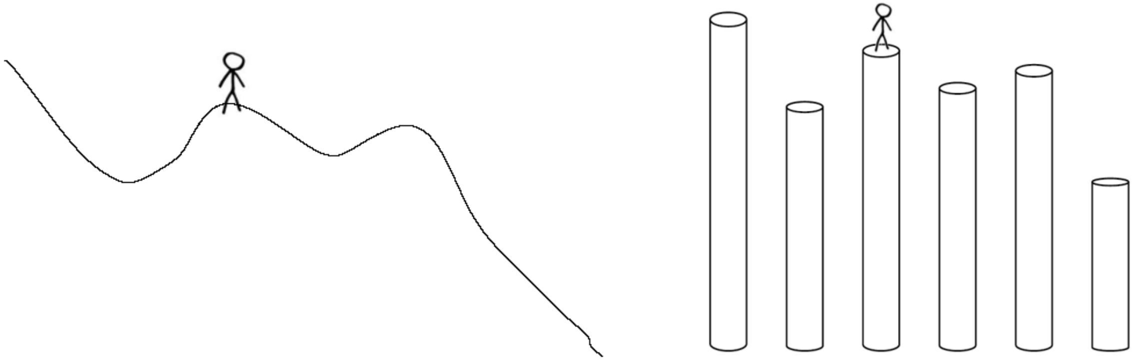


- $f_i(x), h_i(x)$: constratins
  - $f_i(x)$ inequality constraints
  - $h_i(x)$ equality/equation constraints
  - any x satisifying $f_i(x) \le 0$, $h_i(x) = 0$ is called a *feasible solution*
  - all the feasible solutions form the *feasible set/feasible domain*
  - the feasible set is not empty, the problem is *feasible*
- optimal solution $x^* = \underset{x}{\operatorname{argmin}}\{f_0(x), \text{s.t.} \dots\}$: no better feasible solution

Optimization Variable

$$
\begin{aligned}
\min_{x} \quad & f_0(x) \\
\text{s.t.} \quad & f_i(x) \le 0, i = 1, \dots, m \\
& h_i(x) = 0, i = 1, \dots, p
\end{aligned}
$$

- $x \in D$: it is very important to distinguish the possible value of x
  - real
  - Boolean (0/1)
  - integer



continuity is very important

Sudoku problem

- requirement
  - each column
  - each row,
  - each 3×3 subregion

  have different values but equal summarization

- if the item chooses value from real
  - very easy
- if it can only take integer 1,2,…,9
  - a complicated problem
  - could solved by continuous optimization

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Optimization Variable

$$\begin{aligned} \min_{x} \quad & f_0(x) \\ \text{s.t.} \quad & f_i(x) \leq 0, i = 1, \dots, m \\ & h_i(x) = 0, i = 1, \dots, p \end{aligned}$$

- $x \in D$: it is very important to distinguish the possible value of x
  - if the variables can take any real values, the problem is relatively easy

    many machine learning tasks can be modeled as continuous problems, which will be our main focus

  - some or all variables are constrained to take on integer values, it is called *integer programming*, usually results in a *combination programming*

    combination programming becomes more and more important in machine learning, e.g., alpha-GO, auto-ML, network architecture search (NAS), ...

目录 Contents

1 What is Optimization

2 History of Optimization

3 Optimization in Machine Learning

4 Course Information

Linear Programming

- **General Problem**

$$\begin{aligned} \min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\ \text{s.t.} \quad & f_i(x) \le 0, i = 1, \ldots, m, \\ & h_i(\mathbf{x}) = 0, i = 1, \ldots, p. \end{aligned}$$

- **Linear Programming**

$$x^\top y = \langle x, y \rangle = \Sigma_i \, x(i) y(i)$$

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}_0^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{f}_i^\top \mathbf{x} + a_i \le 0, i = 1, \ldots, m, \\ & \mathbf{h}_i^\top \mathbf{x} + b_i = 0, i = 1, \ldots, p. \end{aligned}$$

Linear Programming

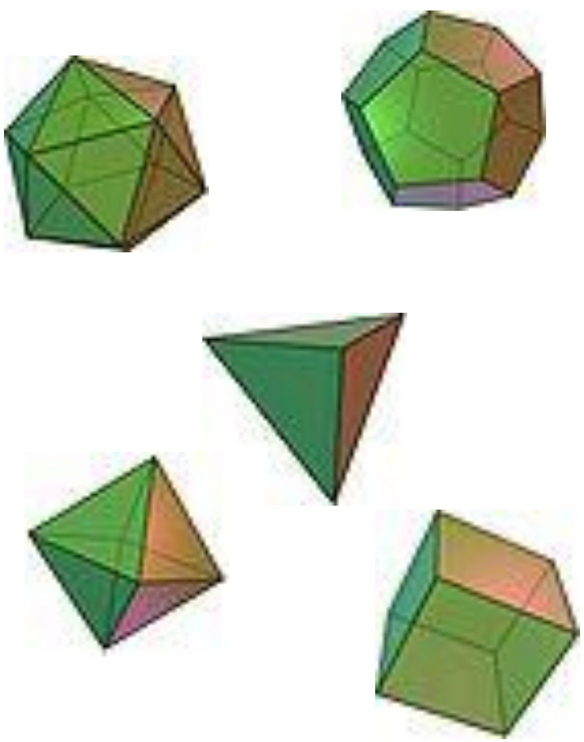
- **General Problem**

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & f_0(\mathbf{x}) \\
\text{s.t.} \quad & f_i(x) \le 0, i = 1, \ldots, m, \\
& h_i(\mathbf{x}) = 0, i = 1, \ldots, p.
\end{aligned}
$$

- **Linear Programming**

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{f}_0^\top \mathbf{x} \\
\text{s.t.} \quad & \mathbf{f}_i^\top \mathbf{x} + a_i \le 0, i = 1, \ldots, m, \\
& \mathbf{h}_i^\top \mathbf{x} + b_i = 0, i = 1, \ldots, p.
\end{aligned}
$$

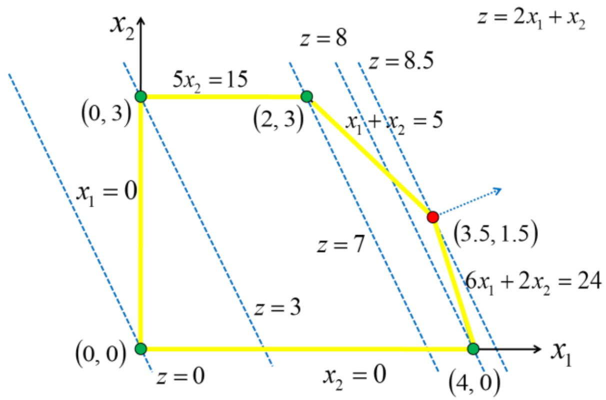
  - minimize a linear function with linear constraints
  - minimize a linear function over a **polyhedron**



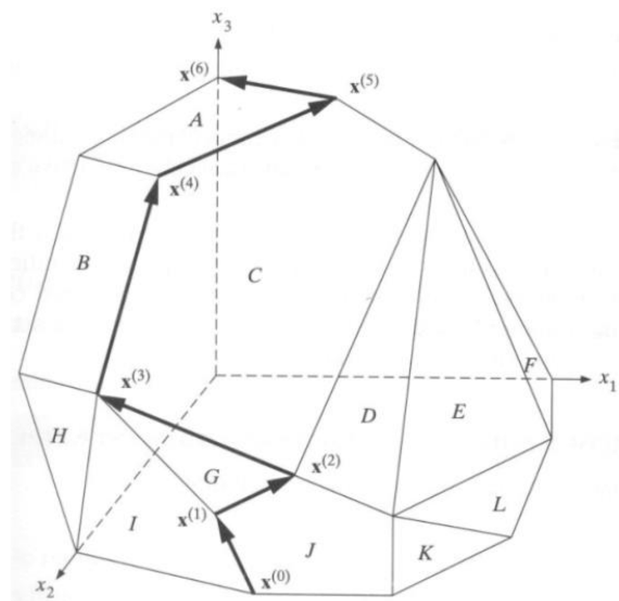
Simplex Method

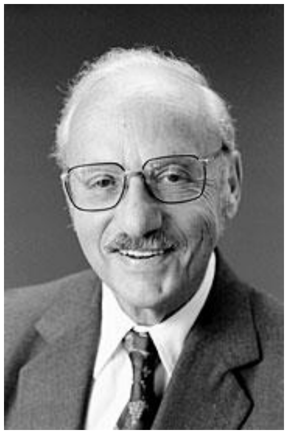
- **Dantzig: simplex method**
  - one of the vertices is optimal
  - search along the edges

"A few days later I apologized to Neyman for taking so long to do the homework — the problems seemed harder to do than usual"
George B. Dantzig (1914-2005)



$$
\begin{aligned}
\max \quad & z = 2x(1) + x(2) \\
\text{s.t.} \quad & x(1) + x(2) \le 5 \\
& 6x(1) + 2x(2) \le 24 \\
& 5x(2) \le 15 \\
& x(1) \ge 0, x(2) \ge 0
\end{aligned}
$$



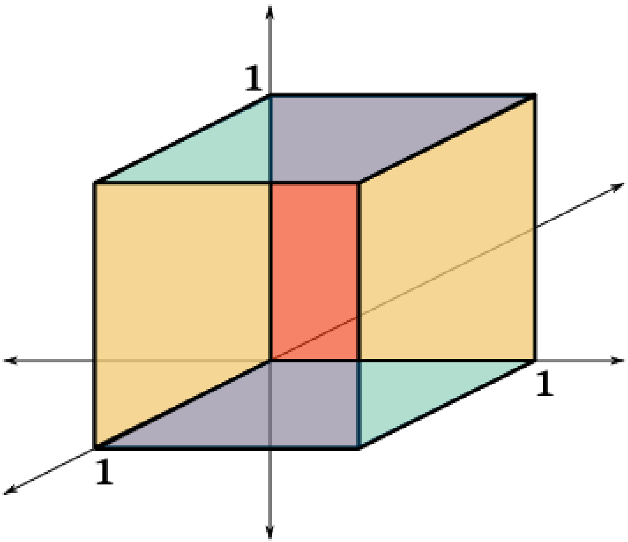


Discussion from Complexity Theory

- **1972, Klee-Minty cube**

  Simplex Method is NP-hard

  (non-deterministic polynomial-time)



- **1979, Khachain** designed **Ellipsoid Method**,

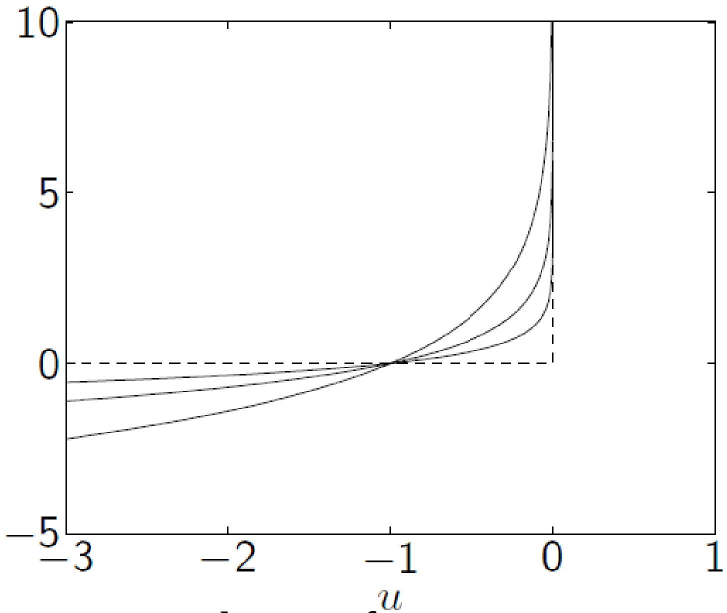
  which is proved that LP could be solved in polynomial time.

- Simplex method has high complexity, but is quite efficient in practice

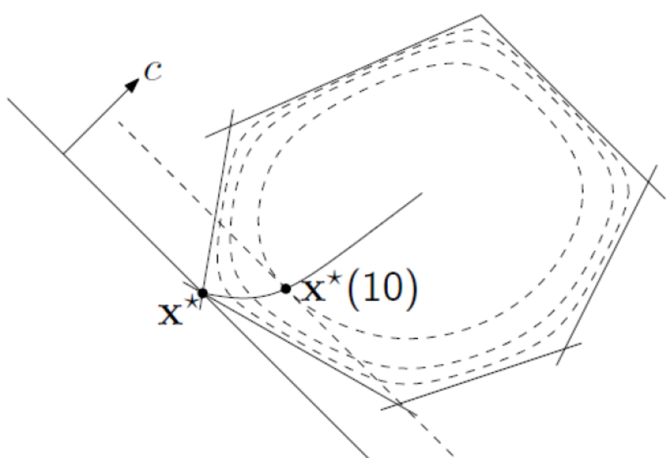
  Ellipsoid method has low complexity, but is not efficient in practice

Interior-point-method

- **1984, Karmarkan** designed **interior-point-method**
  - **von Neumanm** 1940s
  - low complexity & efficient



barrier function

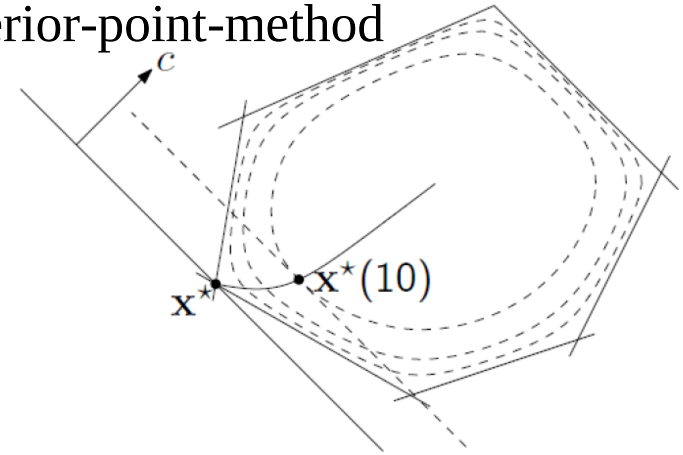


central path

S. Boyd and L. Vandenberghe, Convex Optimization

Interior-point-method for Convex Optimization

- Polynomial property seems less important in interior-point-method



- 1994, **Nemirovski** and **Nesterov** gave the interior-point-method for **Convex Optimization**

- Convex Optimization:

  minimize a **convex function**

  over a **convex set**

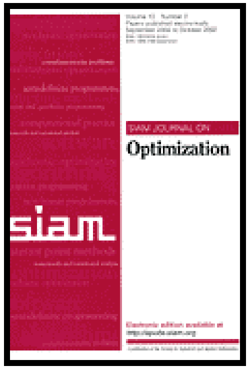
Convex Optimization

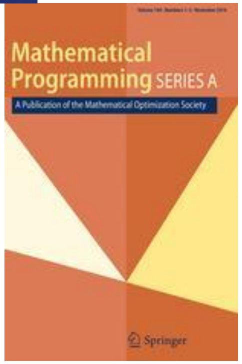
"convex optimization is an important enough topic that everyone who uses computational mathematics should know at least a little bit about it .... is a natural next topic after advanced linear algebra, and linear programming." —— S. Boyd and L. Vandenberghe

Trend in Optimization

From convex to non-convex problems; Towards large scale; Towards applications

- Convergence analysis: condition, speed, acceleration
- Parallel computing
- Stochastic optimization
- Robust optimization
- Discrete optimization
- Intelligent optimization





目录 Contents

1 What is Optimization

2 History of Optimization

3 **Optimization in Machine Learning**

4 Course Information

Basic Learning Framework

- *"Machine learning is at the core of artificial intelligence and data science."*

  —— Michael I. Jordan

- Machine learning is to establish f from data generated by unknown g
- $f(x)$ and $g(x)$ could be similar but the structures of f and g generally are totally different

$$\min_{f \in F} \; p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

$$f^* \approx g$$

"All models are wrong but sometimes useful" —— George Box

Loss Function

- Regression
- Classification
- Clustering
- Dimension reduction

……

$$\boxed{\min_{f \in F}} \; \boxed{p(f)} + \boxed{c(f)}$$

how to learn

where to learn

what to learn

Prior knowledge

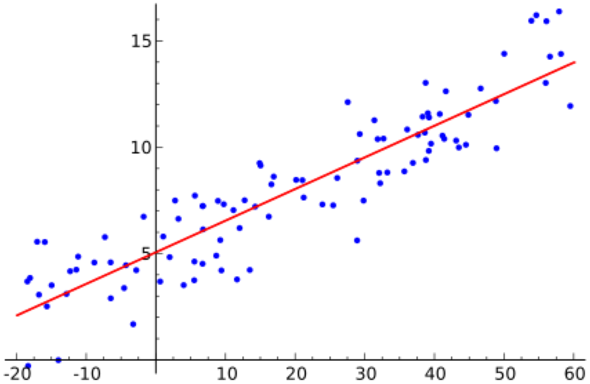
Loss Function

- **Regression**
- Classification
- Clustering
- Dimension reduction

……

$$\sum_i (f(x_i) - y_i)^2$$

- other requirements?
- why squares?
- why sum?



$$\underbrace{\min_{f \in F}} \; p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

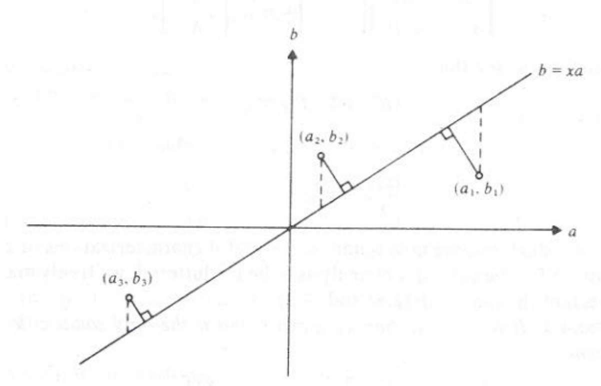
Loss Function

- **Regression**
- Classification
- Clustering
- Dimension reduction

……

$$\sum_i (f(x_i) - y_i)^2$$

- other requirements?
- why squares?
- why sum?



$$\min_{f \in F} \; p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

consider about learning and optimization simultaneously

Loss Function

- Regression
- **Classification**
- Clustering
- Dimension reduction

……

$$\sum_i I(f(x_i) \neq y_i)$$

- misclassification loss
- however discontinues

$$\sum_i \max\{0, 1 - y_i f(x_i)\}$$

- hinge loss

- quadratic cost
- cross-entropy
- softmax (log-likelihood)

$$\min_{f \in F} \; p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

consider about learning and optimization simultaneously

Loss Function

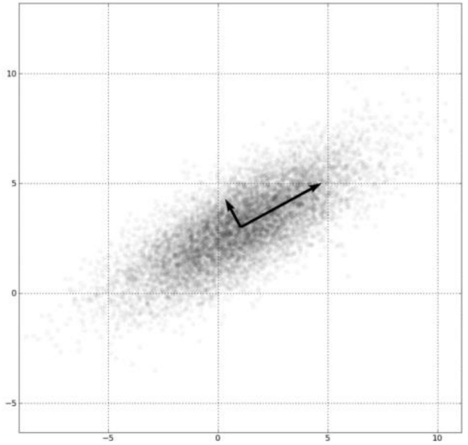
- Regression
- Classification
- Clustering
- **Dimension reduction**

……

Project data in to one direction

$$w^\top x$$

to reserve information



$$\boxed{\min}_{\boxed{f\in F}} \boxed{p(f)} + \boxed{c(f)}$$

how to learn

where to learn

what to learn

Prior knowledge

consider about learning and optimization simultaneously

Loss Function

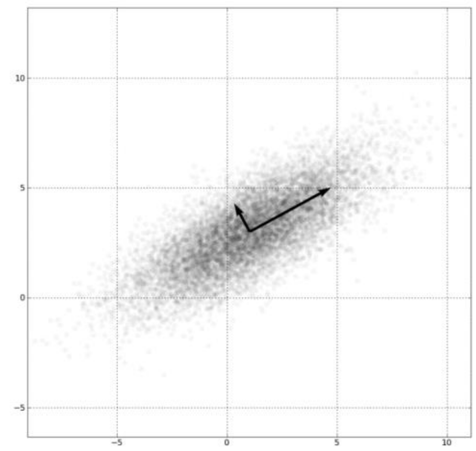
- Regression
- Classification
- Clustering
- **Dimension reduction**

……

Project data in to one direction

$$w^\top x$$

to reserve information



maximize the covariance of the (centralized) projected data

$$-w^\top X^\top X w$$

$$\min_{f \in F} p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

consider about learning and optimization simultaneously

Regularization Term

- the minimizer of empirical loss (loss on the training data) is not necessarily optimal to expected loss (loss on the unknown distribution/new data)

$$R_\rho(f) \leq R_{\mathrm{empirical}}(f) + R_{\mathrm{structural}}(f)$$

- this general inequality implies the importance of complexity control

$$\min_{f \in F} p(f) + c(f)$$

how to learn

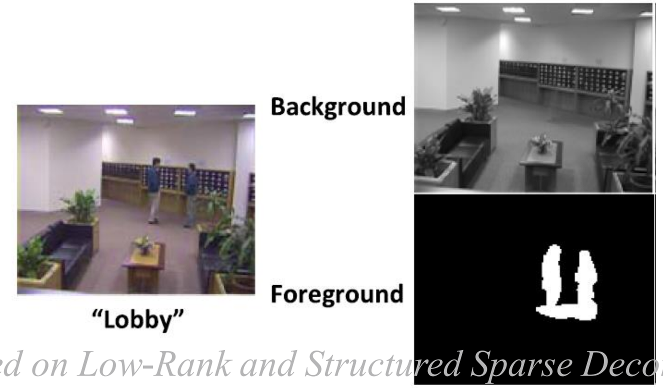
where to learn

what to learn

Prior knowledge

Regularization Term

- we want to decompose foreground and background,



X. Liu et al. "Background Subtraction Based on Low-Rank and Structured Sparse Decomposition", IEEE-TIP 2015

$$\min_{f \in F} p(f) + c(f)$$

how to learn

where to learn

what to learn

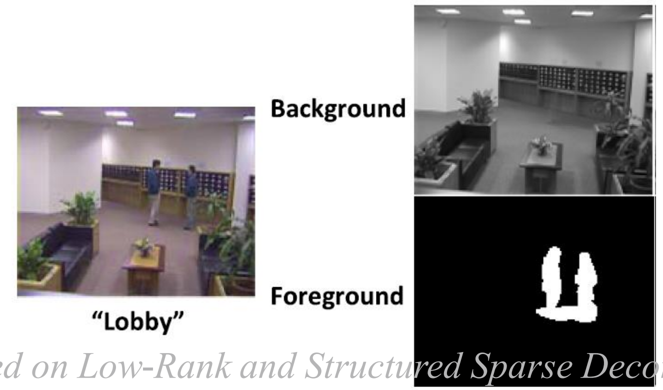
Prior knowledge

how to efficiently model prior knowledge

Regularization Term

- we want to decompose foreground and background,

$$\min_{F,B} \|H-(F+B)\|_2 + \lambda||F||_1 + \gamma||B||_*$$



Background is **low-rank**

Foreground is **sparse**

X. Liu et al. "Background Subtraction Based on Low-Rank and Structured Sparse Decomposition", IEEE-TIP 2015

$$\min_{f \in F} \; p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

how to efficiently model prior knowledge

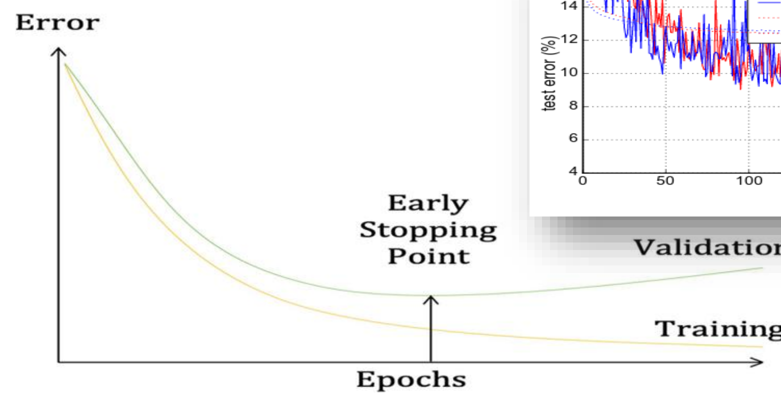
Regularization Term

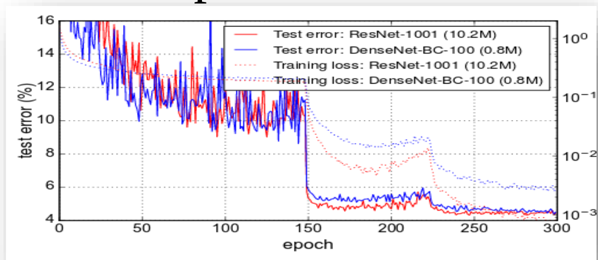
- we want to decompose foreground and background,

X. Liu et al. "Background Subtraction Based on Low-Rank and Structured Sparse Decomposition", IEEE-TIP 2015

Regularization Term

- The essence of regularization term is to control the functional space
  - Transfer learning
  - Manifold constraint
  - Early Stop
  - Overparametrization





$$\min_{f\in F}\; p(f) + c(f)$$

how to learn

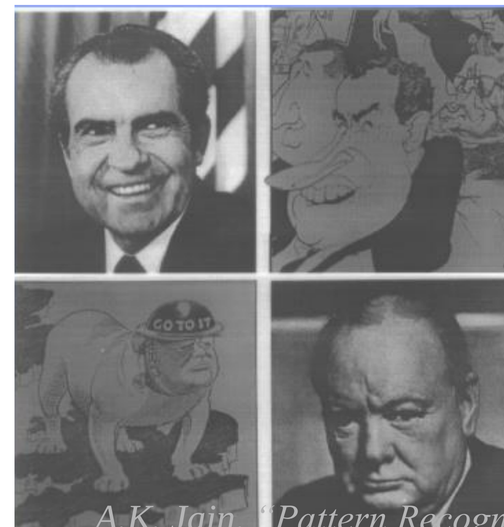
where to learn

what to learn

Prior knowledge

Functional Space

- Functional space becomes more and more complex：
  - linear model $f(x) = w^\top x + b$
  - basis function $f(x) = w^\top \phi(x) + b$
  - kernel method
  - neural networks



A.K. Jain, "Pattern Recognition"

$$\min_{f \in F} p(f) + c(f)$$

how to learn

where to learn

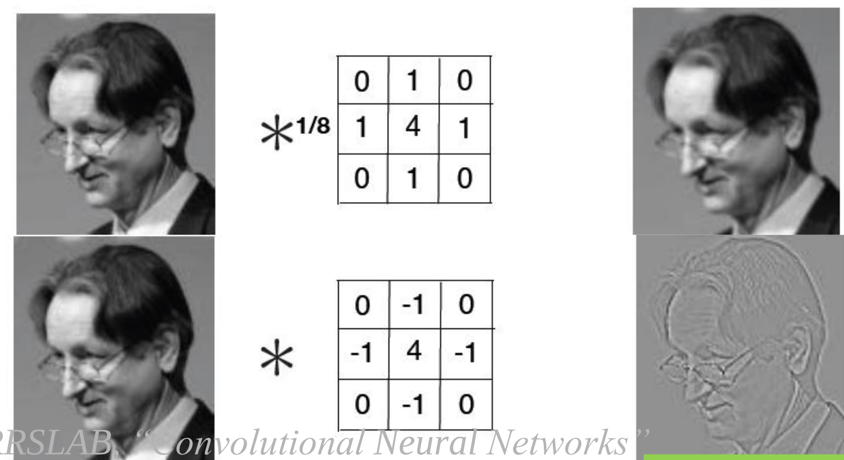
what to learn

Prior knowledge

can we learn features automatically in a data-driven way

Functional Space

- Convolution operator：



PARRSLAB, “Convolutional Neural Networks”

$$\min_{f\in F} p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

can we learn features automatically in a data-driven way

Functional Space

- Convolution operator：

$$* \frac{1}{8}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$* \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$* \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

can you image the result?

PARRSLAB, "Convolutional Neural Networks"

$$\min_{f \in F} p(f) + c(f)$$

how to learn

where to learn

what to learn

Prior knowledge

can we learn features automatically in a data-driven way

Functional Space

- Convolution operator：

$$* \tfrac{1}{8}\begin{bmatrix}0 & 1 & 0\\ 1 & 4 & 1\\ 0 & 1 & 0\end{bmatrix}$$

$$* \begin{bmatrix}0 & -1 & 0\\ -1 & 4 & -1\\ 0 & -1 & 0\end{bmatrix}$$

PARRSLAB, "Convolutional Neural Networks"

why it should be 2? How about 1.99 and 2.01

$$* \begin{bmatrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{bmatrix}$$

can you image the result?

$$\min_{f \in F} \; p(f) + c(f)$$

how to learn

where to learn

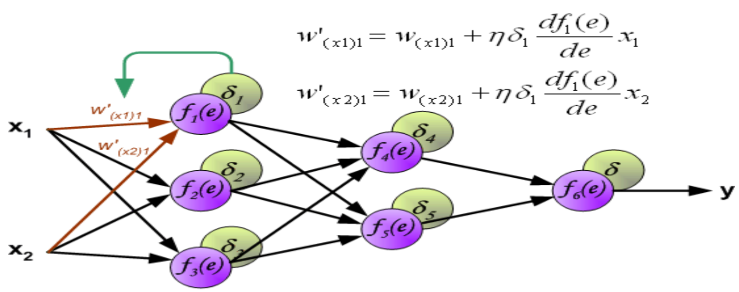
what to learn

Prior knowledge

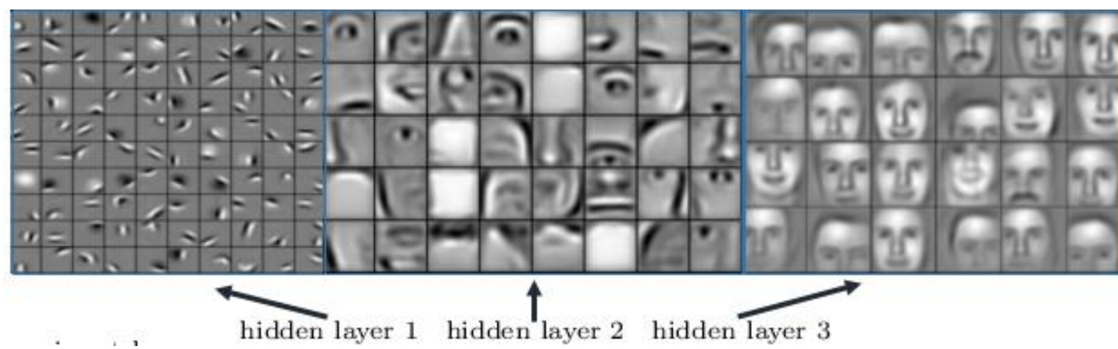
can we learn features automatically in a data-driven way

Functional Space

- by backpropagation, we can optimize the parameters and learn features



http://www.rsipvision.com/exploring-deep-learning/



$$\min_{f \in F} p(f) + c(f)$$

how to learn

where to learn

what to learn

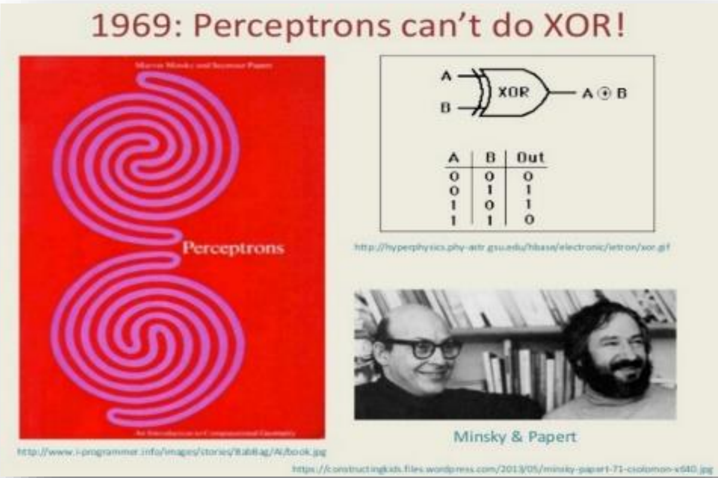
Prior knowledge

can we learn features automatically in a data-driven way

Optimization

Recall the development of neural networks

- 1950’s Perceptrons
- 1980’s CNN, RNN (original)



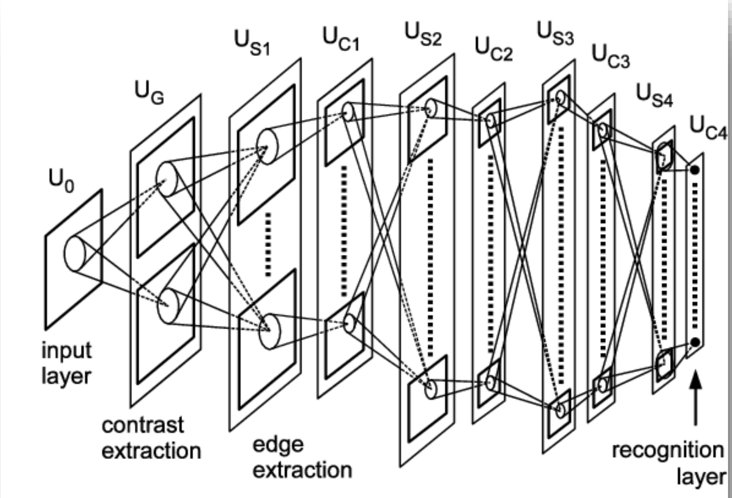


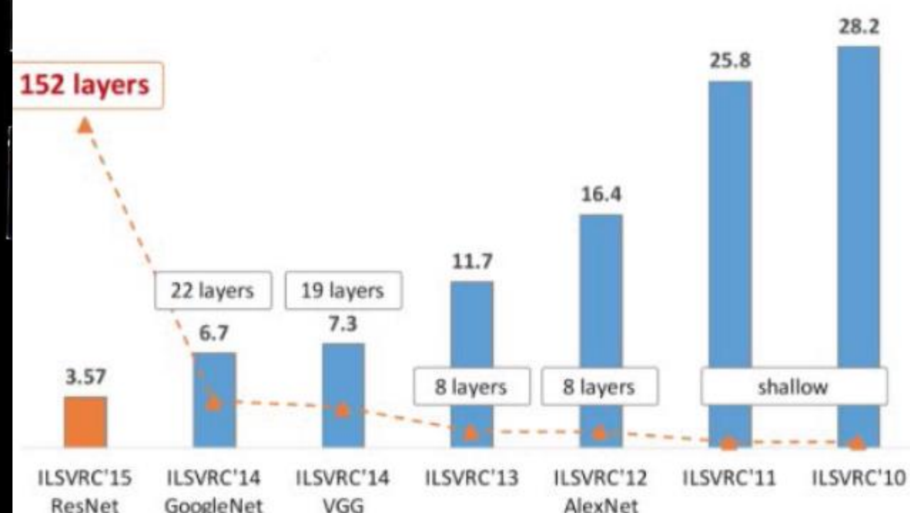
Fig. 1. The architecture of the neocognitron (Fukushima, 2003).

Kunihiko Fukushima
福島 邦彦

Optimization

Recall the development of neural networks

- 1950's Perceptrons
- 1980's CNN, RNN
- **2005- Deep Neural Networks**



Optimization

Recall the development of neural networks

- 1950's Perceptrons
- 1980's CNN, RNN
- **2005- Deep Neural Networks**

when we can efficiently optimize the parameters, we can use more and more complicated models

$$\min_{f\in F} p(f) + c(f)$$

how to learn

where to learn

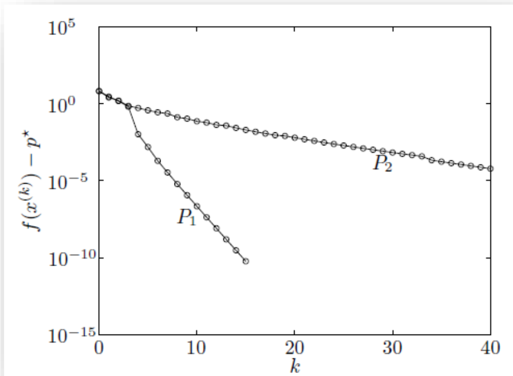
what to learn

Prior knowledge

Optimization in Machine Learning

For the same problem, the difference between different algorithms could be vast.

- Solving accuracy
- Computational time
- Different operations
    - GPU
    - CPU
    - vector-friend
    - matrix/tensor-friend



S. Boyd and L. Vandenberghe, Convex Optimization

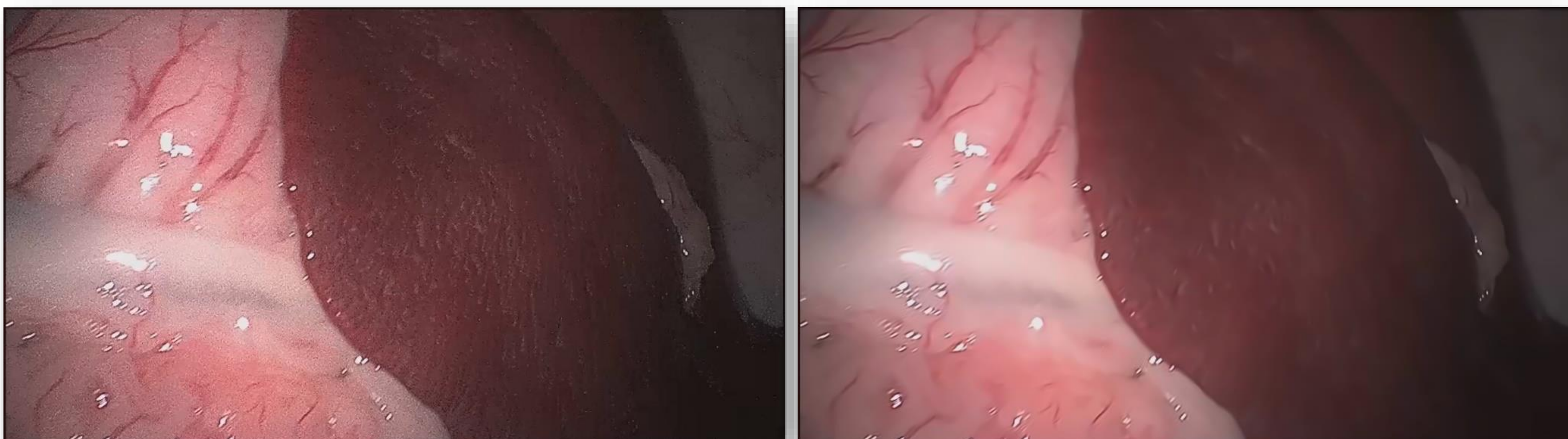
TABLE I
AVERAGE COMPUTATIONAL TIME WITH DIFFERENT M, N WHEN $K = 100, s_n = 10, s = 10\%$.

Methods	M=500 N=1000	M=1000 N=1000	M=500 N=2000	M=1000 N=2000	M=1500 N=2000
LASSO	0.0121 s	0.0436 s	0.0296 s	0.1333 s	0.1563 s
RDCS	0.9355 s	0.5129 s	8.5300 s	7.9630 s	5.5730 s
M1bit-CSC	0.9627 s	1.0500 s	8.5410 s	9.2600 s	8.7560 s
Alg.1–sL1	0.0929 s	0.1340 s	0.3713 s	0.5177 s	0.7089 s
Alg.1–MCP	0.1306 s	0.1663 s	0.5907 s	0.7127 s	0.7265 s
Alg.1–L0	0.1073 s	0.1430 s	0.5604 s	0.6758 s	0.6958 s
Alg.1–L1	0.1004 s	0.1375 s	0.5548 s	0.6671 s	0.6854 s

F. He, X. Huang, Y. Ming, Fast Signal Recovery from Saturated Measurements

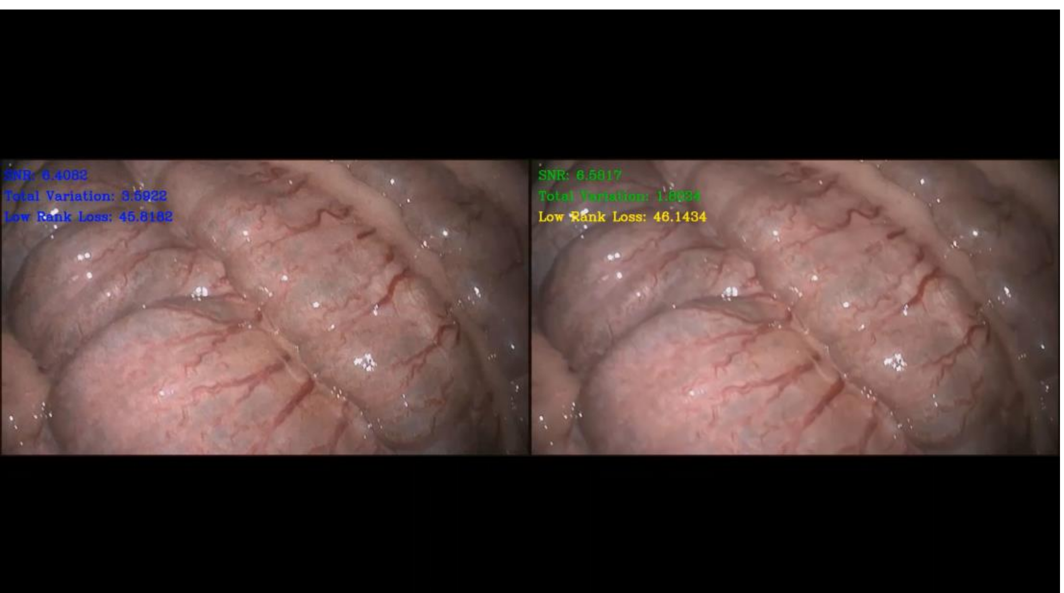
Optimization in Machine Learning

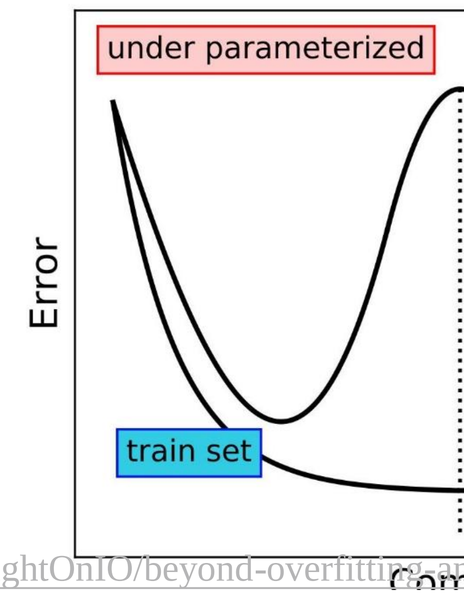
- Accuracy requirement is **not** very high, especially for **large-scale problem**
  - modeling has error, e.g., image quality evaluation itself is a challenging problem



Optimization in Machine Learning

- Accuracy requirement is **not** very high, especially for **large-scale problem**
  - modeling has error, e.g., image quality evaluation itself is a challenging problem
  - there is gap between training and test

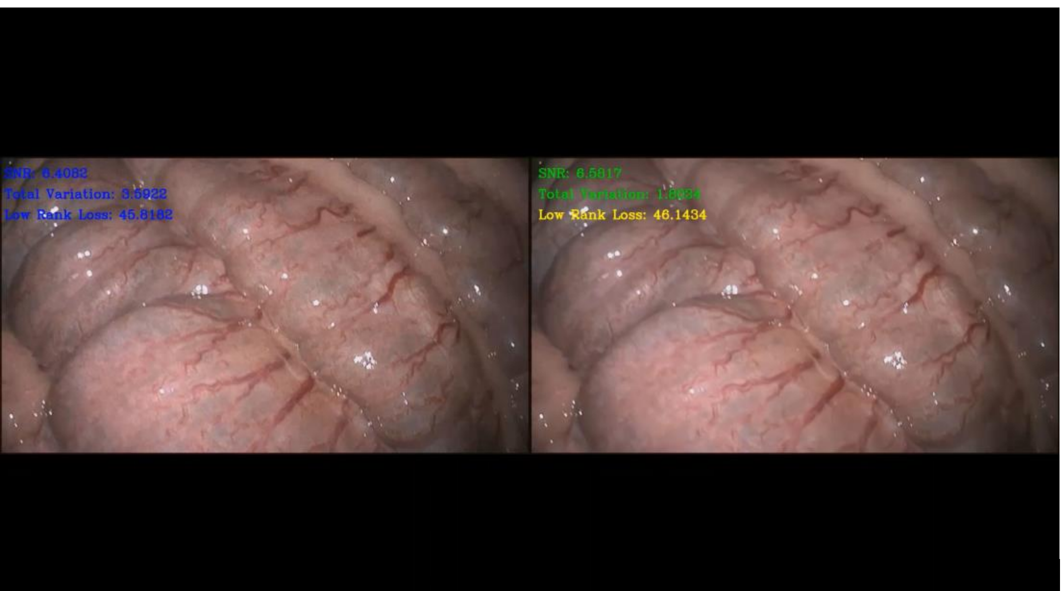


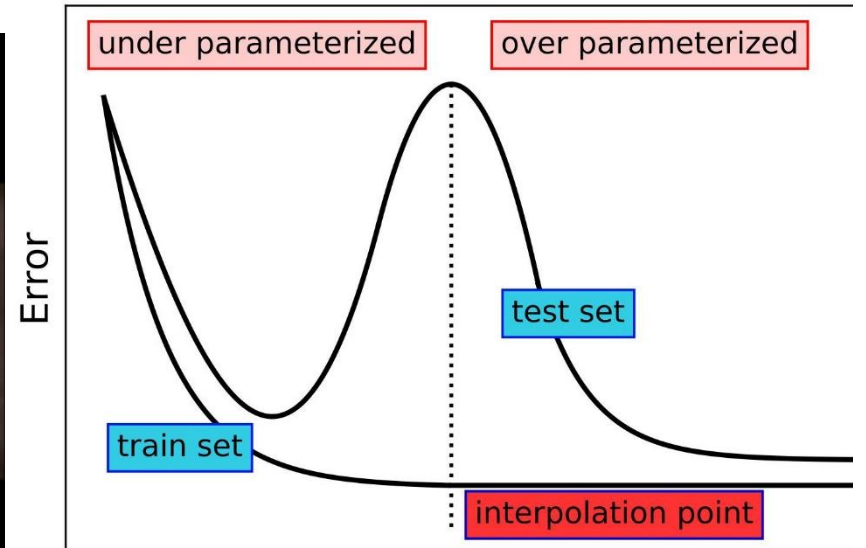


https://medium.com/@LightOnIO/beyond-overfitting-ar
beyond-silicon-the-double-descent-curve-18b6d9810e1b

Optimization in Machine Learning

- Accuracy requirement is **not** very high, especially for **large-scale problem**
  - modeling has error, e.g., image quality evaluation itself is a challenging problem
  - there is gap between training and test
  - in other topics, e.g., scheduling, signal processing, accuracy **is very important**





https://medium.com/@LightOnIO/beyond-overfitting-ar
beyond-silicon-the-double-descent-curve-18b6d9810e1b

Optimization in Machine Learning

- Accuracy requirement is **not**
  - modeling has error, e.g., in … m
  - there is gap between traini
  - in other topics, e.g., schedu

Problem Scale	Operations
small size	any, e.g., Hessian matrix
medium size	inverse A^{-1}
large size	multiplication Ax
huge size	addition $x + y$

—— **Yurri Nesterov**

- Scale could be **very large**

$$\min_{x} \ \Sigma_{i=1}^{m} (f(x_i) - y_i)^2$$

- Objective function usually could be **decomposed** in some way
- Objective function is usually **non-convex**

- Many things need to consider when put into **practical application**

目录 Contents

1 What is Optimization

2 History of Optimization

3 Optimization in Machine Learning

4 Course Information

Course Information

- Lecturer
  - Xiaolin Huang, Department of Automation, SEIEE
  - xiaolinhuang@sjtu.edu.cn www.pami.sjtu.edu.cn
- TAs： Kaijie Wang, kaijie_wang@sjtu.edu.cn

  Yingwen Wu, yingwen_wu@sjtu.edu.cn
- Text Book:

Stephen Boyd, Lieven Vandenberghe :

Convex Optimization

Cambridge (free download)

王书宁，许鋆，黄晓霖 [译] 凸优化 清华大学出版社 2013

Course Information

- Full score: 100

  class performance/quiz (20) + homework (30) + project (10) + examination (40)

- Schedule (planed)

1. **Tutorial** (2)
2. **Convex Set** (3) **Convex Function** (3) **Convex Optimization** (4) **Convex Machine Learning Models** (4)
3. **Solving Algorithm for Non-constrained Problems** (6) **Large-scale Algorithm I** (6) **Nonconvex Models** (4)
4. **Duality** (6) **Large-scale Algorithm II** (6)
5. **Report Presentation** (2) **Outlook and Conclusion** (2)

THANKS