京都大学集中講義

機械学習と深層学習の数理と応用 (1)

鈴木大慈
東京大学大学院情報理工学系研究科数理情報学専攻
理研AIP
Japan Digital Design

2018年12月17日

自己紹介

- **現職：**
  - 東京大学大学院情報理工学系研究科
    数理情報学専攻　准教授
  - 理研AIP「深層学習理論チーム」チームリーダー

- **専門：**
  - 機械学習の数理
    - 汎化誤差理論
    - 確率的最適化
  - 数理統計学
    - 高次元統計
    - ノンパラメトリック法

本講義の目的

- 対象：機械学習初学者
- 機械学習の数理，特に深層学習の数理を解説
- Pythonによる簡単な実装も体験してもらう

評価

- 出席とレポート
- レポート内容：
  - 数学的問題に回答
  - Pythonによる深層学習の実装 (Google colab)

講義の予定

スライドを用いた講義

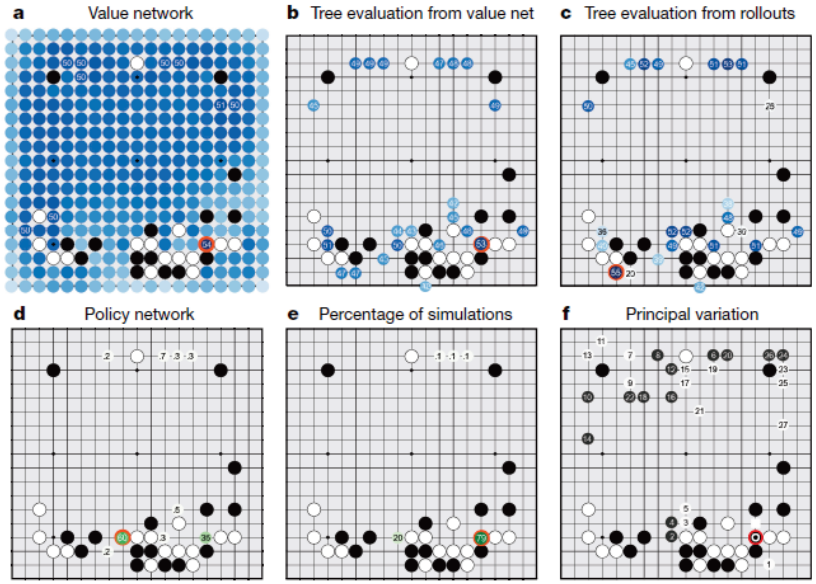
- １回目：機械学習全般の概論
- ２回目：深層学習の概論

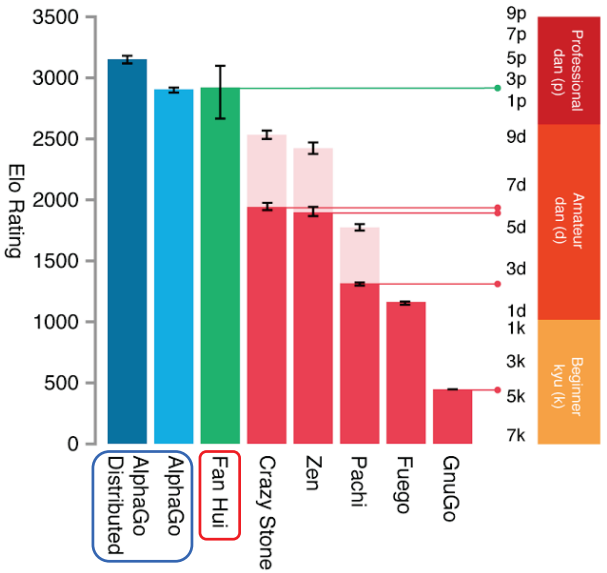
黒板を用いた講義

- ３回目：ニューラルネットワークの近似理論
  - 万能近似能力，Sobolev空間における近似精度
- ４回目：NNの汎化誤差解析
  - Rademacher複雑度
- ５回目：速い汎化誤差の収束レート，その他
  - Talagrandの不等式，ミニマックスレート最適性

AlphaGo/Alpha Zero

深層学習+強化学習+モンテカルロ木探索+自己対戦

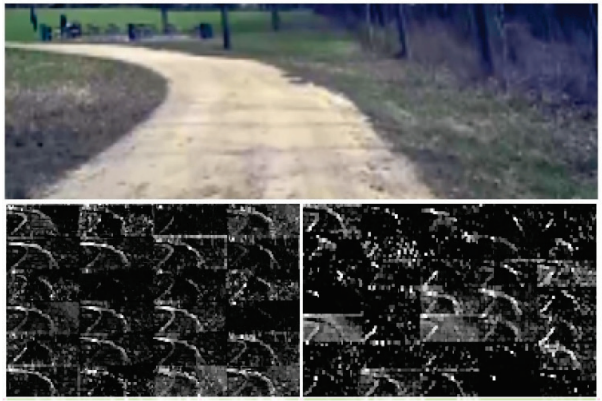




2015年10月5〜9日欧州覇者Fan Hui (2段) に5-0で勝利
2016年3月9〜15日イ・セドルと対局し4-1で勝利
2017年5月23〜27日柯潔と対局し3-0で勝利
2017年12月 AlphaZeroが4時間の自己対戦のみでelmo, AlphaGo Zeroを上回る.

[Silver et al. (Google Deep Mind): Mastering the game of Go with deep neural networks and tree search, Nature, 529, 484—489, 2016]

自動運転



詳報：トヨタが頼った謎のAI半導体メーカー

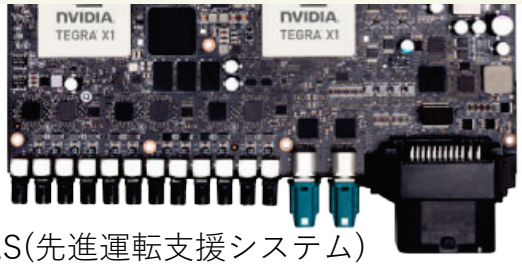
産業秩序が激変、自動車を「操る」のは誰だ

島津 翔　　バックナンバー

2017年5月22日（月）

AI（人工知能）による産業構造の激変が始まった。

売り上げ規模など従来の序列は全く関係ない。対応できない既存勢力は没落する。強固なピラミッドを持つ自動車産業とて安泰ではない。AIによる自動運転の実用化が、激変の号砲



ADAS(先進運転支援システム)

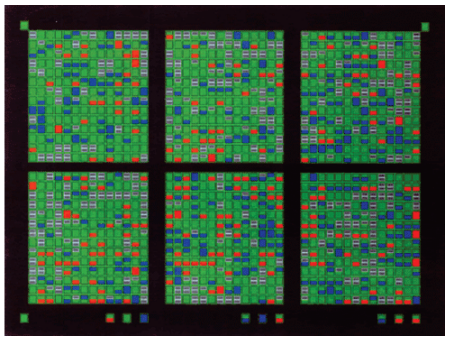
[End to End Learning for Self-Driving Cars, Nvidia 2016]

身近にあふれる機械学習

検索エンジン

推薦システム

遺伝子データ解析



音声認識

機械学習プラットフォーム

Google Cloud Platform

Microsoft Azure

amazon web services™

人工知能と機械学習

人工知能

本日の様相
「人工知能」≒「機械学習」

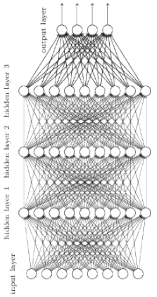
自分で問題設定ができ，
その解決もできる．

“汎用人工知能”

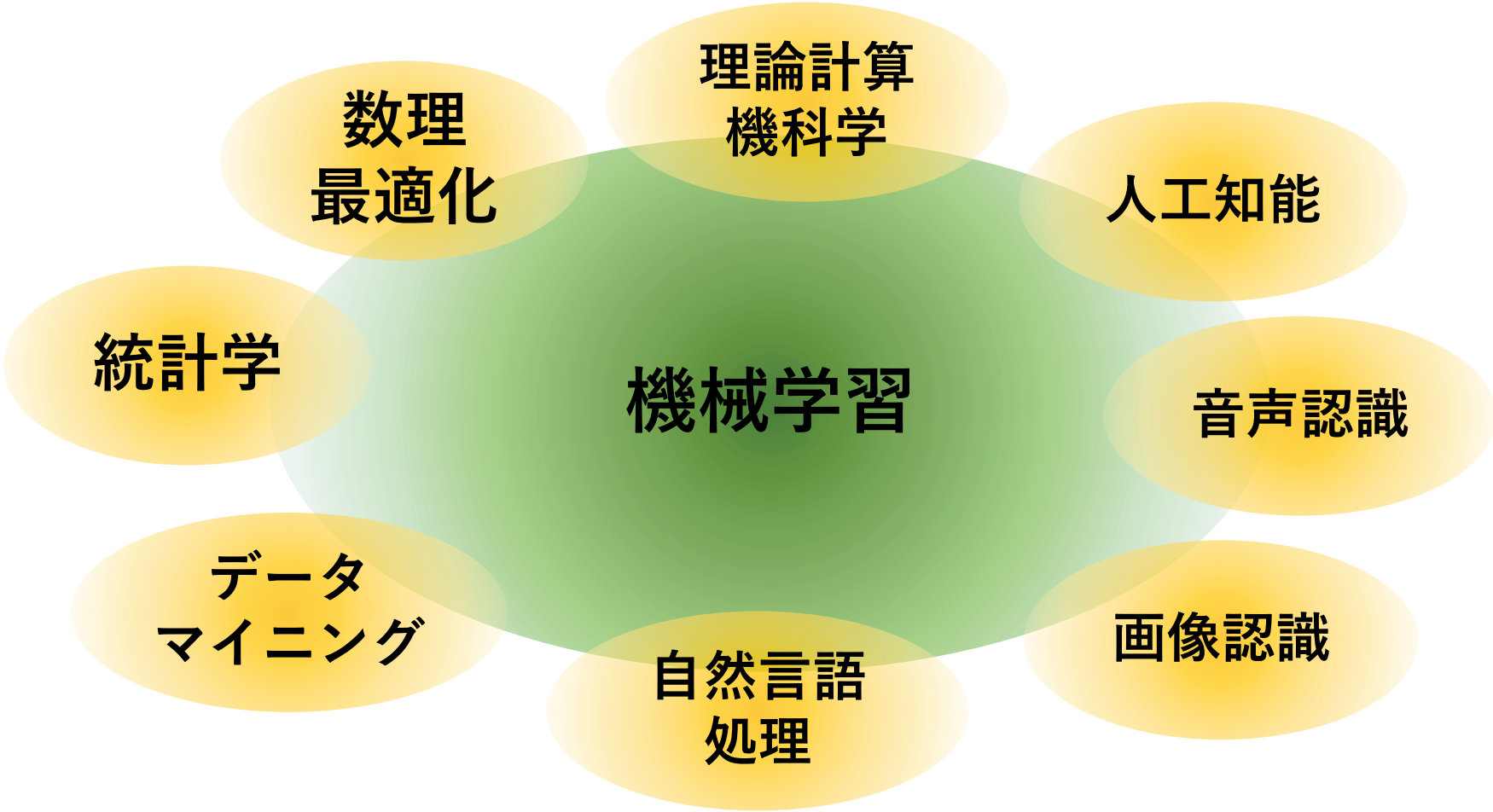
機械学習
統計的アプローチ

SVM
トピックモデル
スパース学習
テンソル学習
…

深層学習



機械学習の立ち位置



さまざまな分野の複合領域

機械学習コミュニティの実体

機械学習主要国際会議

NeurIPS (Neural Information Processing Systems)
ICML (International Conference of Machine Learning)
COLT (Conference of Learning Theory)
ICLR (International Conference on Learning Representations)
AISTATS, UAI, ECML, …

関連国際会議

- データマイニング
  KDD, ICDM, WWW, WISDM, SIGIR, SDM
- 人工知能
  IJCAI, AAAI
- コンピュータビジョン
  CVPR, ICCV, ECCV
- 自然言語処理
  ACL, NAACL, EMNLP, COLING

NIPS2015@Montreal

ICML2016@NYC

➢ 全て査読あり：ダブルブラインド，採択率20～25%
  - NIPS2016 (568/2500, 22.7%), ICML2016 (322/1327, 24.3%)

NeurIPS2018

- 投稿数：4856件
- 採択数：1011件
- 査読プロセス：double blind

(Senior) Area Chair: 約280人，査読者：約2500人

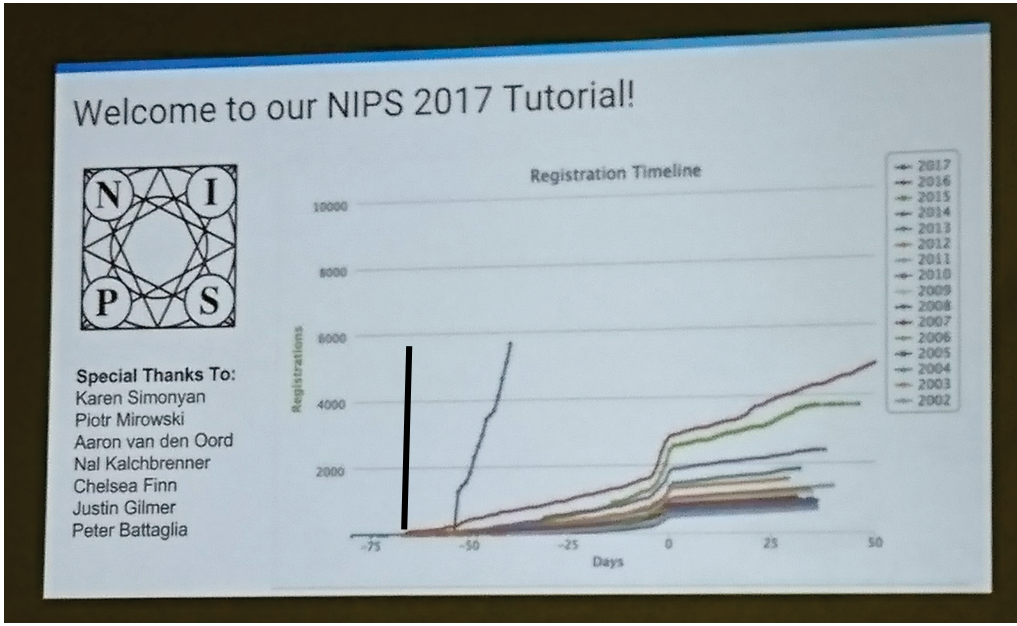
✓1査読者あたり6本を査読，そのうち1-2本がaccept

通常レジストレーション（2000席）

<u>11分38秒</u>で完売

（論文採録者，トップ査読者は別）

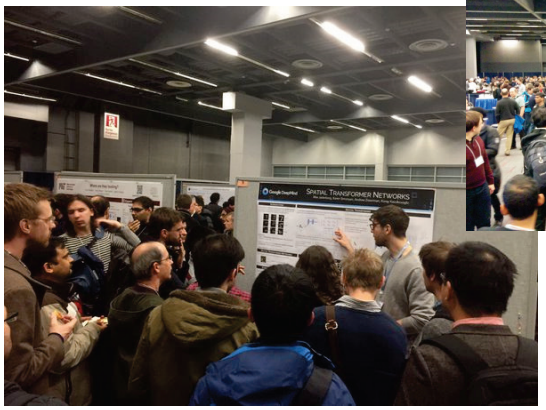
[去年は8000席が2週間で完売]

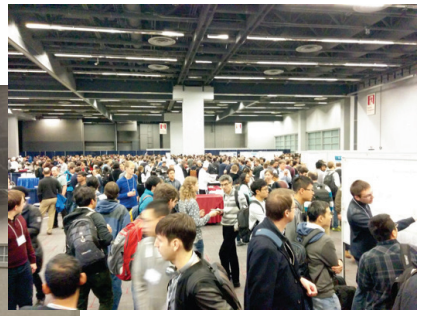


会場の様子

2015年の様子

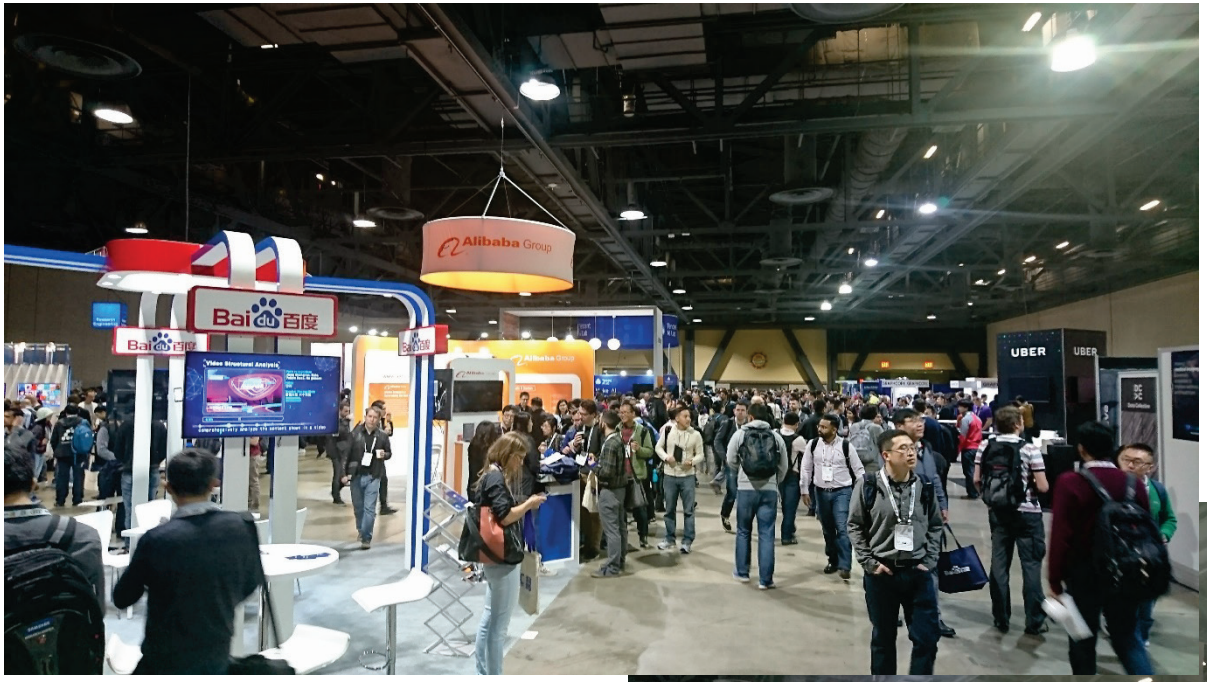
Deep learning tutorial

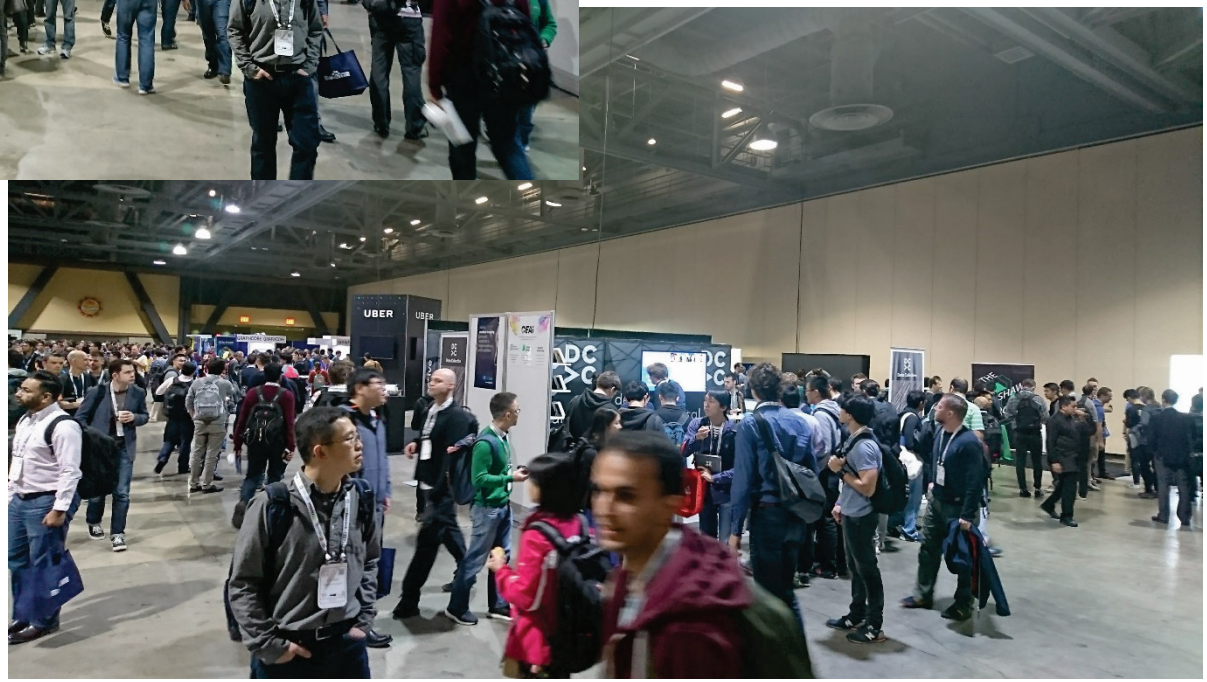




ポスター発表

企業ブースの様子（NIPS2017）

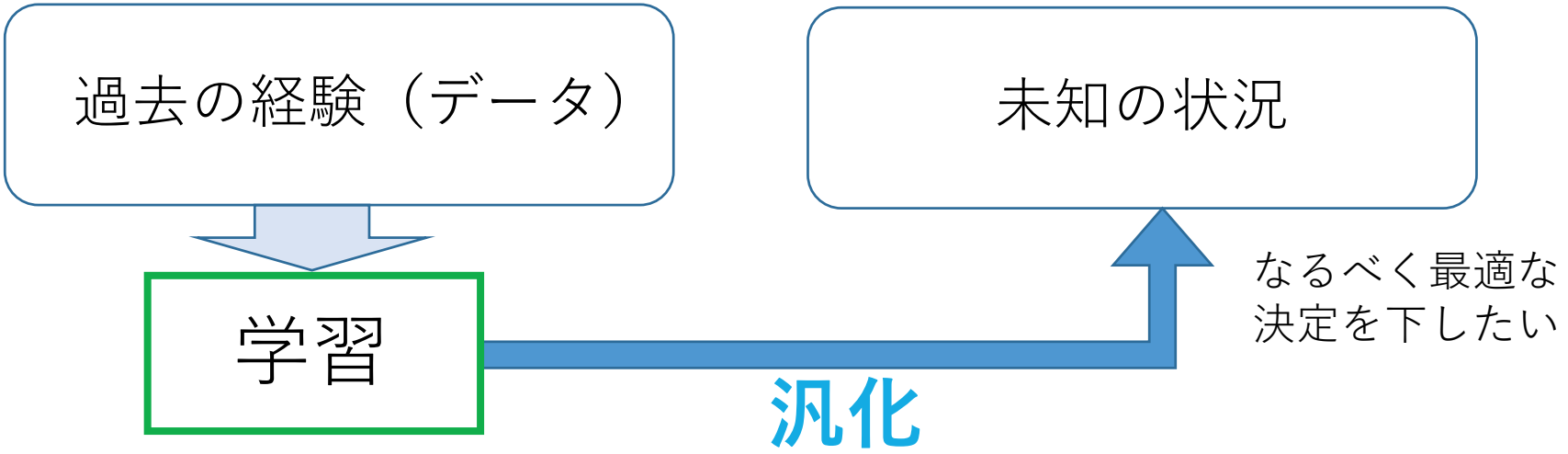




機械学習の目的

- 人間と同様の知的情報処理を計算機で実現するための技術・手法

Arthur Samuel 「Field of study that gives computers the ability to learn without being explicitly programmed 」（1959）



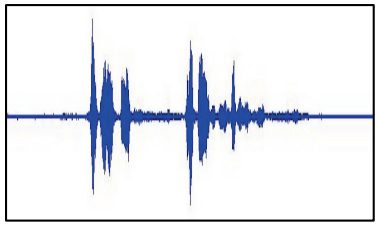
予測と推測

機械学習の活用法

予測

（より機械学習的）

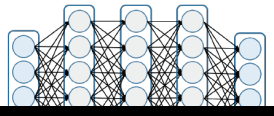
➡ 船



➡ “Hello”

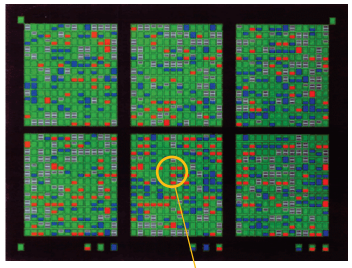
- Outcomeを正しく当てる.
- 解釈よりも予測精度を重視.

例：深層学習



推測

（より数理統計的）

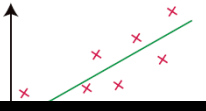


⟷ 肺癌

第〇〇遺伝子が肺癌に寄与
有意水準5%

- 原因の究明.
- 仮説検定は典型例.

例：線形回帰分析



理論的にはこの二つはある種のトレードオフの関係にある.

機械学習における数学の役割

1. 問題の数学的定式化（モデリング）
2. 手法の導出と記述（アルゴリズム）
3. 正当性の保証（学習理論）

記述言語：数学

- 確率-統計，線形代数，関数解析，最適化理論

機械学習の歴史

機械学習と人工知能の歴史

1946： ENIAC，高い計算能力
フォン・ノイマン「俺の次に頭の良い奴ができた」
1952： A. Samuelによるチェッカーズプログラム

統計的学習

ルールベース

1960年代前半：
ELIZA（イライザ），
擬似心理療法士

1980年代：
エキスパートシステム

人手による学習ルールの作りこみの限界
「膨大な数の例外」

Siriなどにつながる

1957：Perceptron，ニューラルネットワークの先駆け
第一次ニューラルネットワークブーム
1963：線形サポートベクトルマシン

線形モデルの限界

1980年代：多層パーセプトロン，誤差逆伝搬，
畳み込みネット
第二次ニューラルネットワークブーム
1992： 非線形サポートベクトルマシン
（カーネル法）

非凸性の問題

1996： スパース学習（Lasso）
2003： トピックモデル（LDA）
2012： Supervision（Alex-net）
第三次ニューラルネットワークブーム

データの増加
＋計算機の強化

人間

四則演算

単純なルール

機械

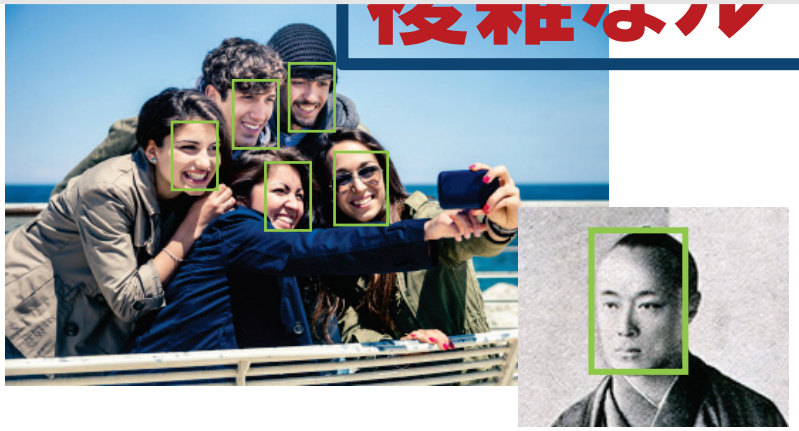
難

$193707721 \times 761838257287 - 2^{67} = -1$

易

人の手でプログラムするのは無理

learn without being explicitly programmed



易

難

人によって顔が違う，照明の当たり方で見え方・色が変わる，
表情の違い，髪型の違い，顔の向きの違い，．．．．

統計的学習の考え方

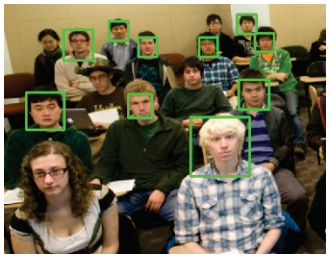
- 人がプログラムするのは認識の仕方ではなく学習の仕方

→数学で記述

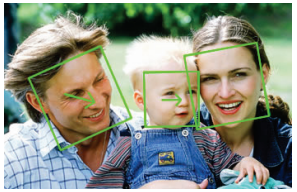
大量画像データ

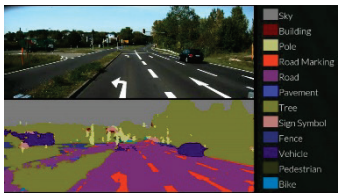
学習

初見の画像も
正しく認識
(汎化)



- 強い将棋ソフトを作りたい → 大量の棋譜データで学習
- 顔認識ソフトを作りたい → 大量の画像データで学習
- 車道を認識したい → 大量の車載カメラ画像で学習





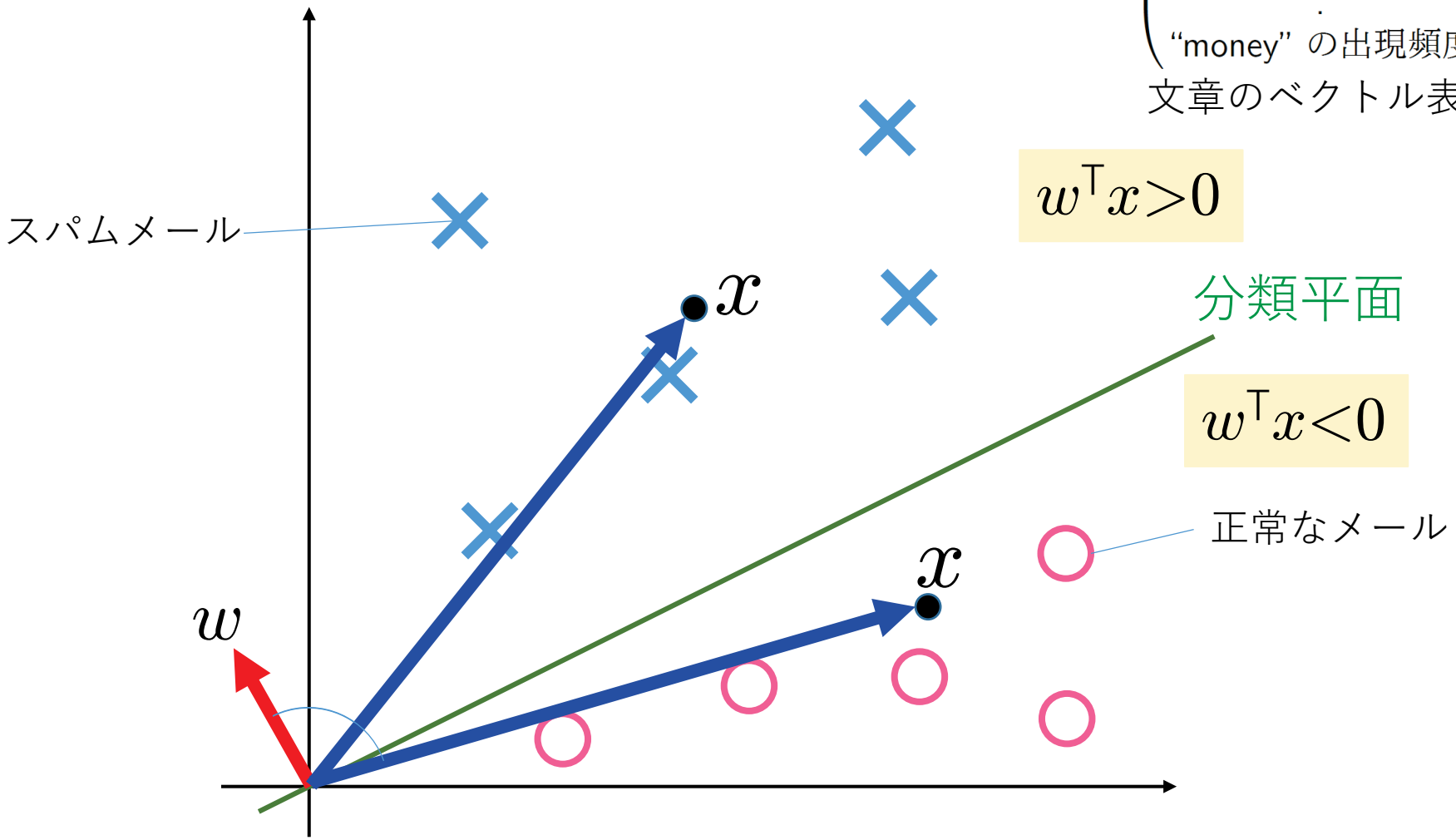
線形分類機

$$w^\top x = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$$

Bag-of-words

$$x = \begin{pmatrix} \text{“please” の出現頻度} \\ \text{“credit” の出現頻度} \\ \vdots \\ \text{“money” の出現頻度} \end{pmatrix}$$

文章のベクトル表現例



サポートベクトルマシン（SVM）

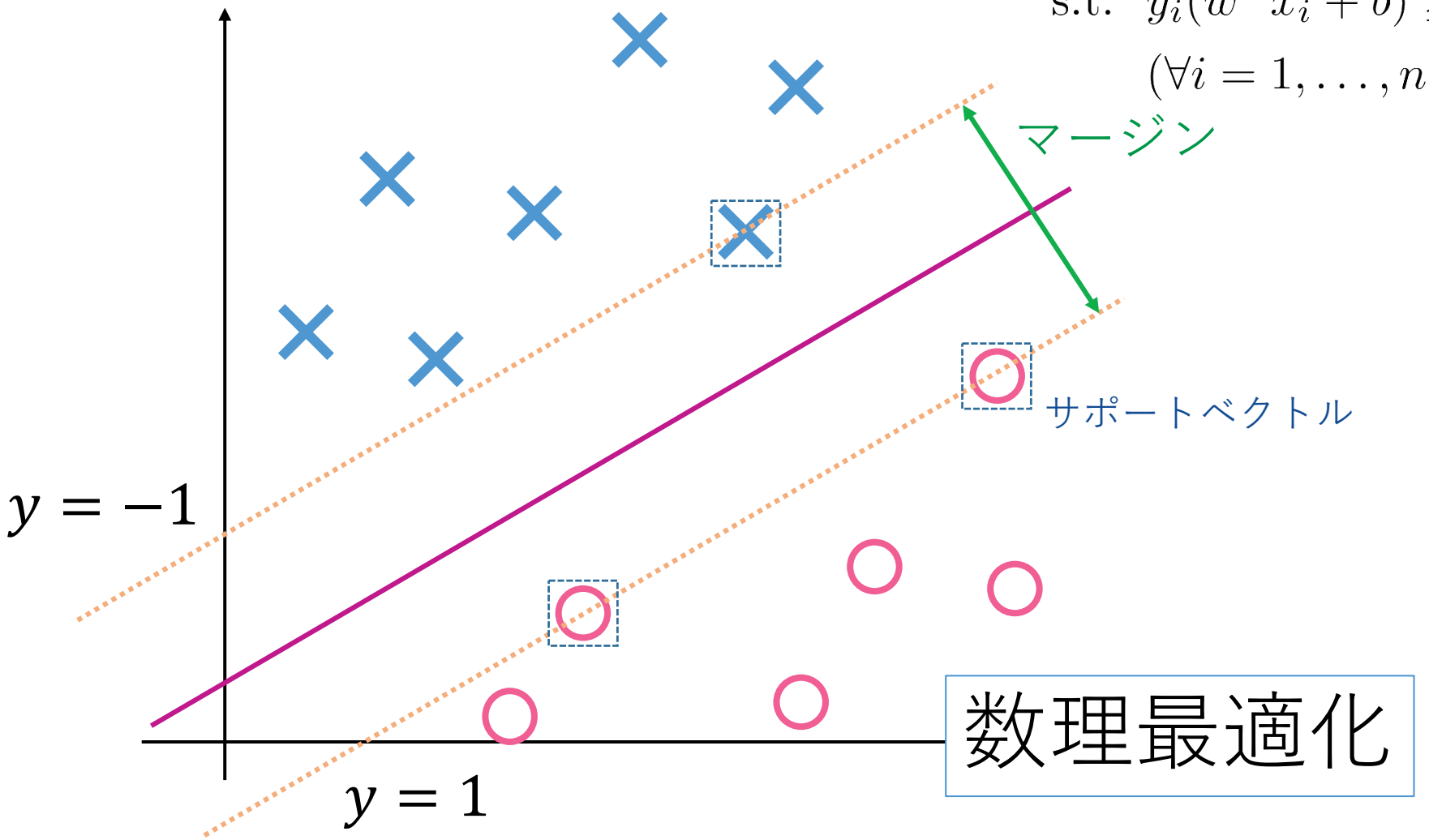
[Vapnik,63]

マージンを最大化

VC (Vapnik-Chervonenkis) 理論による正当化

$$\min_{w,b} \quad \frac{\|w\|^2}{2}$$

$$\text{s.t.} \quad y_i(w^\top x_i + b) \geq 1 \quad (\forall i = 1, \ldots, n)$$



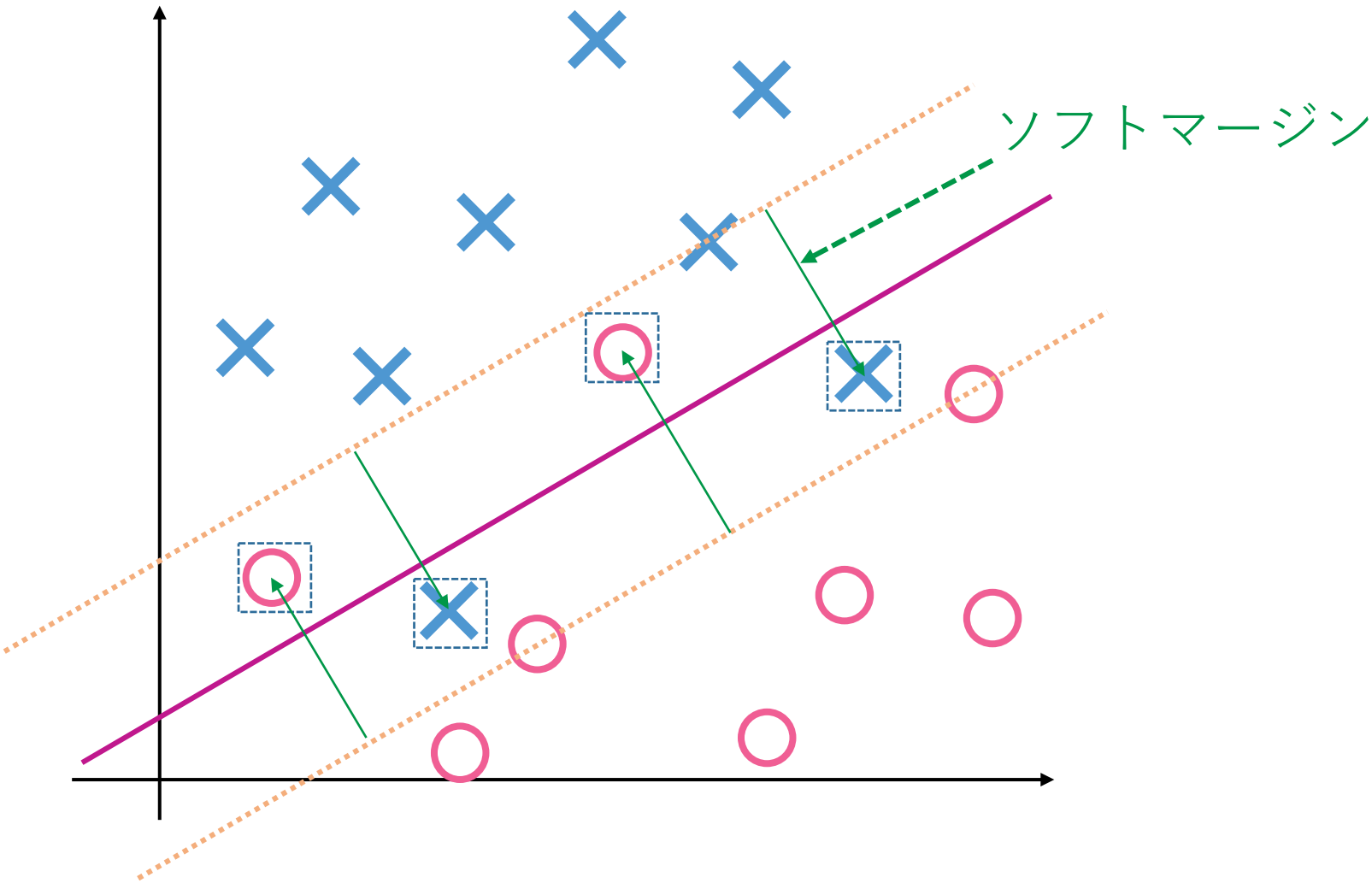
数理最適化

ソフトマージンSVM

[Cortes+Vapnik,95]

マージンを最大化
誤分類も許す

$$\min_{w,b} \sum_{i=1}^{n} \max\{1 - y_i(w^\top x_i + b), 0\} + C\frac{\|w\|^2}{2}$$



機械学習と人工知能の歴史

1946： ENIAC，高い計算能力
フォン・ノイマン「俺の次に頭の良い奴ができた」
1952： A. Samuelによるチェッカーズプログラム

統計的学習

1957：Perceptron，ニューラルネットワークの先駆け
第一次ニューラルネットワークブーム
1963：線形サポートベクトルマシン

線形モデルの限界

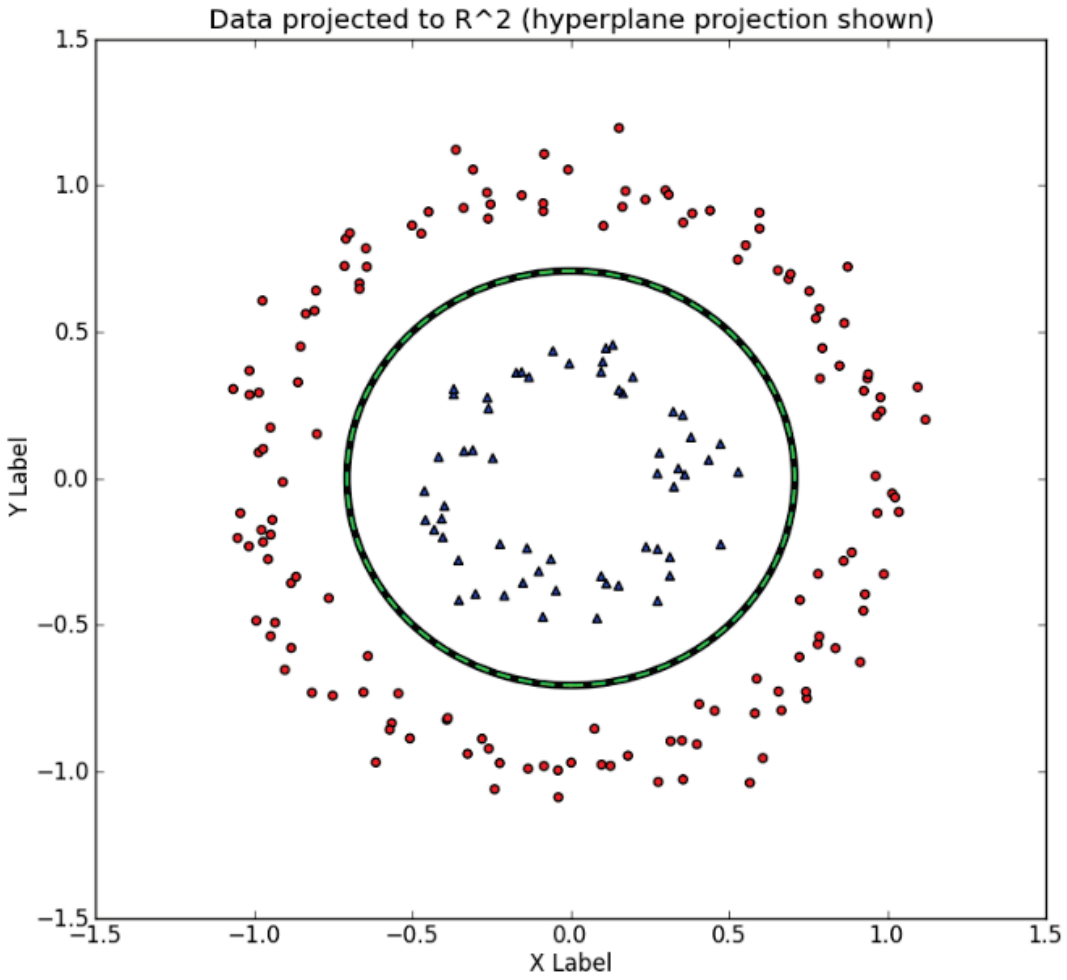
1980年代：多層パーセプトロン，誤差逆伝搬，
畳み込みネット
第二次ニューラルネットワークブーム

非凸性の問題

1992： 非線形サポートベクトルマシン
（カーネル法）
1996： スパース学習（Lasso）
2003： トピックモデル（LDA）
2012： Supervision（Alex-net）
第三次ニューラルネットワークブーム

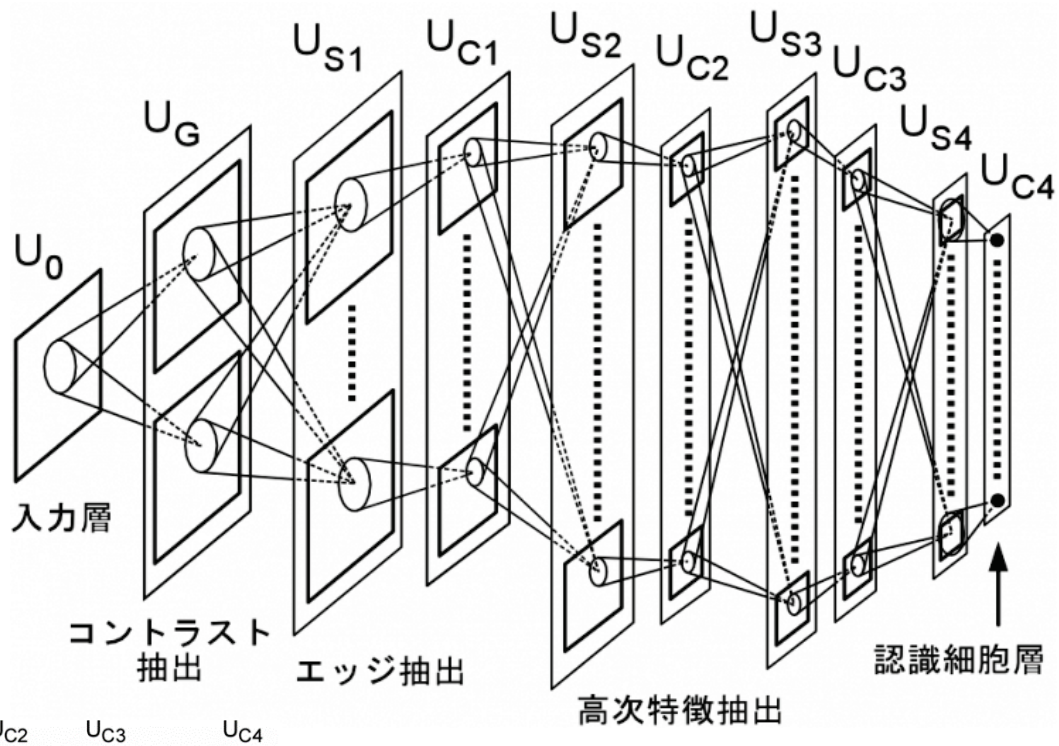
データの増加
＋計算機の強化

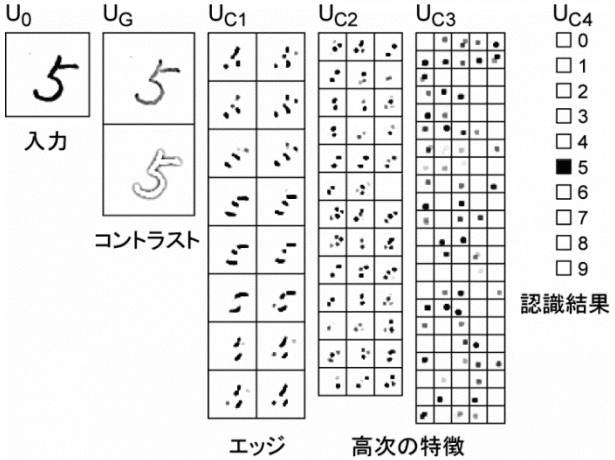
非線形判別

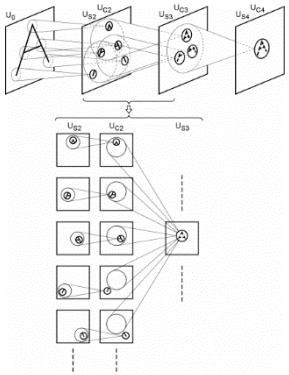

Data projected to R^2 (hyperplane projection shown)
Y Label
X Label
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

ネオコグニトロン

[福島,79]



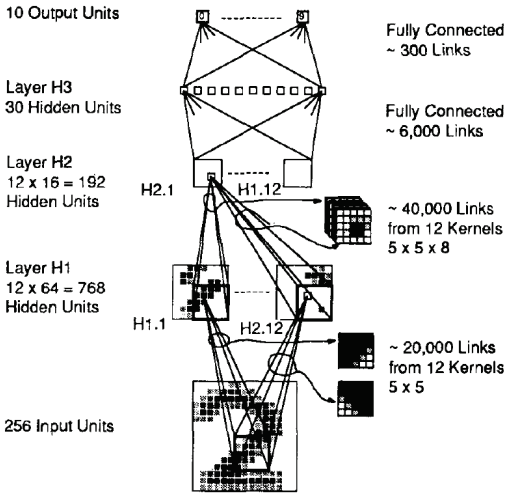




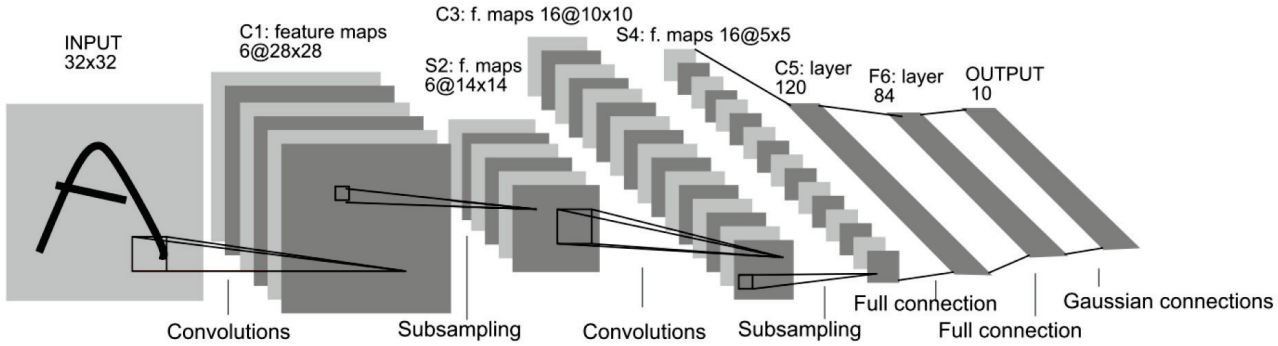
- 人間の脳を模倣
- 畳み込みネットの初期型
- 自己組織型学習
  →素子を足してゆく

LeNet

[LeCun+etal,89]



LeNet-5
[LeCun etal,98]



- 畳み込み＋プーリング：現在も使われている構造
- 誤差逆伝搬法でパラメータを更新
- 手書き文字認識データセット（MNIST）で９９％の精度を達成

機械学習と人工知能の歴史

1946： ENIAC，高い計算能力
フォン・ノイマン「俺の次に頭の良い奴ができた」
1952： A. Samuelによるチェッカーズプログラム

統計的学習

ルールベース

1960年代前半：
ELIZA（イライザ），
擬似心理療法士

1980年代：
エキスパートシステム

人手による学習ルールの作りこみの限界
「膨大な数の例外」

Siriなどにつながる

1957：Perceptron，ニューラルネットワークの先駆け
第一次ニューラルネットワークブーム
1963：線形サポートベクトルマシン

線形モデルの限界

1980年代：多層パーセプトロン，誤差逆伝搬，
畳み込みネット
第二次ニューラルネットワークブーム
1992： 非線形サポートベクトルマシン
（カーネル法）

非凸性の問題

1996： スパース学習（Lasso）
2003： トピックモデル（LDA）
2012： Supervision（Alex-net）
第三次ニューラルネットワークブーム

データの増加
＋計算機の強化

問題点

- 誰でも実装できるわけではなかった.

  e.g. 「LeNetはYan LeCunしか実装できない」といった噂
- 様々な職人芸的なノウハウが存在.
  - ➢パラメータのチューニング：学習率，層の数，層の幅
- 大域的最適解の計算が難しい.

  局所最適解しか得られない.

→これらは現代でも未解決

(実装に関してはライブラリの充実でかなり解決)

誰でも実装できて，最適解が一つの手法が欲しい.

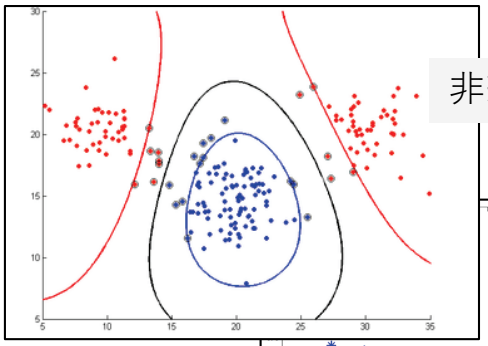
→　カーネルを用いたサポートベクトルマシン

カーネル法

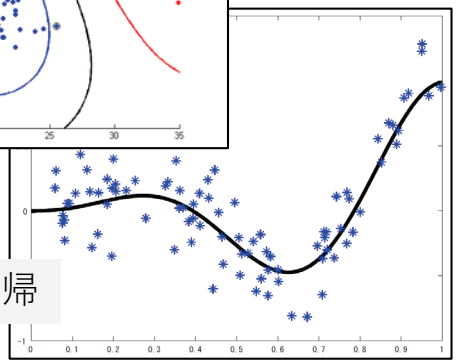
$$\min_{\alpha_i, b} \sum_{i=1}^{n} \max\left\{1 - y_i\left(\sum_{j=1}^{n} k(x_j, x_i)\alpha_j + b\right), 0\right\} + C\sum_{i,j=1}^{n} \alpha_i \alpha_j k(x_i, x_j)$$

カーネルトリック

$$k(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$



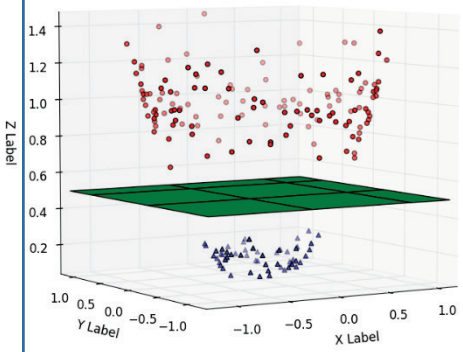
非線形分類

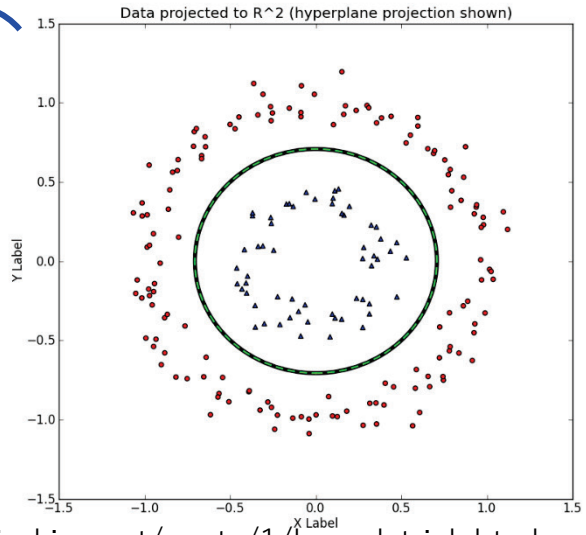


非線形回帰

http://wiki.eigenvector.com/index.php?title=Svmda

非線形写像 ϕ





http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

関数解析：再生核ヒルベルト空間の理論

- **凸最適化問題**で解ける.
  - ✓ 効率的な最適化手法が存在.
  - ✓ 解は一つ．誰が解いても同じ答えが返ってくる.
- **VC理論・経験過程の理論**による汎化誤差の保証.

$$\|\hat{f} - f_0\|_{L_2}^2 \leq O_p(n^{-\frac{1}{1+s}})$$

Vladimir Vapnik

９０年代以降

データ解析としての機械学習

統計学やデータマイニングとの融合

- 高次元スパース学習
- ベイズモデリング
- オンライン学習，確率的最適化

　→　ビッグデータ解析への活用

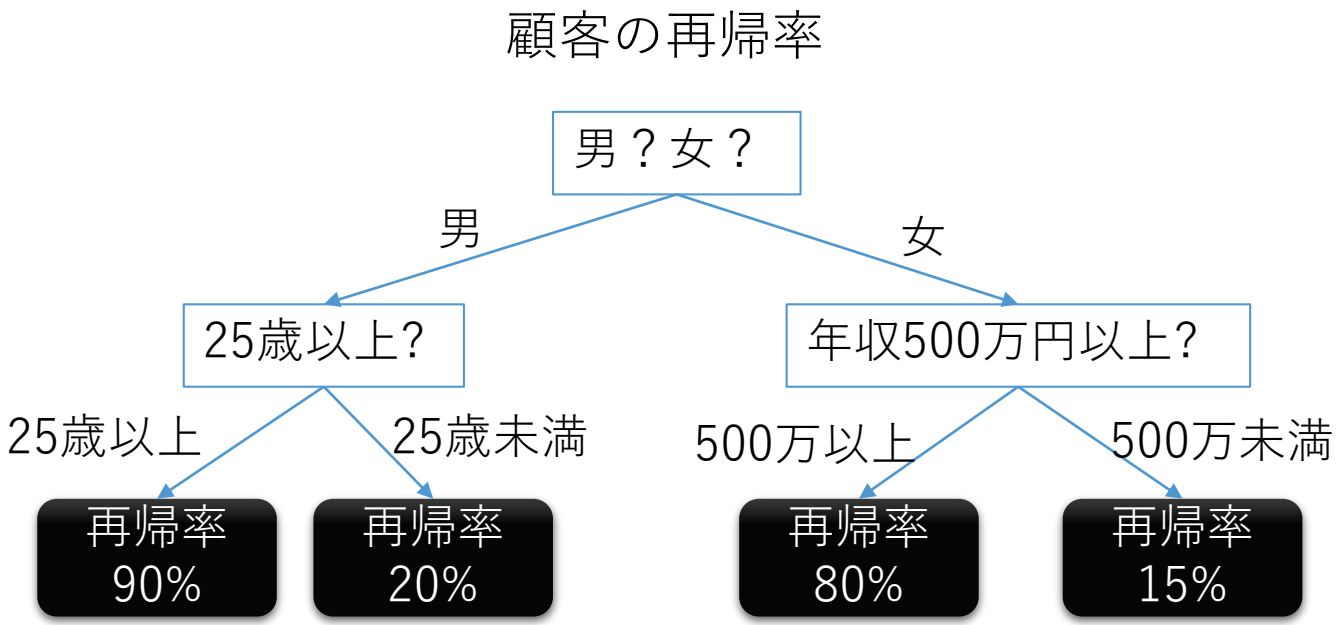
メディアに出る「人工知能」はここに属することが多い

- データの増加＋計算機の強化

　→深層学習再興，**第三次ニューラルネットワークブーム**へ

決定木

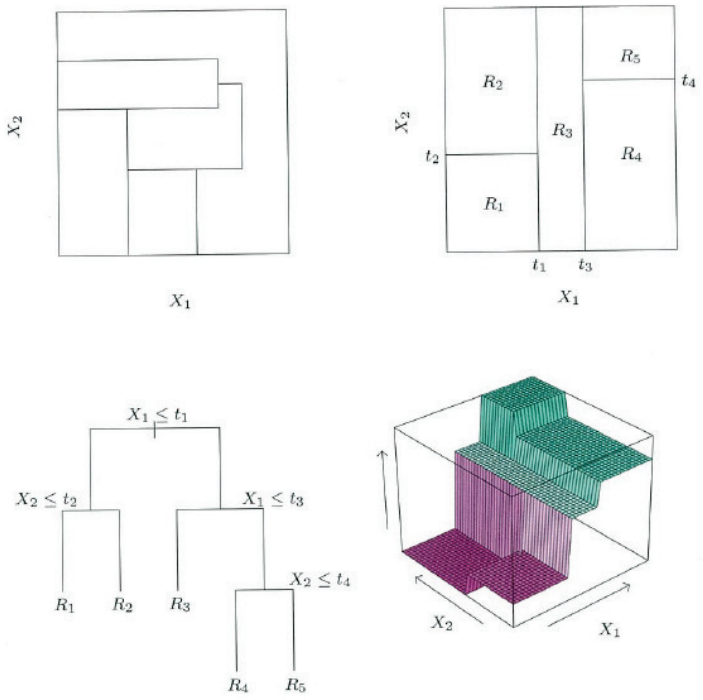
- 解釈可能性が高い
- 決定木を組み合わせた勾配ブースティングはデータマイニング系コンペティションで常連
  (判別だけでなく回帰にも利用可能)



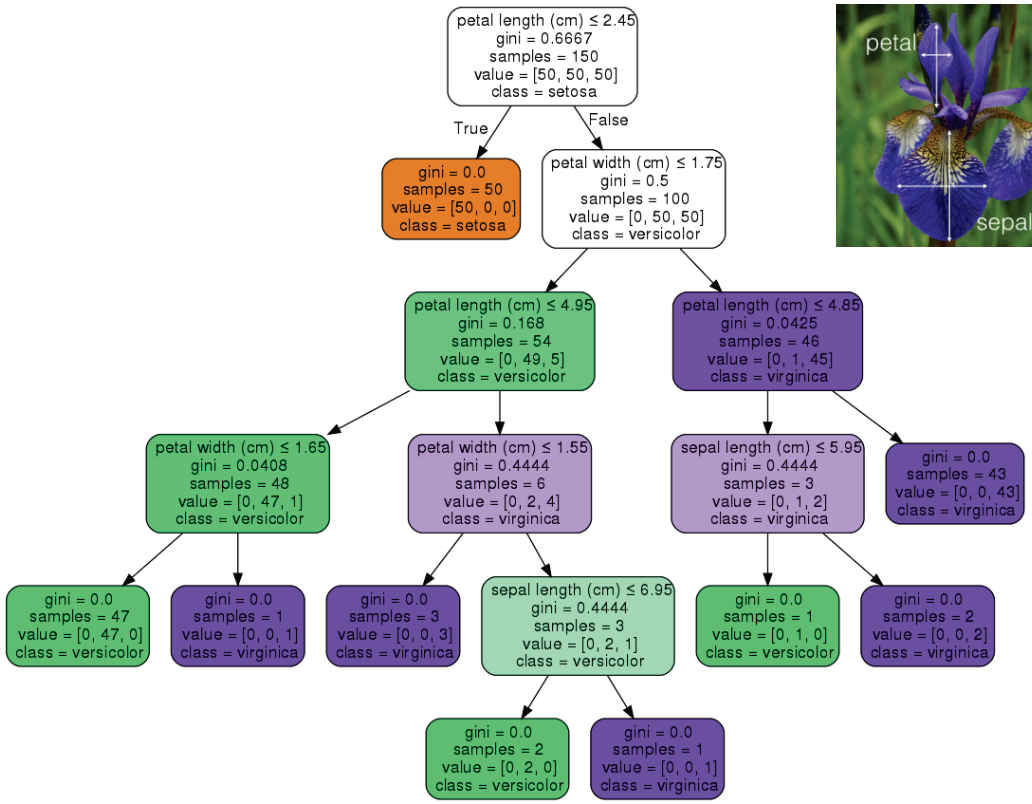
- 学習された決定木から要因を把握しやすい.
- 分析結果から対策を立てるのに有用.

決定木

決定木の様子
2次元の説明変数



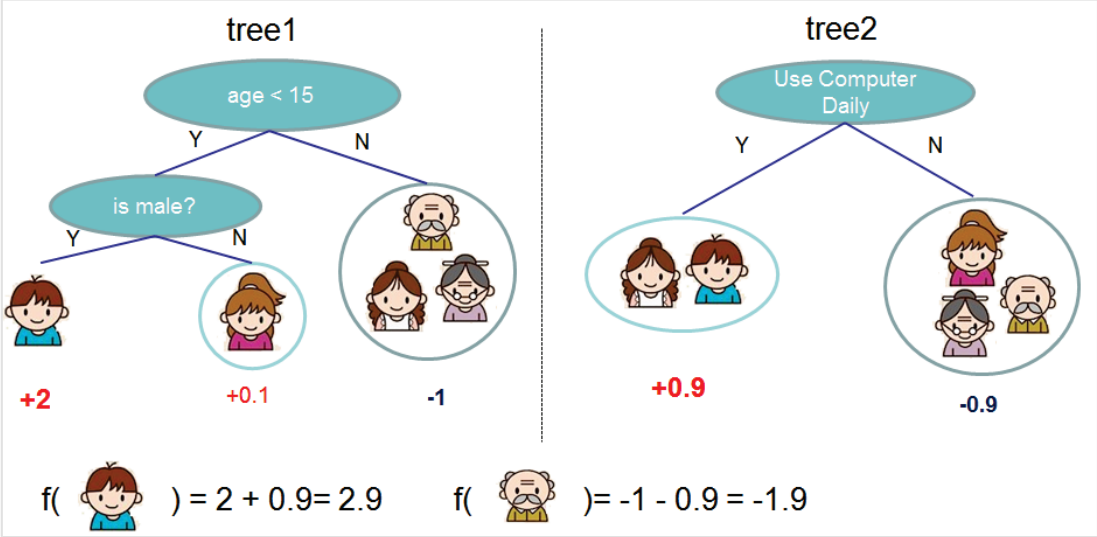
図はHastie, Tibshirani, Friedman: The Elements of Statistical Learning, Springer, 2001.より



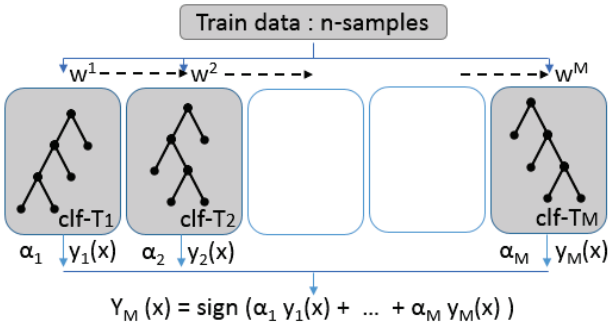
Python scikit-learnによるiris(アヤメ)データ分類

勾配ブースティング

- XGBoostやLightGBMが有名
- 「決定木の和」で強力な判別を実現
  - 決定木一つでは複雑な判別が難しい→ 複数用意してその和(多数決)を取る
  - 和の取り方に勾配ブースティングと呼ばれる技法を使用



「コンピュータゲームが好きか？」を判別



沢山の決定木の**多数決**を出力

[Chen, Guestrin: XGBoost: A Scalable Tree Boosting System. KDD2016.]

[Ke, Meng, Finley, Wang, Chen, Ma, Ye, Liu: LightGBM: A Highly Efficient Gradient Boosting Decision Tree. NIPS2017.]

勾配ブースティング

XGBoostは各種データ解析コンペティションで好成績

Machine Learning Challenge Winning Solutions

XGBoost is extensively used by machine learning practitioners to create state of art data science solutions, this is a list of machine learning winning solutions with XGBoost. Please send pull requests if you find ones that are missing here.

- Maksims Volkovs, Guangwei Yu and Tomi Poutanen, 1st place of the 2017 ACM RecSys challenge. Link to paper.
- Vlad Sandulescu, Mihai Chiru, 1st place of the KDD Cup 2016 competition. Link to the arxiv paper.
- Marios Michailidis, Mathias Müller and HJ van Veen, 1st place of the Dato Truely Native? competition. Link to the Kaggle interview.
- Vlad Mironov, Alexander Guschin, 1st place of the CERN LHCb experiment Flavour of Physics competition. Link to the Kaggle interview.
- Josef Slavicek, 3rd place of the CERN LHCb experiment Flavour of Physics competition. Link to the Kaggle interview.
- Mario Filho, Josef Feigl, Lucas, Gilberto, 1st place of the Caterpillar Tube Pricing competition. Link to the Kaggle interview.
- Qingchen Wang, 1st place of the Liberty Mutual Property Inspection. Link to the Kaggle interview.
- Chenglong Chen, 1st place of the Crowdflower Search Results Relevance. Link to the winning solution.
- Alexandre Barachant (“Cat”) and Rafał Cycoń (“Dog”), 1st place of the Grasp-and-Lift EEG Detection. Link to the Kaggle interview.
- Halla Yang, 2nd place of the Recruit Coupon Purchase Prediction Challenge. Link to the Kaggle interview.
- Owen Zhang, 1st place of the Avito Context Ad Clicks competition. Link to the Kaggle interview.
- Keiichi Kuroyanagi, 2nd place of the Airbnb New User Bookings. Link to the Kaggle interview.
- Marios Michailidis, Mathias Müller and Ning Situ, 1st place Homesite Quote Conversion. Link to the Kaggle interview.

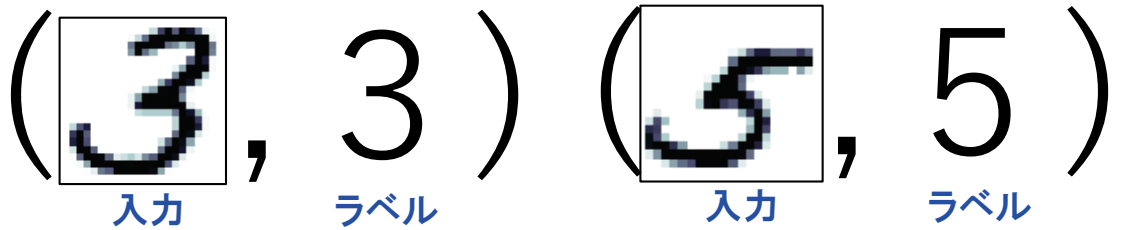
機械学習の数理

機械学習の問題設定

教師あり学習：

データ：(x,y)←ある入力xとそれに対するラベルyの組

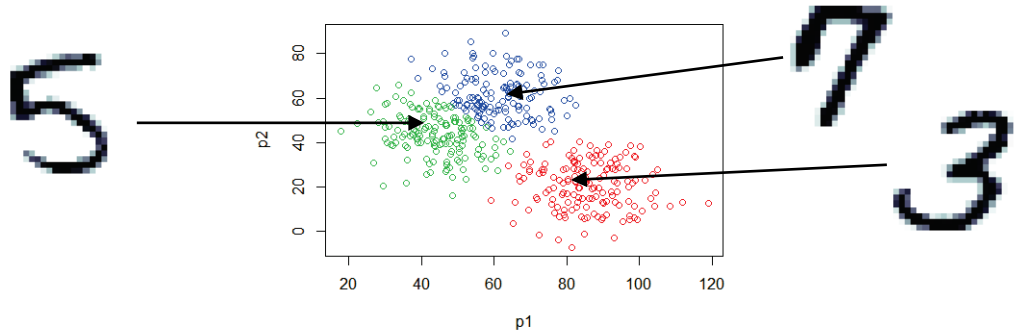
問題の例：回帰，判別



教師なし学習：

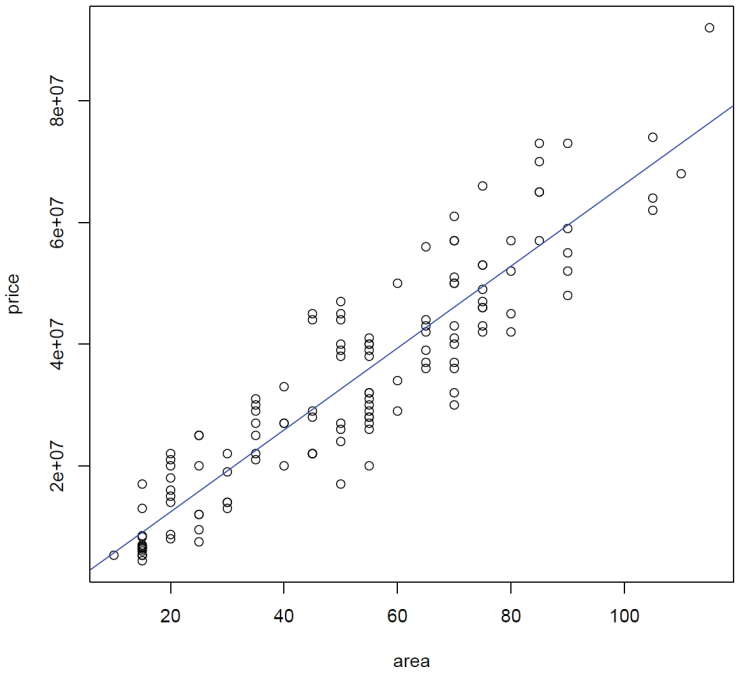
データ：(x) ←ラベルがない

問題の例：クラスタリング，音源分離，異常検知

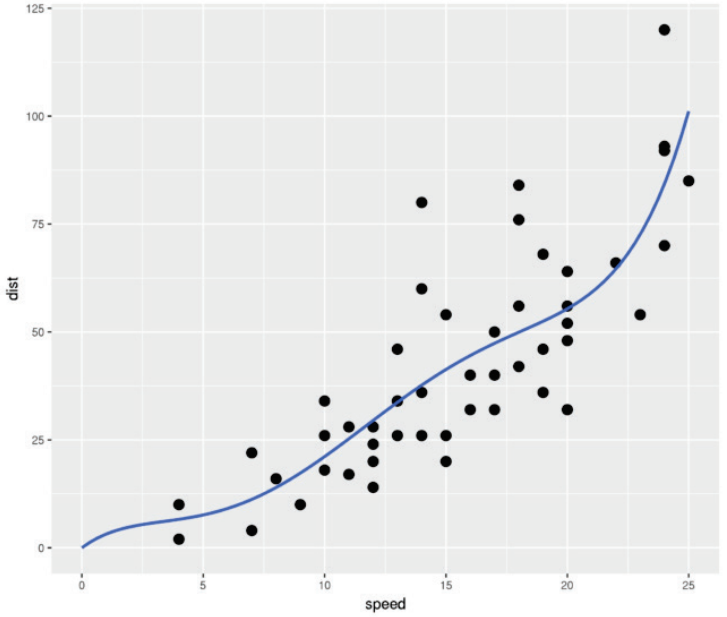


半教師有り学習：ラベルの付いているデータと付いてないデータが混在

回帰



線形

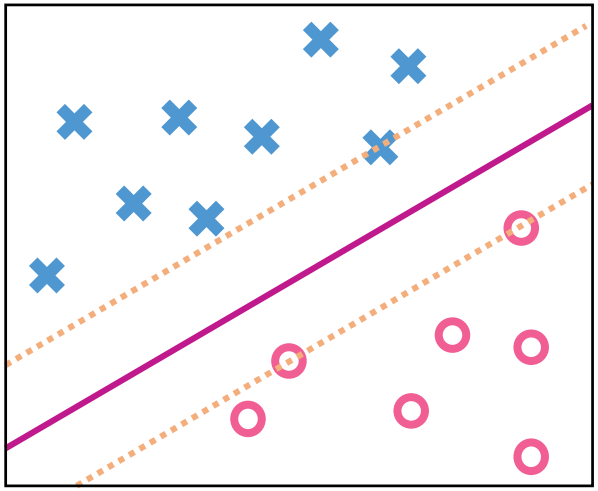


非線形

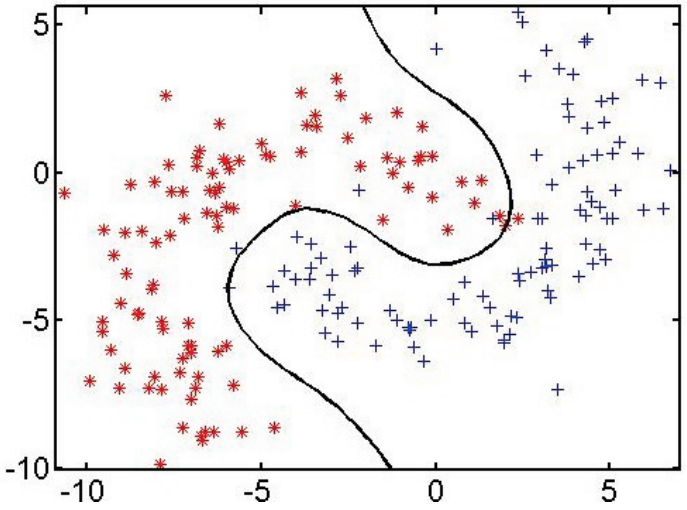
入力xから実数の出力yを予測

- 線形
- 非線形

判別



線形

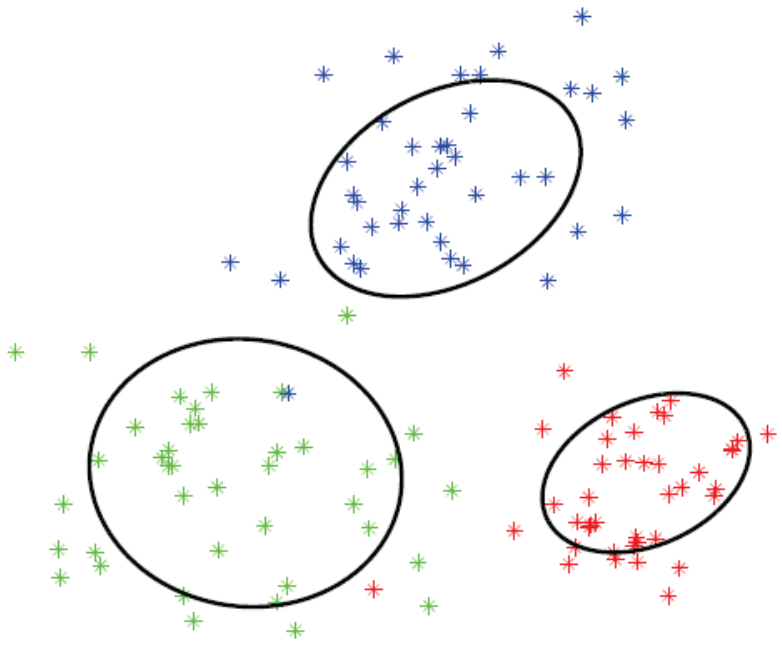


非線形

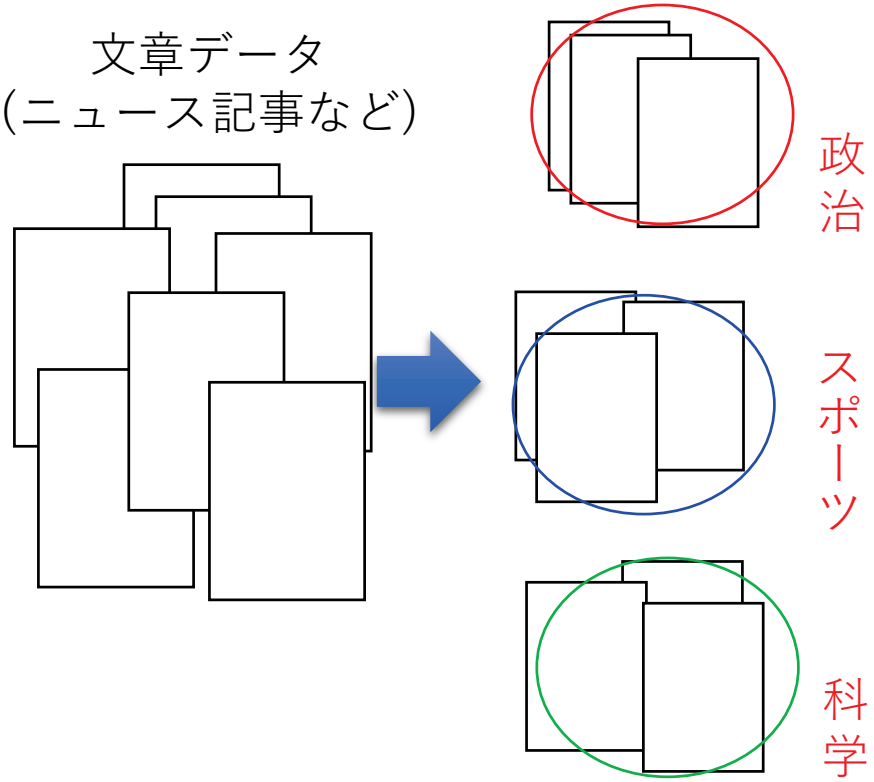
入力xからカテゴリーの出力yを予測

- 線形
- 非線形

クラスタリング



混合ガウス分布によるクラスタリング

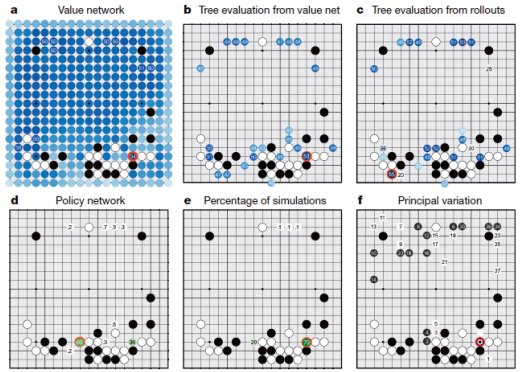


トピックモデル

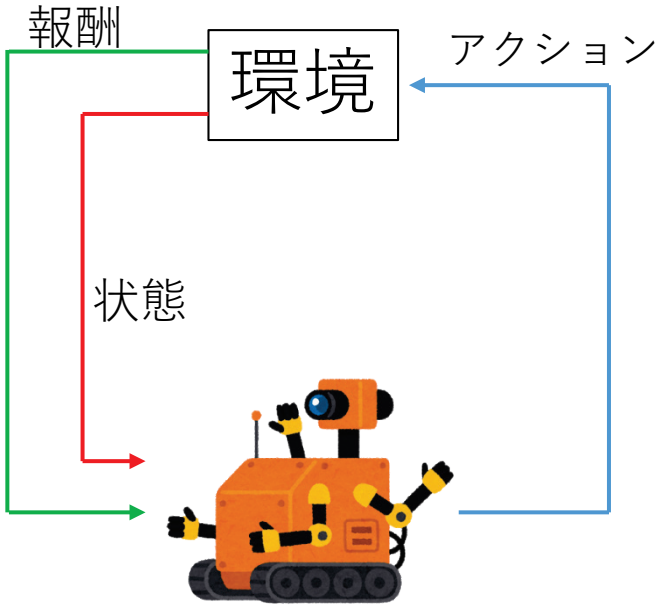
- 文章分類

強化学習

Google research blog, 8/March/2016.
“Deep Learning for Robots: Learning from Large-Scale Interaction.”

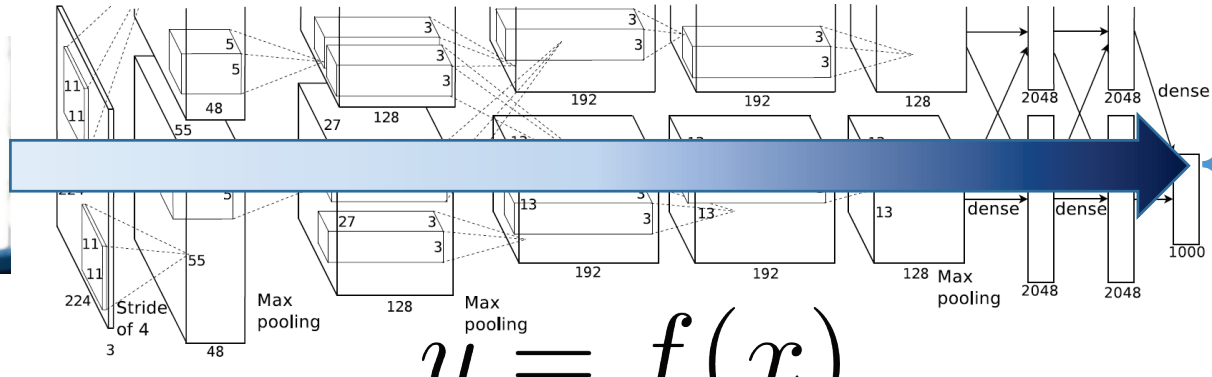


[Silver et al. (Google Deep Mind): Mastering the game of Go with deep neural networks and tree search,
Nature, 529, 484—489, 2016]



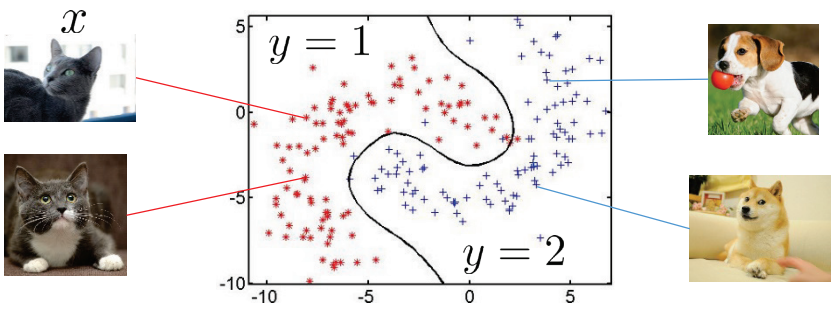
教師あり学習

画像



-猫　(y=1)
-犬　(y=2)
-人間 (y=3)

$$x \qquad y = f(x) \qquad y$$



学習：「関数」をデータに当てはめる
モデル：関数の集合（例：深層NNの表せる関数の集合）

予測モデルの学習

問題ごと　　共有化可能

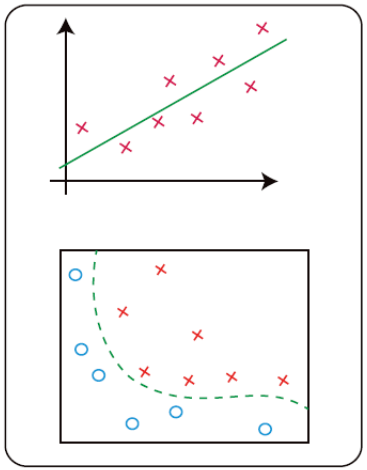
特徴抽出

特徴ベクトル

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix}$$

分析

パラメータ推定



判別・回帰など

学習規準：汎化誤差最小化

分野ごとに様々なノウハウ
深層学習（Deep learning）
自動的に特徴量を抽出

予測モデルの構築

$$y = f(x;\theta)$$

モデルの
パラメータ

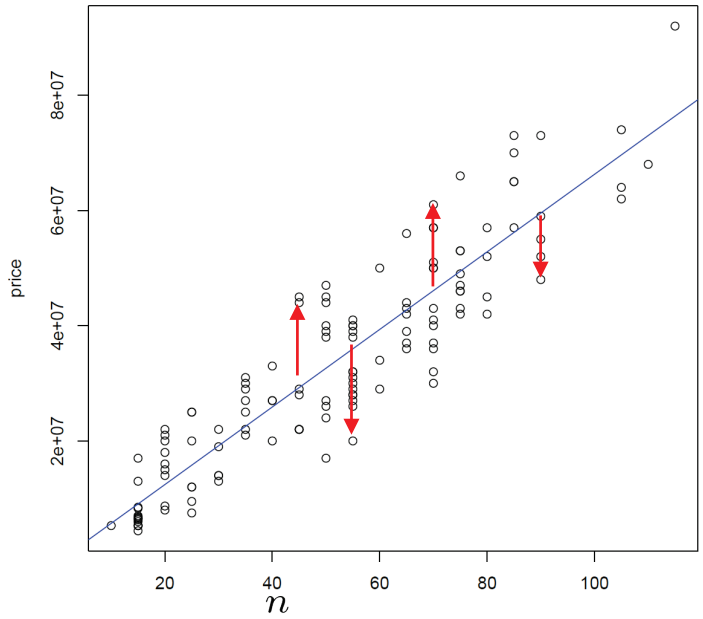
一度特徴ベクトルに変換してしまえばあとは統計の問題.
→ 汎用的な手法（機械学習）を適用できる.

線形モデル

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \beta_0 + \epsilon$$

y:従属変数, x:特徴ベクトル

マンション価格=$\beta_1 \times$ 床面積 $+ \beta_2 \times$ 築年数 $+ \beta_3$+(揺らぎ)



最小二乗法 $\min_{\beta_0,\beta_1,\beta_2,\beta_3} \sum_{i=1} (y_i - \beta_1 x_{i,1} - \beta_2 x_{i,2} - \beta_3 x_{i,3} - \beta_0)^2$

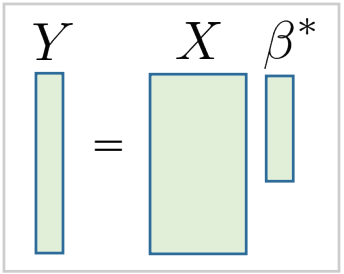
最小二乗法

n個の観測値（サンプル）：　$(y_i, \boldsymbol{x}_i) \in \mathbb{R} \times \mathbb{R}^d \ (i = 1, \ldots, n)$

$$Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} \in \mathbb{R}^n, \ X = \begin{bmatrix} \boldsymbol{x}_1^\top & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_n^\top & 1 \end{bmatrix} \in \mathbb{R}^{n \times (d+1)}, \ \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \in \mathbb{R}^n$$

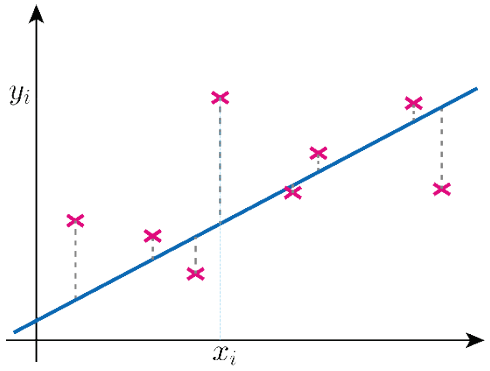
β^*を真の回帰係数（これを推定したい）とすると，

$$Y = X\beta^* + \boldsymbol{\epsilon}$$



最小二乗推定量 (最尤推定量)：

$$\begin{aligned} \hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{d+1}} \sum_{i=1}^{n} (y_i - [\boldsymbol{x}_i^\top 1]\boldsymbol{\beta})^2 \\ &= \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^{d+1}} \|Y - X\boldsymbol{\beta}\|^2 \\ &= (X^\top X)^{-1} X^\top Y \end{aligned}$$



訓練誤差と汎化誤差

パラメータ θ：データの構造を表す変数（例：判別平面）
損失関数 $\ell(Y, f(X, \theta))$：パラメータ θ がデータをどれだけ説明しているか

汎化誤差：損失の期待値

$$\mathrm{E}[\ell(Y, f(X, \theta))]$$

本当は最小化したいもの.

訓練誤差：有限個のデータで代用

$$\frac{1}{n}\sum_{i=1}^{n}\ell(y_i, f(x_i, \theta))$$

代わりに最小化するもの.

※クラスタリング等，教師なし学習も尤度を使ってこのように書ける.

この二つには大きなギャップがある.
[過学習]

基本的な考え方

- パラメータをθとしたときに，観測されたデータが観測される確率（尤度）

尤度　$\prod_{i=1}^{n} p(z_i|\theta)$　：確率モデル

尤度が高ければ，観測データが観測される確率が高い→「尤もらしい」

負の対数尤度　$\sum_{i=1}^{n} \underbrace{-\log(p(z_i|\theta))}_{\ell(z_i,\theta)}$

→最小化で観測データを良く表現するパラメータが得られる．

「最尤推定」

（ベイズ推定も重要だがここでは割愛）

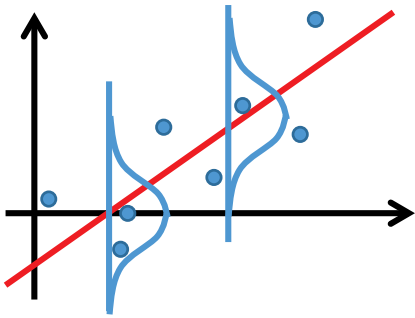
線形回帰

$$p(y_i|x_i,\theta) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y_i - x_i^\top\theta)^2}{2}\right)$$

正規分布
平均$x_i^\top\theta$, 分散1

$$-\log(p(y_i|x_i,\theta)) = \frac{(y_i - x_i^\top\theta)^2}{2} + C$$

→最小二乗法



KL-divergence

$$D(p||q) = \int p(z) \log \left(\frac{p(z)}{q(z)} \right) \mathrm{d}z$$

真の分布　モデルの分布

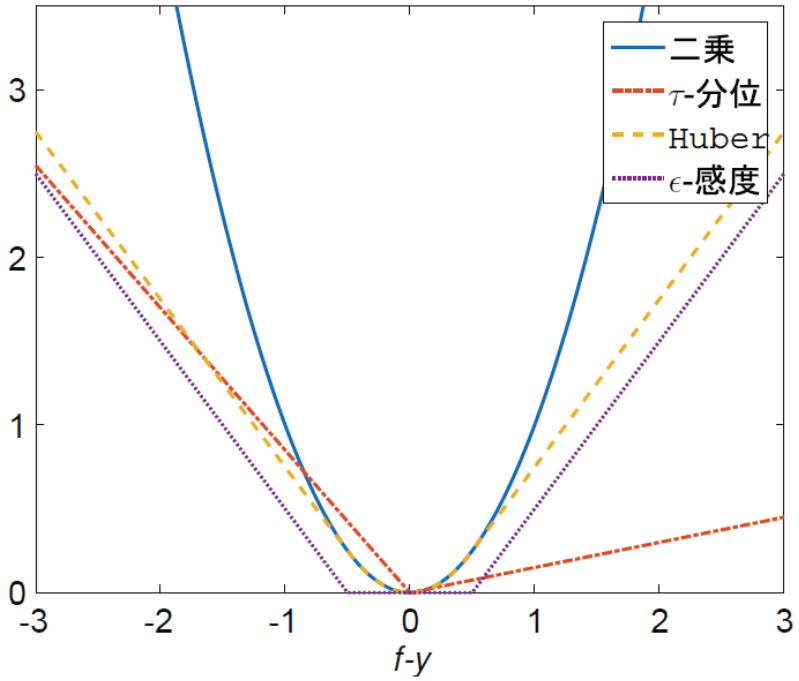
$$D(p^*||p_\theta) = \underbrace{\int -p^*(z) \log \left(p_\theta(z) \right) \mathrm{d}z}_{\text{サンプル平均で代用}} + \int p^*(z) \log \left(p^*(z) \right) \mathrm{d}z$$

$$-\frac{1}{n} \sum_{i=1}^{n} \log \left(p_\theta(z_i) \right)$$

対数尤度最大化はKL-divergence最小化の近似ともみなせる

回帰の損失関数

- 二乗損失: $\ell(y, f) = \frac{1}{2}(y - f)^2$.
- τ-分位点損失: $\ell(y, f) = (1 - \tau)\max\{f - y, 0\} + \tau \max\{y - f, 0\}$.
  ただし，$\tau \in (0, 1)$. 分位点回帰に用いられる.
- ϵ-感度損失: $\ell(y, f) = \max\{|y - f| - \epsilon, 0\}$,
  ただし，$\epsilon > 0$. サポートベクトル回帰に用いられる.



※各損失関数は必ずしも確率モデルと対応するわけではない

判別

- 判別

損失関数

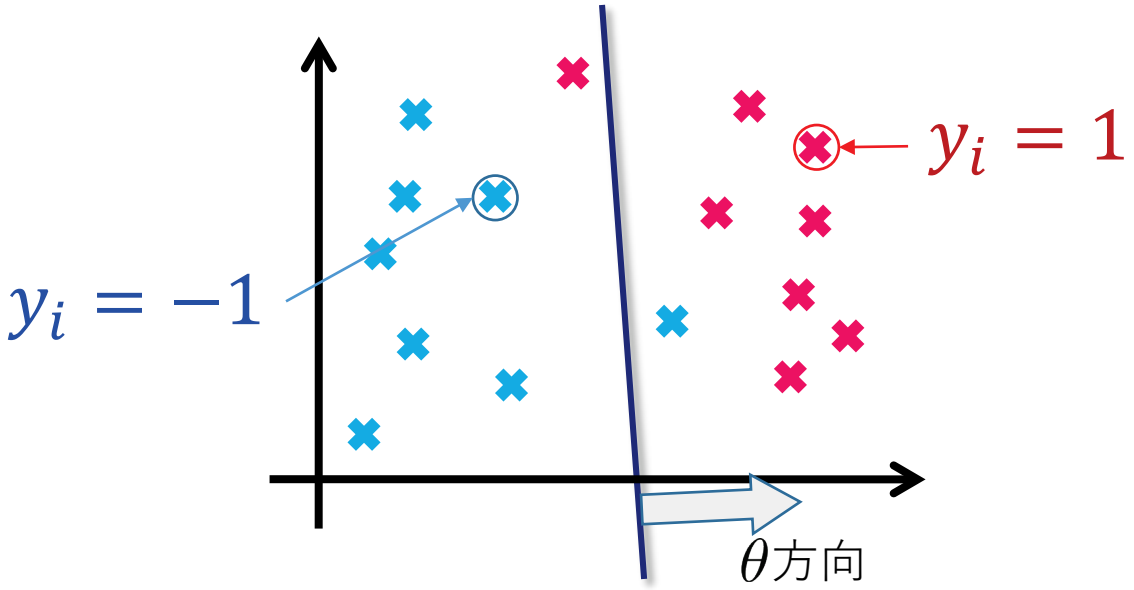
$$\ell(y, x^\top \theta) = \log(1 + \exp(-yx^\top \theta))$$

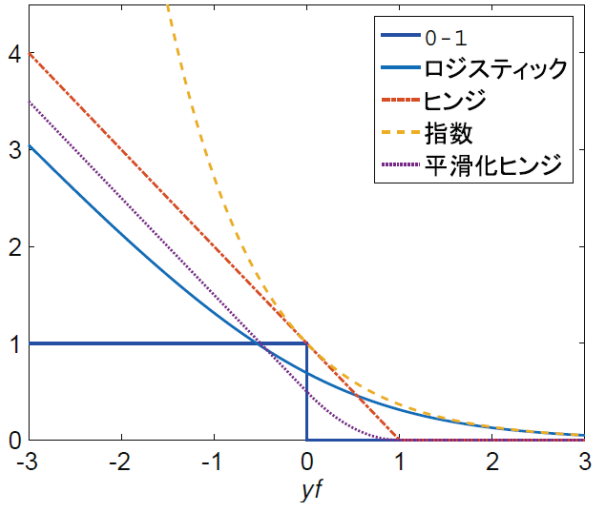
（ロジスティック損失）

訓練誤差最小化

$$\min_{\theta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, x_i^\top \theta) = \min_{\theta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \log(1 + \exp(-y_i x_i^\top \theta))$$

（ロジスティック回帰）



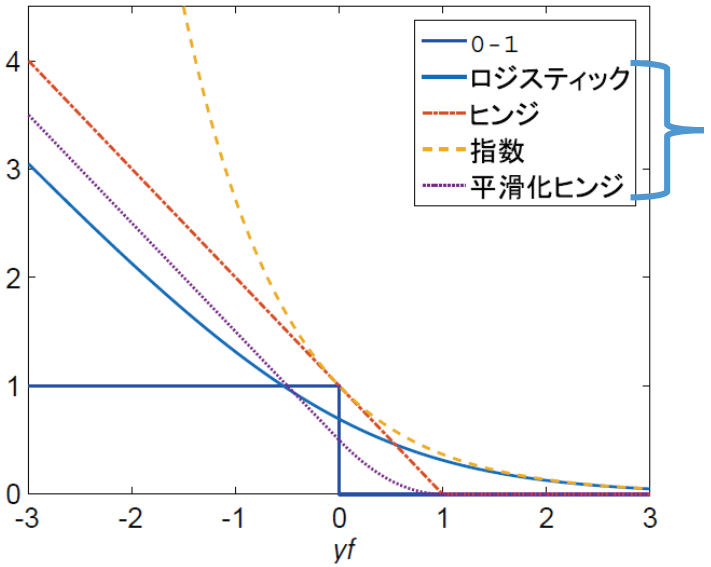


判別の損失関数

$y \in \{\pm 1\}$

- ロジスティック損失: $\ell(y, f) = \log((1 + \exp(-yf))/2).$
- ヒンジ損失: $\ell(y, f) = \max\{1 - yf, 0\}.$
- 指数損失: $\ell(y, f) = \exp(-yf).$
- 平滑化ヒンジ損失:

$$\ell(y, f) = \begin{cases} 0, & (yf \geq 1), \\ \frac{1}{2} - yf, & (yf < 0), \\ \frac{1}{2}(1 - yf)^2, & (\text{otherwise}). \end{cases}$$

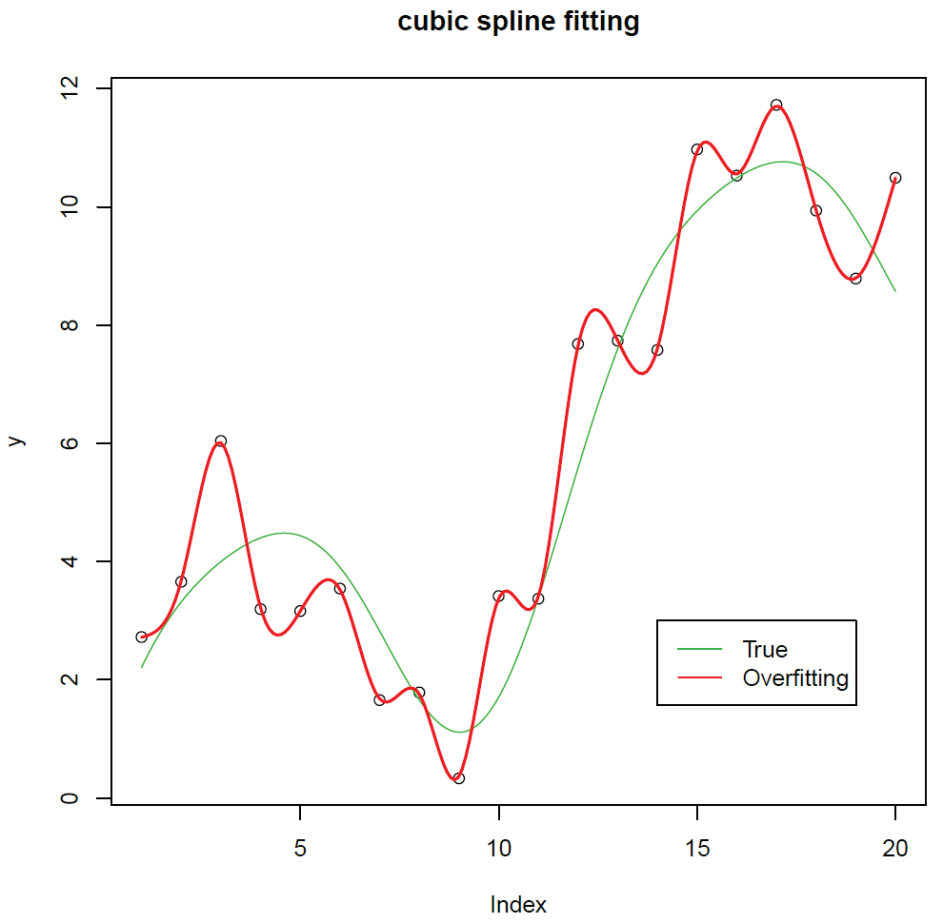


凸代理損失

過学習

複雑なモデル（例えば深層ニューラルネット）を用いるのが常に良い選択か？

→ そうとは限らない．「**過学習**」に注意する必要あり．



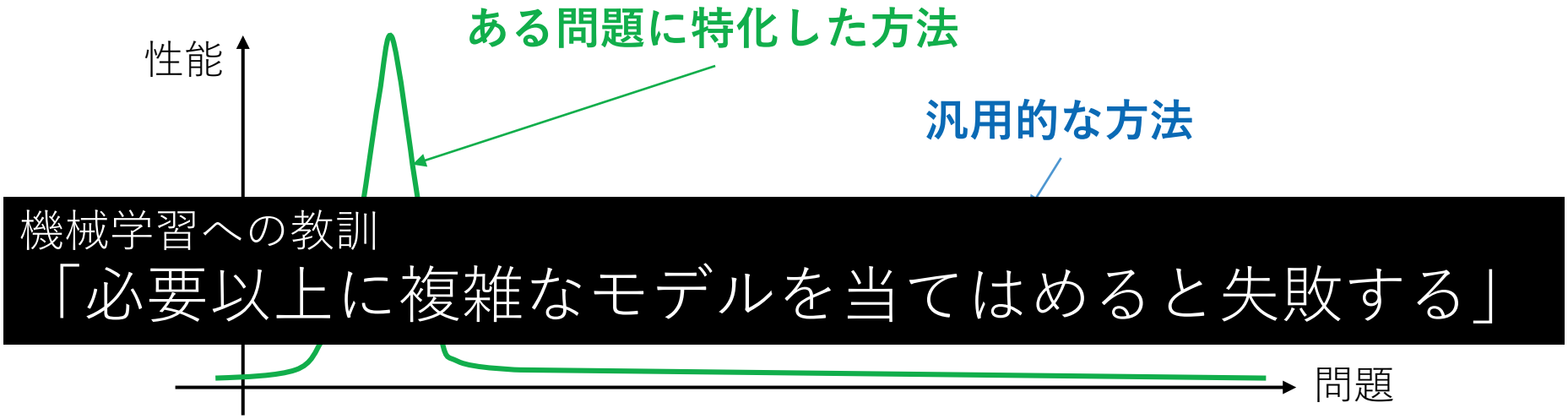
学習機の複雑さと学習能力

- オッカムの剃刀

「ある事柄を説明するためには、必要以上に多くを仮定するべきでない」とする指針

- No free lunch theorem

「あらゆる問題で性能の良い汎用的学習機は実現不可能であり，ある問題に特殊化された手法に勝てない」

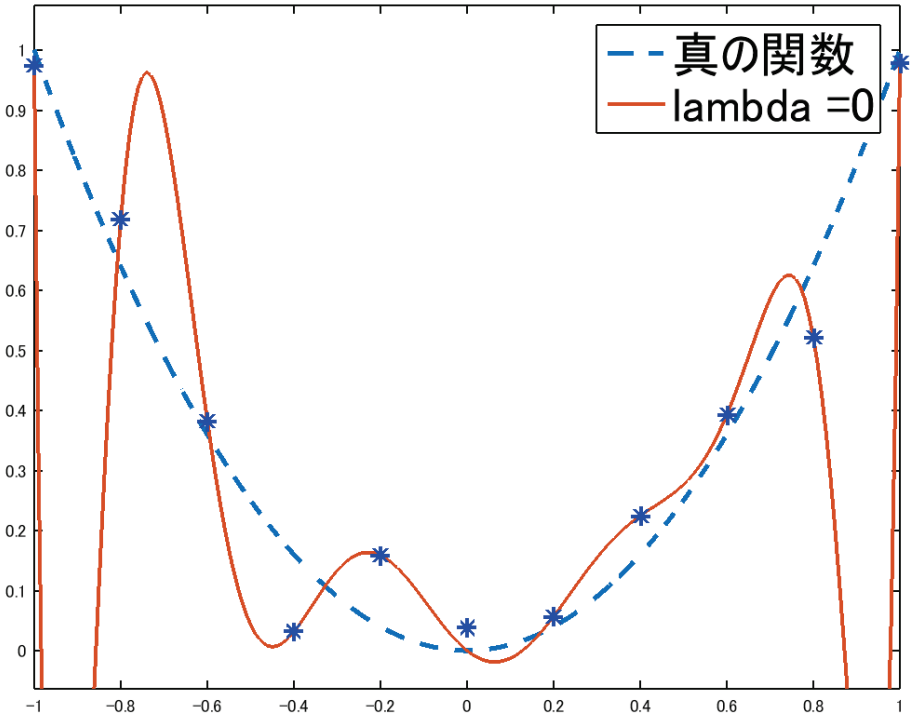


William of Ockham：1285-1347．スコラ学の神学者，哲学者．

No free lunch theorem: [D.H.Wolpert and W.G. Macready: 1995,1997][Y.C. Ho and D.L. Pepyne: 2002]

多項式回帰の過学習

$$\min_{\beta \in \mathbb{R}^{15}} \sum_{i=1}^{n} \{y_i - (\beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_{15} x_i^{15})\}^2$$



正則化学習法

正則化：データに合った単純なモデルを当てはめる

→ 過学習を回避

正則化訓練誤差最小化

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 + \lambda \psi(\beta)$$

手元にあるデータへの当てはまり　　正則化項

複雑さへの罰則

代表的な例：リッジ正則化（L2ノルム）

$$\psi(\theta) = \|\theta\|_2^2 = \sum_{j=1}^{d} \theta_j^2$$

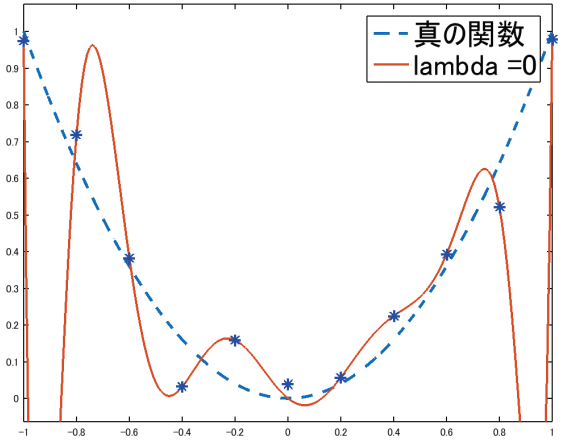
正則化の代表例

多項式回帰（１５次多項式，リッジ回帰）

$$\min_{\beta \in \mathbb{R}^{15}} \sum_{i=1}^{n} \{y_i - (\beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_{15} x_i^{15})\}^2 + \lambda \|\beta\|_2^2$$

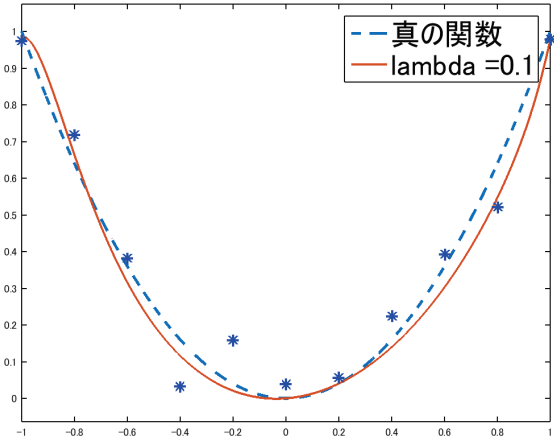
リッジ正則化と言う

手元のデータには良くあてはまるが真の関数からは遠い



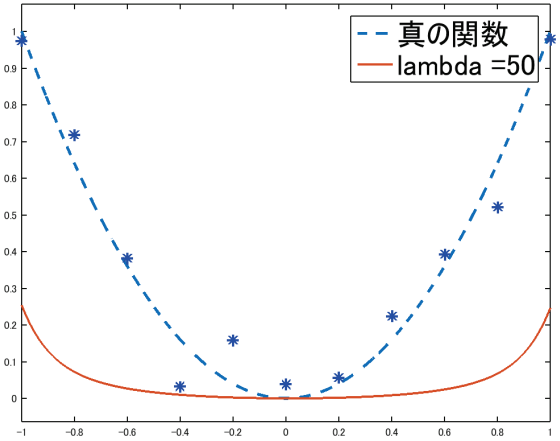
$\lambda = 0$

過学習



$\lambda = 0.1$

良い推定



$\lambda = 50$

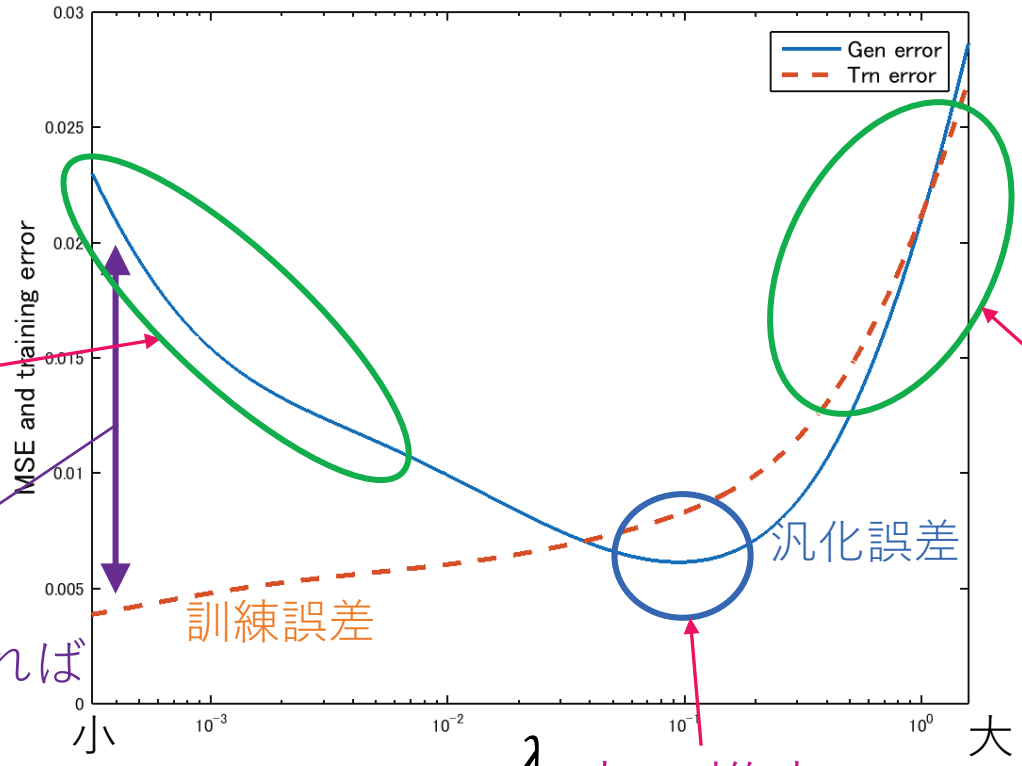
過小学習

正則化によってあまり複雑にならないよう制御がかかる

正則化の強さと汎化誤差の関係

多項式回帰（１５次多項式，リッジ回帰）

$$\min_{\beta \in \mathbb{R}^{15}} \sum_{i=1}^{n} \{y_i - (\beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_{15} x_i^{15})\}^2 + \lambda \|\beta\|_2^2$$



過学習

訓練誤差が小さければ
よいわけではない

訓練誤差

汎化誤差

過小学習

良い推定

横軸：正則化パラメータ(log-スケール)．　縦軸：汎化誤差（青），訓練誤差（赤）．

適切な λ を選ぶ方法→交差検証法，Mallows' Cp

バイアスとバリアンスの分解

線形モデル（ノイズは平均0，分散 σ^2 ）：

$$Y = X\beta^* + \epsilon$$

任意の推定量に対して以下の分解が成り立つ：

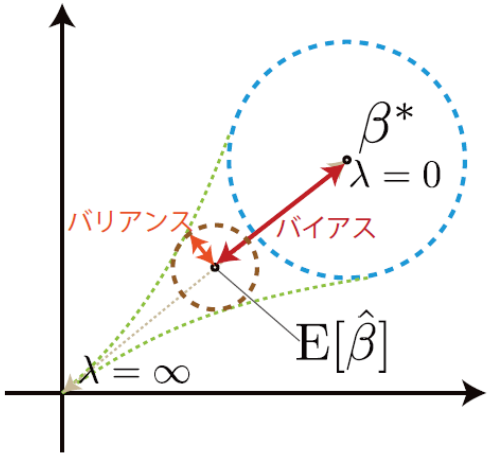
$$\mathrm{E}[\|\hat{\beta} - \beta^*\|^2] = \underbrace{\mathrm{E}[\|\mathrm{E}(\hat{\beta}) - \beta^*\|^2]}_{\text{バイアス項}} + \underbrace{\mathrm{E}[\|\hat{\beta} - \mathrm{E}(\hat{\beta})\|^2]}_{\text{バリアンス項}}$$

リッジ正則化の場合：$\hat{\beta}_{(\lambda)} = \underset{\beta \in \mathbb{R}^p}{\mathrm{argmin}} \left\{ \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|^2 \right\}$

$$= \underset{\beta \in \mathbb{R}^p}{\mathrm{argmin}} \{ \|Y - X\beta\|^2 + \lambda \|\beta\|^2 \}$$

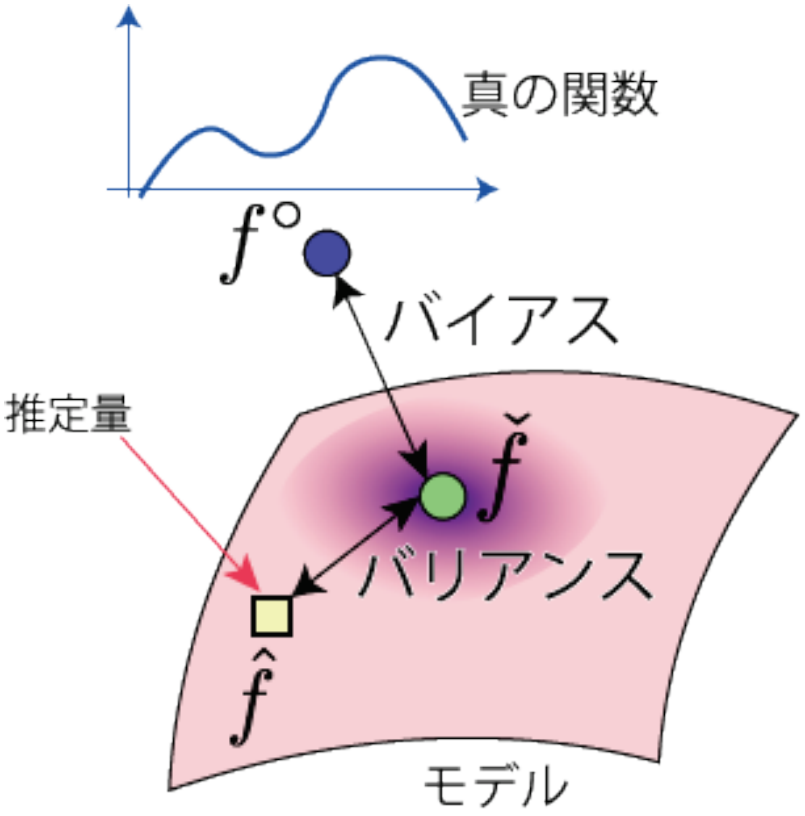
正則化パラメータとバイアス-バリアンス

	バイアス	バリアンス
$\lambda = 0$	0	$\sigma^2 \mathrm{Tr}[(X^\top X)^{-1}]$
$\lambda = \infty$	$\|\beta^*\|^2$	0



※両方を同時に小さくすることはできない．

バイアスとバリアンスの分解



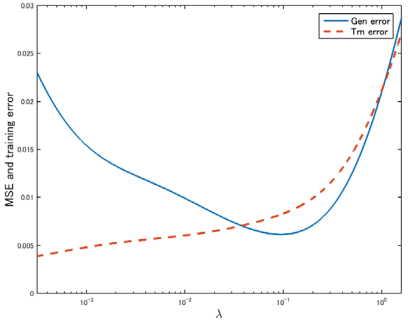
バイアスとバリアンスのトレードオフ

- モデルが大きい：バイアス小，バリアンス大
- モデルが小さい：バイアス大，バリアンス小

サンプルサイズに合わせて適切なモデルを選択する必要がある．

Mallows' CP規準

$$\hat{\beta}_{(\lambda)} = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} (y_i - x_i^\top \beta)^2 + \lambda \|\beta\|^2 \right\}$$
$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \{ \|Y - X\beta\|^2 + \lambda \|\beta\|^2 \}$$



Mallows' CP規準

$$\hat{L}(\lambda) := \sum_{i=1}^{n} (y_i - x_i^\top \hat{\beta}_{(\lambda)})^2 + 2\hat{\sigma}^2 \mathrm{Tr}[X(X^\top X + \lambda \mathrm{I})^{-1} X^\top]$$

訓練誤差　　補正項

$\hat{L}(\lambda)$ を最小にする λ を選択.

- Mallows' CP 規準 $\hat{L}(\lambda)$ は予測誤差 $\mathrm{E}_{x,y}[(y - x^\top \hat{\beta})^2]$ の推定量.
- $\hat{\sigma}^2$ としては最小二乗推定量 $\hat{\beta}_{\mathrm{LS}}$ を用いて $\hat{\sigma}^2 = \|Y - X\hat{\beta}_{\mathrm{LS}}\|^2/n$ を用いることが多い.

※ λ =0の時，AICと一致する.

クロスバリデーション (Cross-Validation)

適切なハイパーパラメータを選ぶ方法

- 観測データへの当てはまりではなく予測誤差を最小化.
- 観測データへの当てはまりを最良にするのは $\lambda = 0$.

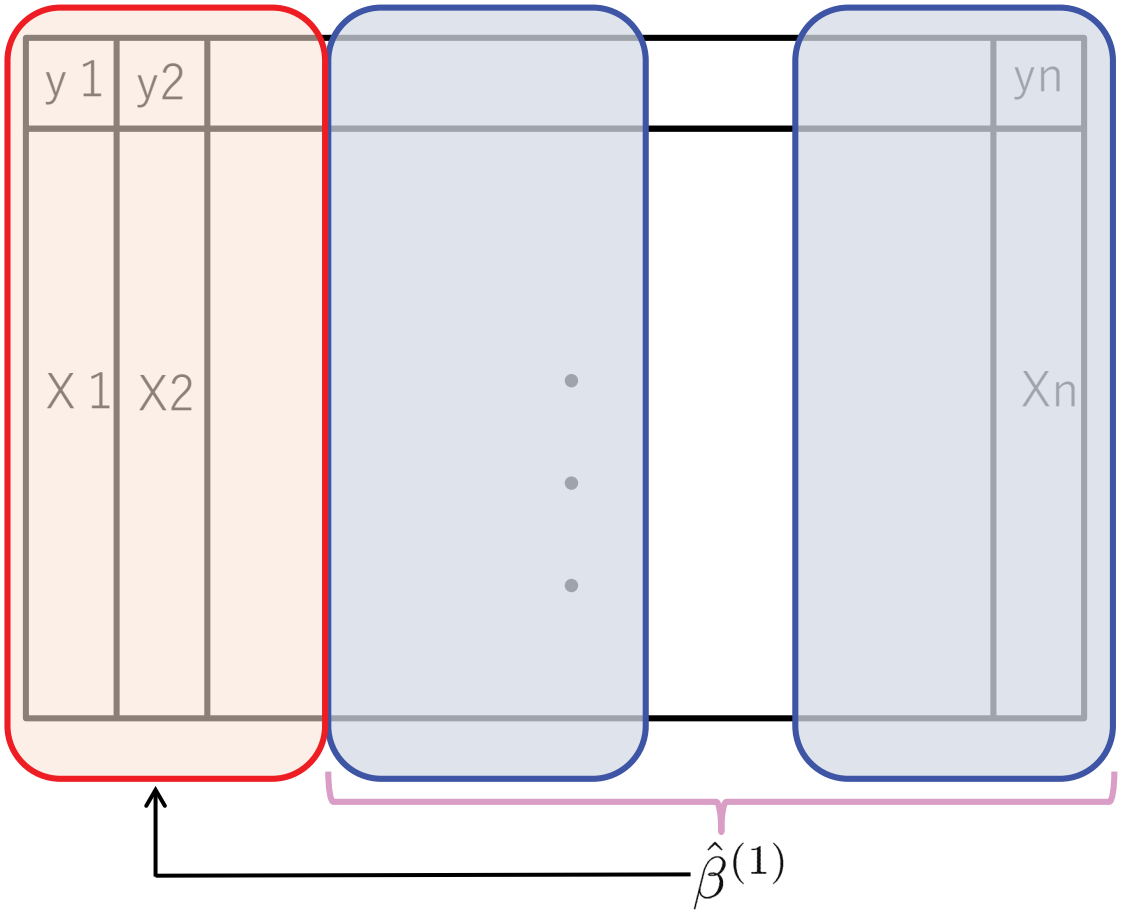
とにかくあらゆる問題に適用可能
「とりあえずクロスバリデーション」

k-fold クロスバリデーション

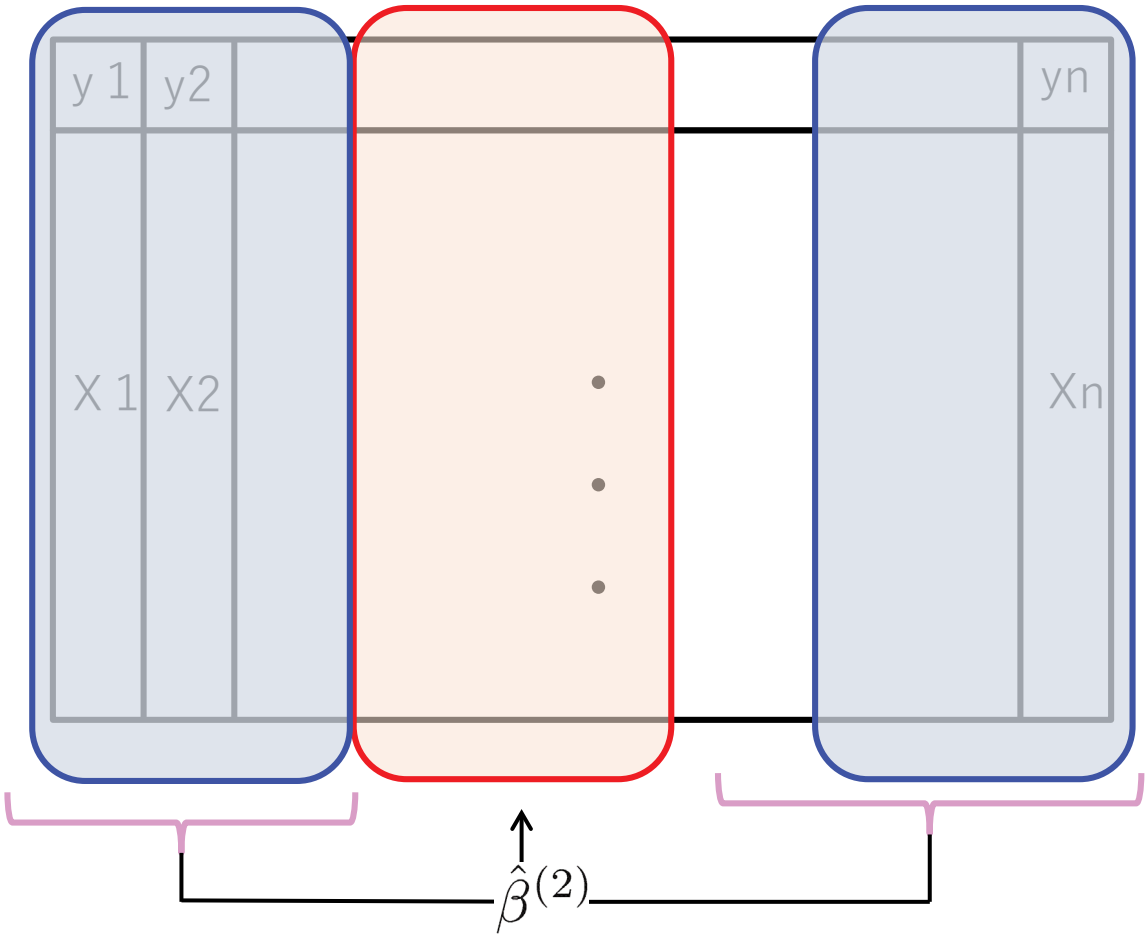
1. まずデータを k 個に分割する.
2. 分割したデータの一つをテストデータとしてとっておき，残りのデータで推定.
3. テストデータ上での予測誤差を計算.
4. 手順 2-3 を k 個のテストデータの取り方について繰り返す.
5. k 回繰り返しの予測誤差の平均を取る=CV スコア.

CV スコアを最小にする λ を選べば良い.

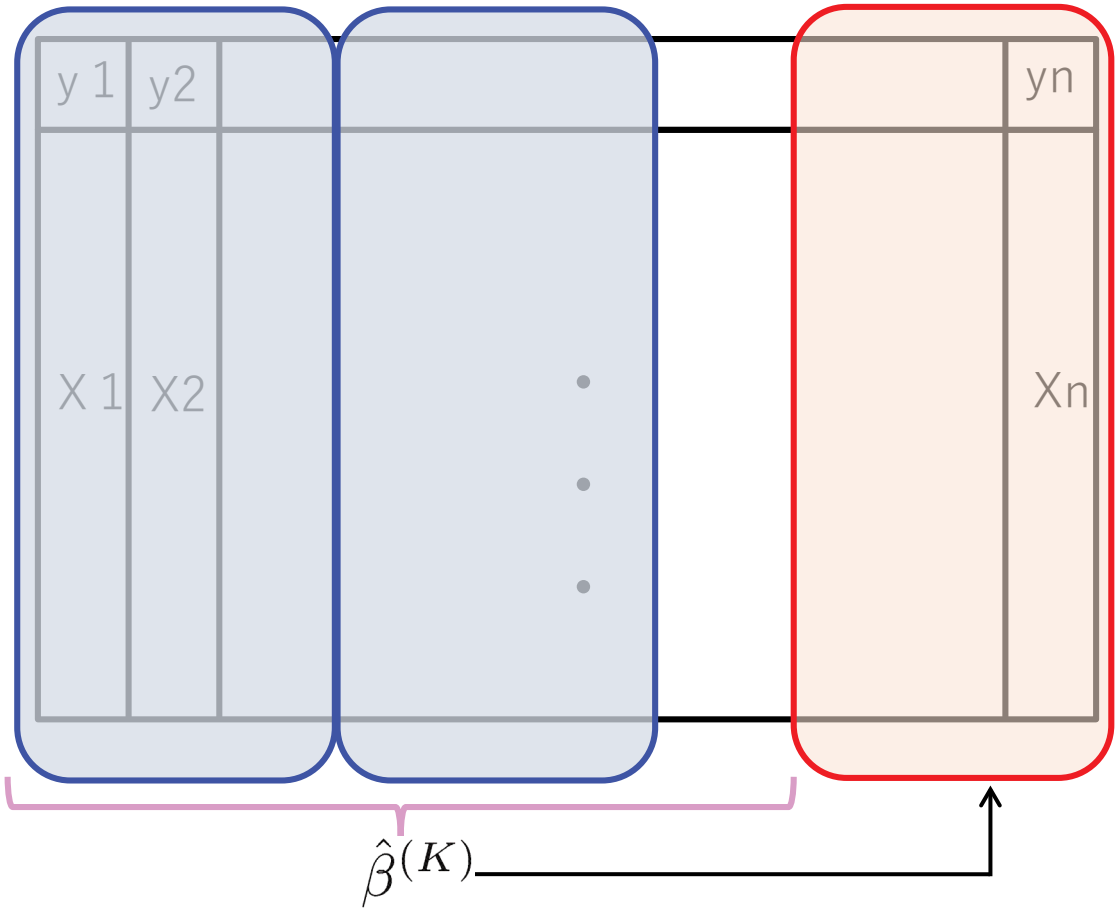
特に $k = n$ (サンプルサイズ) の時，Leave-One-Out-CV (LOOCV) と呼ぶ.



$$\frac{1}{|I_1|}\sum_{i\in I_1}\ell(y_i, \hat{\beta}^{(1)\top}x_i)$$



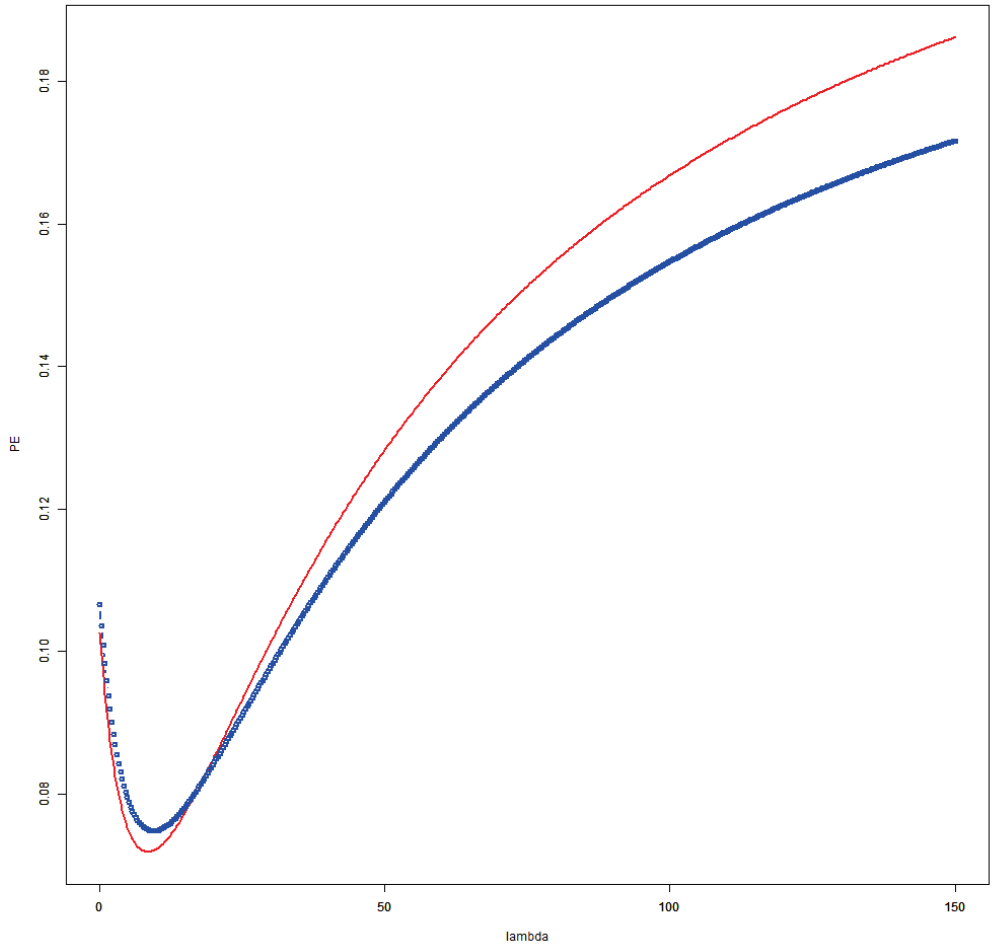
$$\frac{1}{|I_2|}\sum_{i\in I_2}\ell(y_i,\hat{\beta}^{(2)\top}x_i)$$



$$\frac{1}{|I_K|} \sum_{i \in I_K} \ell(y_i, \hat{\beta}^{(K)\top} x_i)$$

実例

$n = 100,\ d = 10$ のリッジ回帰 (ガウスマルコフモデル+二乗ノルム正則化)



予測誤差 (赤線) とCVスコア (青線)

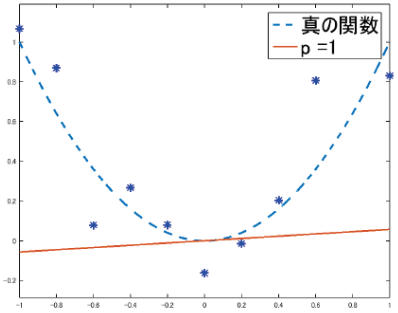
特徴選択

- 重回帰分析では説明変数を追加するごとに残差は小さくなってゆく.
- 余分な説明変数を使うと過学習を起こしてしまう.
  - 観測済みデータによく当てはまっても，未観測のデータへの当てはまりが悪くなることがある.
  - サンプルサイズに比して複雑なモデルは使うべきではない.

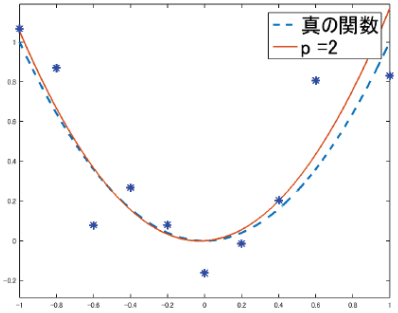
中古マンション価格の予測：床面積，築年数，駅からの距離，建ぺい率，...

例：多項式回帰

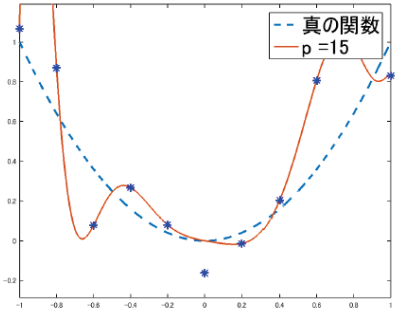
$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} \{y_i - (\beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d)\}^2$$



$d = 1$



$d = 2$



$d = 15$

AICによる特徴選択

赤池情報量規準，AIC（Akaike Information Criterion）

予測精度が一番良いモデルを選択するための規準

$\widehat{\boldsymbol{\beta}}^{(d)} = [\widehat{\beta}_1, \ldots, \widehat{\beta}_d, 0, \ldots, 0]^\top$　：d変数のみを用いた最小二乗推定量

$$\mathrm{AIC} = \frac{\| Y - X\widehat{\boldsymbol{\beta}}^{(d)} \|^2}{\sigma^2} + 2d$$

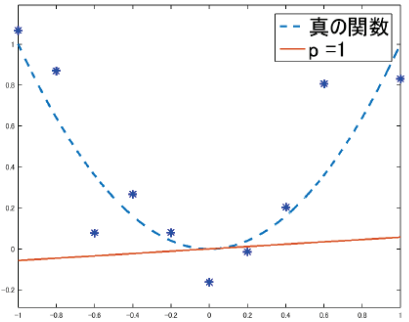
AICを最小化する変数の組を探す．

AIC = データへの当てはまりの良さ + モデルの複雑さ

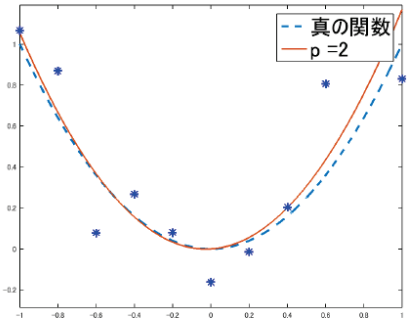
- d変数を用いた推定は最初のd変数である必要はない．
- dを増やせばAICの第一項は減少し，第二項は増大．
- AICの期待値は予測誤差になることが示せる．

例：多項式回帰

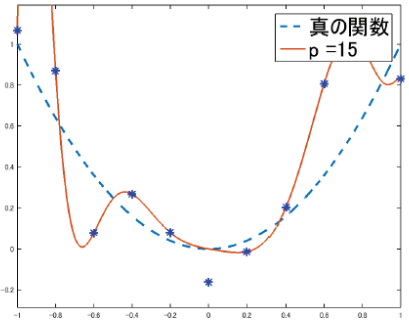
$$\min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} \{y_i - (\beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d)\}^2$$



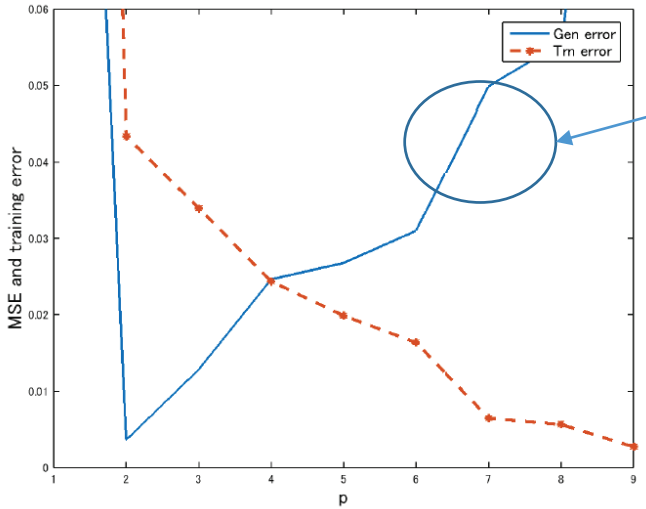
$d = 1$



$d = 2$



$d = 15$



過学習

汎化誤差と次元dの関係

訓練誤差は単調に減少するが，
汎化誤差は途中で増大する．
AIC により適切な次元が選ばれる．

中古マンション価格データの分析

国土交通省が公開している不動産取引価格情報から世田谷区の中古マンション取引価格データ (平成２５年度第３四半期分) を取得．ここから一部を抜粋したデータで回帰分析をやってみる．
http://www.land.mlit.go.jp/webland/download.html

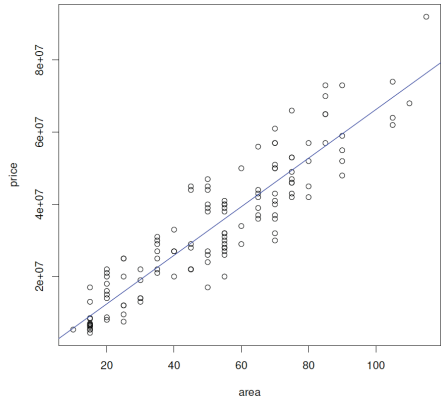
従属変数：価格
説明変数：
1. 最寄駅からの距離（徒歩）
2. 延床面積
3. 建物の構造
4. 建ぺい率
5. 容積率
6. 建築年
7. 最寄り駅に急行が止まるか (0-1変数で表現)

Rの関数 lm を使って分析.

回帰分析関数（lm）

最小二乗法の計算

```
sman.lm <- lm(price ~ area,data=sman)  #回帰分析はこの一行で OK
plot(sman$area,sman$price, xlab="area",ylab="price")  #結果をプロット
abline(sman.lm , lwd=1 , col="blue")
```



AICによる特徴選択

```
sman.lmall <- lm(price ~., data=sman)
sman.lmAIC <- step(sman.lmall)
summary(sman.lmAIC)
```

step() でAIC 最小のモデル(説明変数の組) を探索.

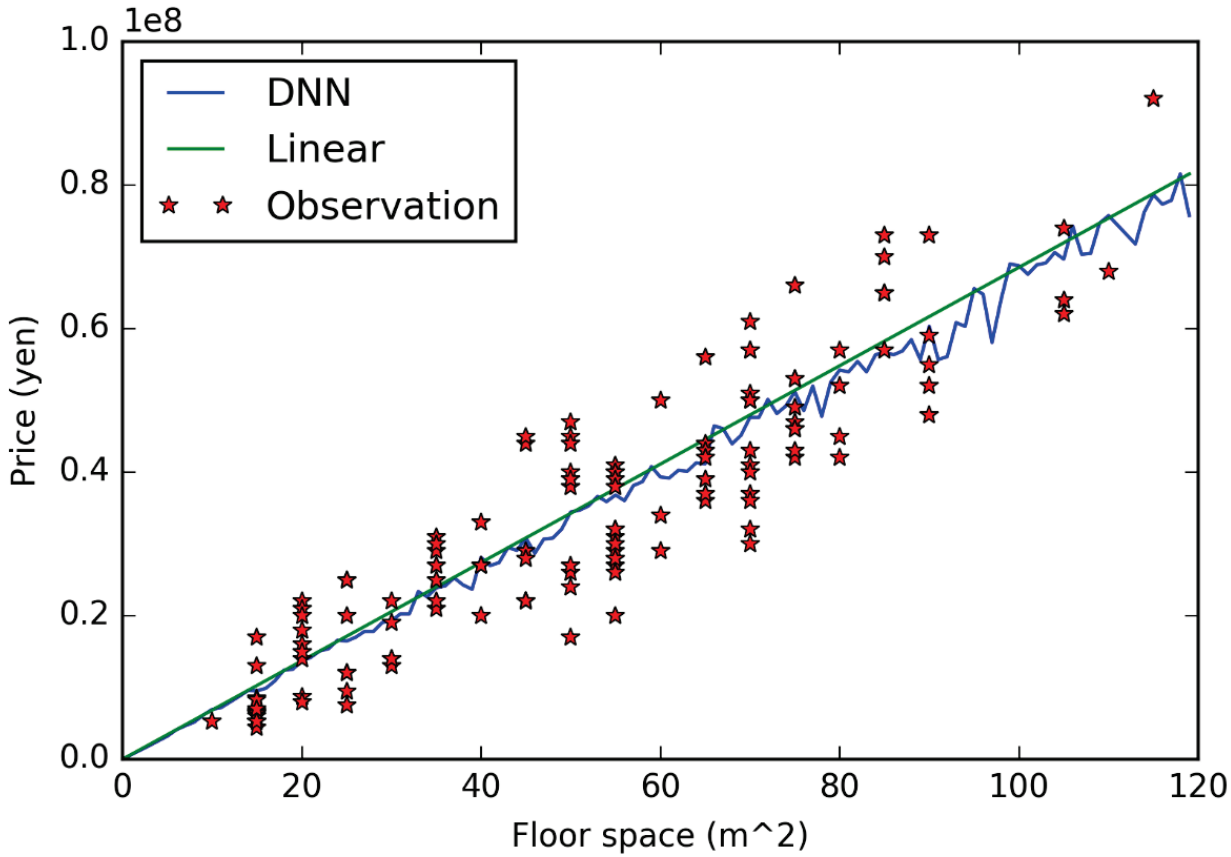
最寄駅からの距離 + 床面積 + 築年数

の三変数モデルが採用された.

1. 最寄駅からの距離（徒歩）
2. 延床面積
3. 建物の構造
4. 建ぺい率
5. 容積率
6. 築年数
7. 最寄り駅に急行が止まるか

線形モデル vs 深層学習

過学習の例



深層学習を使うには簡単&データが少なすぎる

マンションの価格推定

DNN: 中間層２層横幅100の深層NN, **Linear**: 単回帰モデル

汎化誤差（平均二乗誤差）： DNN: 1.30×10^{15}, Linear: 6.26×10^{13}

一概に何でもかんでも深層学習が良いとは言えない

これまでのまとめ

- 機械学習の歴史
- 機械学習の考え方
  - 複雑な規則をデータから学ぶ
- モデルと損失
  - 学習：期待損失最小化
- 過学習の問題
  - 複雑なモデルを当てはめれば良いわけではない.
  - 正則化
  - 変数選択

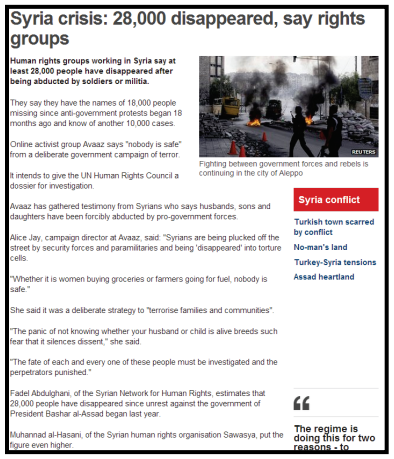
高次元スパース推定

高次元データ

インターネットや計測機器の発達により多様なデータが取得可能

多くの場合で高次元

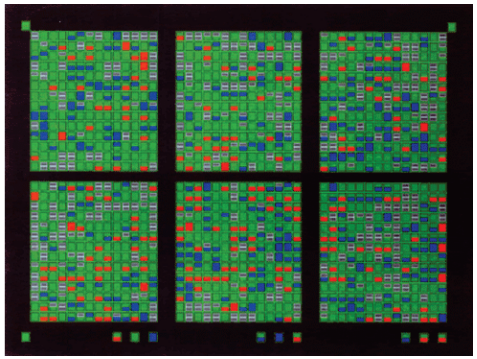
- 遺伝子データ
- テキストデータ
- マーケティングデータ
- 金融データ



Bag of words

数百万次元

Syria	13
people	5
bomb	7
economy	1
immigrants	2
soccer	0
walk	1

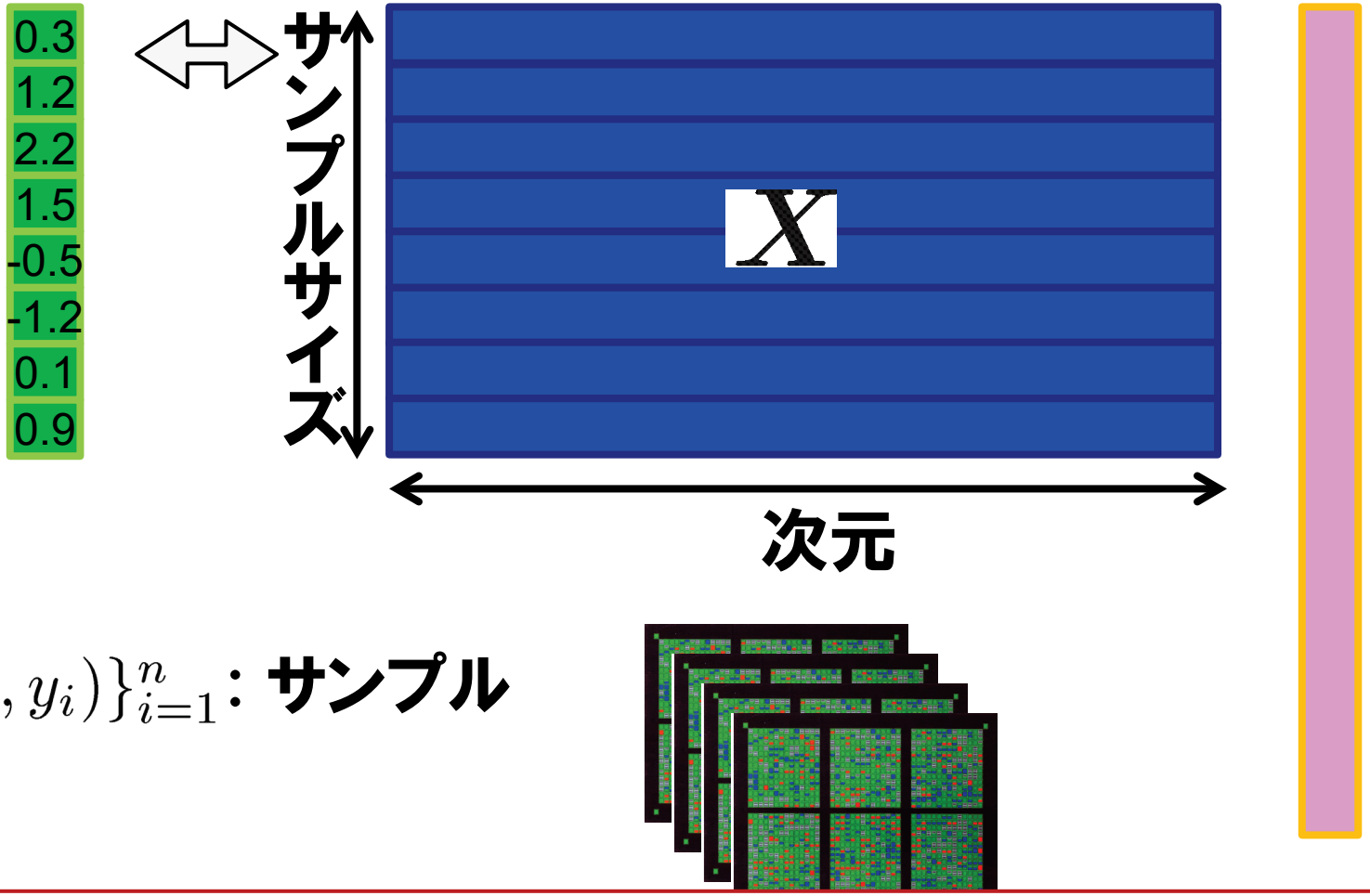


遺伝子発現量

数万次元

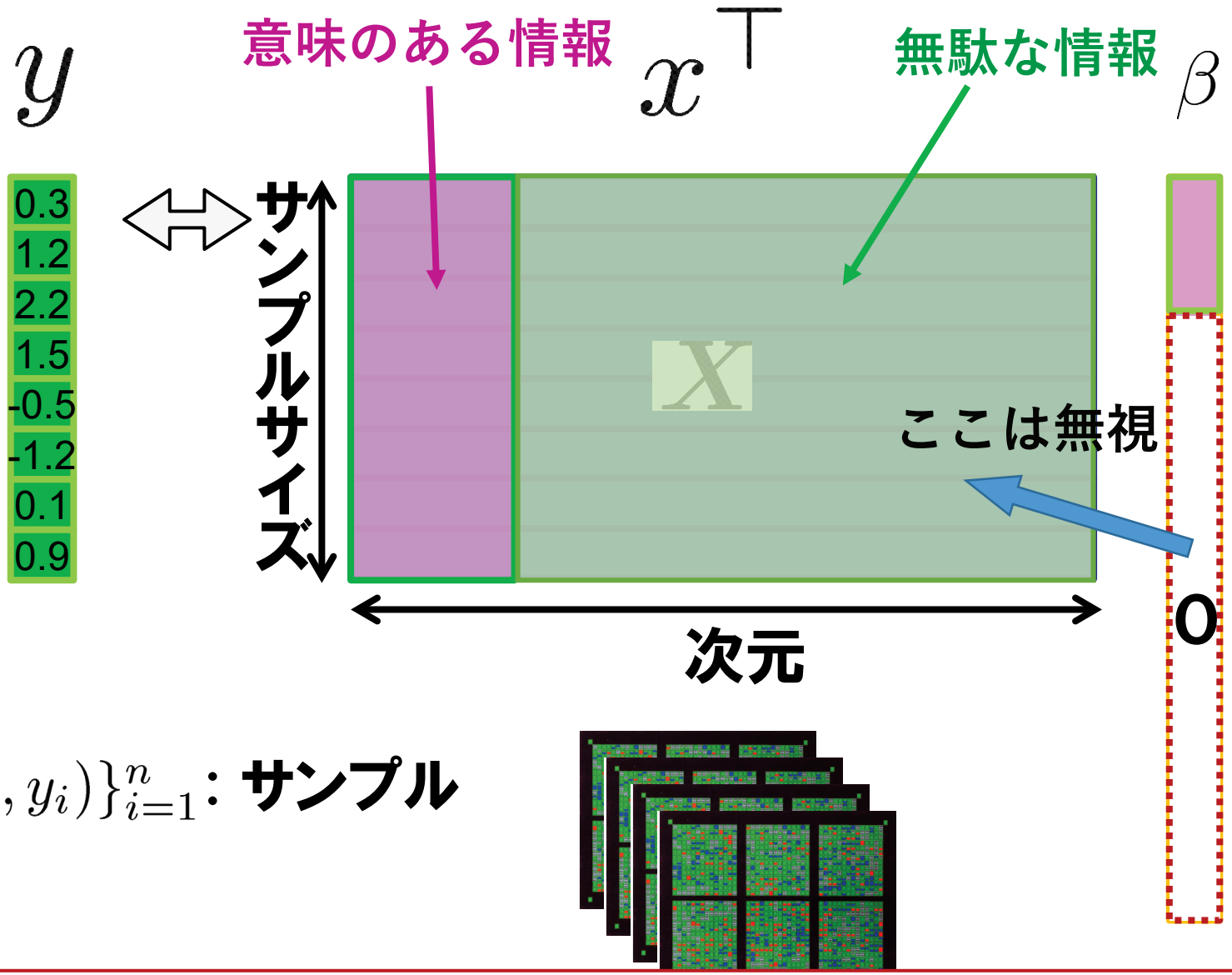
0.5
2.4
4.2
0.2
1.3
0.1
5.3

y $x^\top$ β



$\{(x_i, y_i)\}_{i=1}^n$: サンプル

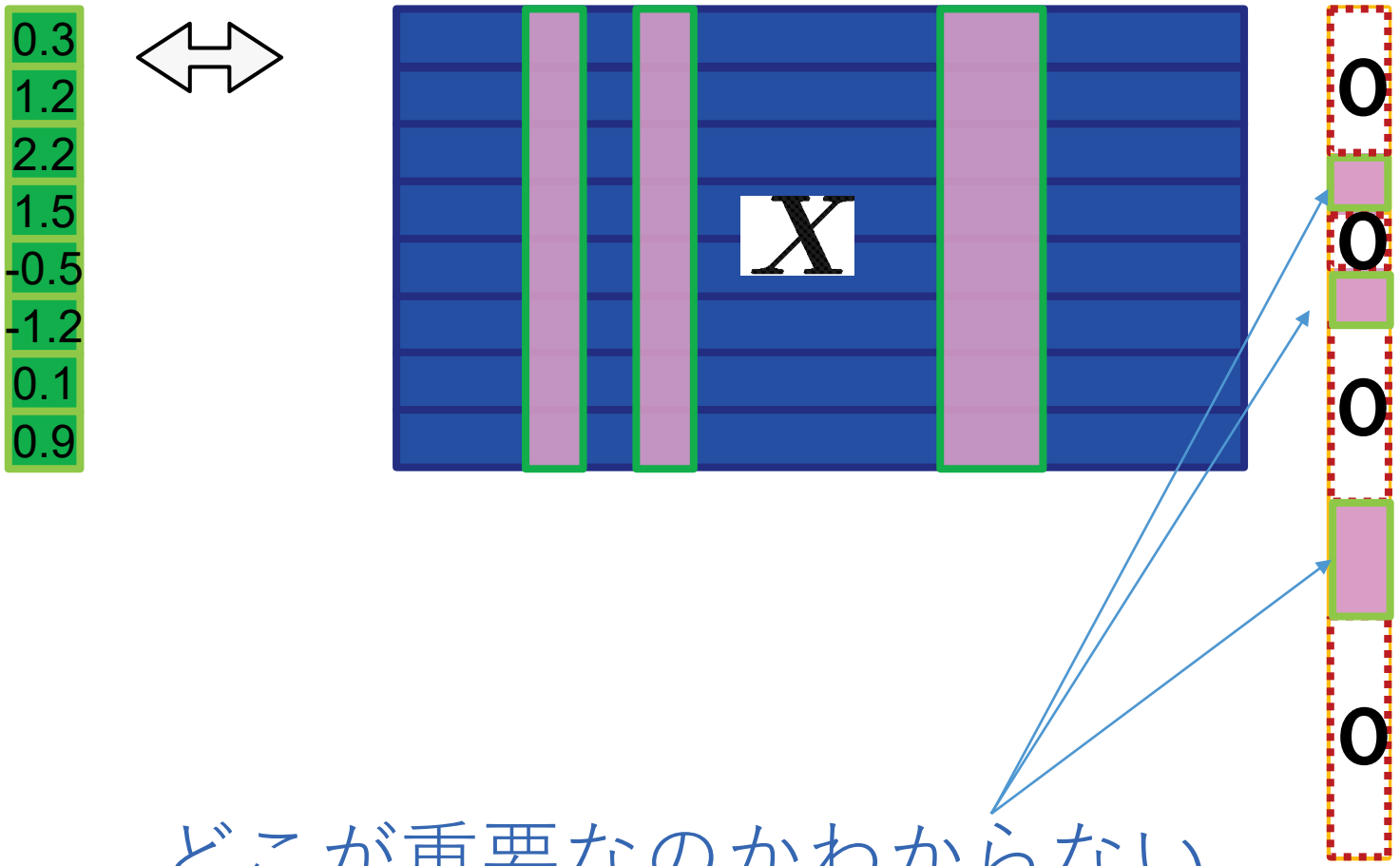
次元 > サンプルサイズ → 余分な情報を落としたい



$\{(x_i, y_i)\}_{i=1}^n$: サンプル

次元 > サンプルサイズ → 余分な情報を落としたい

スパースモデリング



どこが重要なのかわからない

→ 特徴選択：データから学習

予測に寄与する特徴量を特定できれば解釈性も上がる

AICによる特徴選択（組み合わせ的方法）

AIC: 赤池情報量規準 → 最尤推定量の予測誤差の不偏推定量

AIC最小化

$$\hat{\beta}_{\mathrm{AIC}} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \|Y - X\beta\|^2 + 2\sigma^2 \|\beta\|_0$$

データへの当てはまり

次元に対する罰則（正則化）

ただし$\|\beta\|_0 = \beta$の非ゼロ要素の個数 ：L_0ノルムと言う.

- 予測誤差を近似的に最小化
- 変数の組み合わせの数： 2^p個の候補 (膨大)
- NP困難

線形モデルを仮定

$$Y = X\beta^* + \xi$$

サンプルサイズn, 次元p

観測ノイズ：分散σ^2の正規分布

LASSOによる特徴選択（凸最適化）

Lasso [L_1正則化] (R. Tsibshirani (1996))

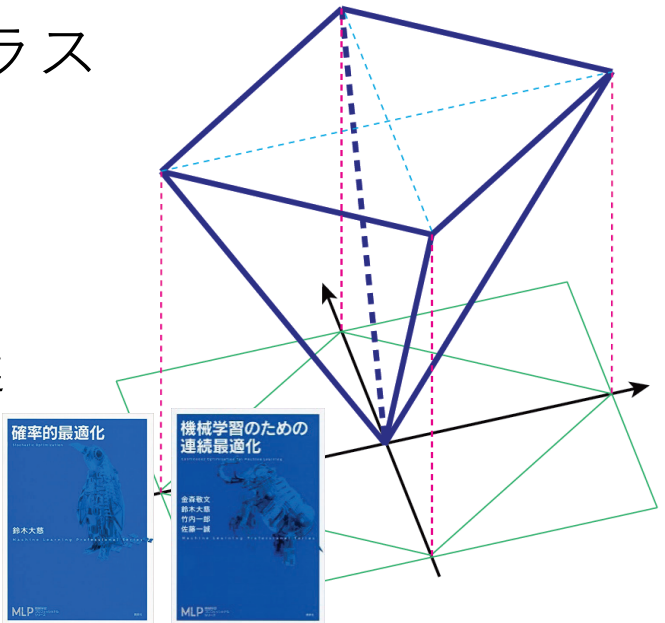
$$\hat{\beta}_{\text{Lasso}} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \|Y - X\beta\|^2 + \lambda \|\beta\|_1$$

データへの当てはまり　　次元に対する罰則（正則化）

ただし $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$ ：L_1ノルムと言う.

Lassoは**凸最適化**と呼ばれる問題のクラス

- **高速に解ける（近接勾配法等）**
- L_1ノルムはL_0ノルムを最も良く近似する凸関数
- パラメータλはクロスバリデーションで選べば良い.
- 理論が豊富.

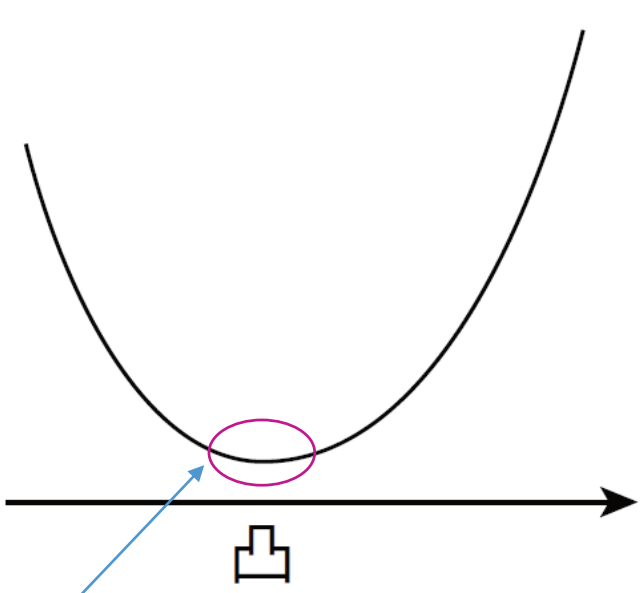


書籍：確率的最適化，機械学習のための連続最適化

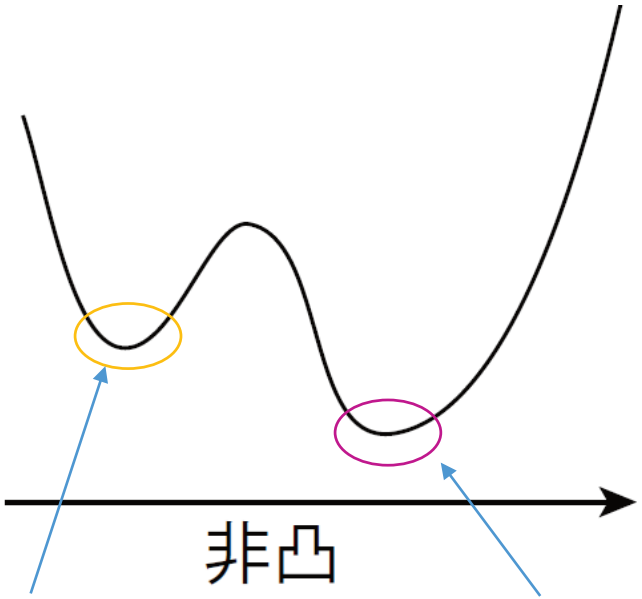
凸関数

凸最適化　＝　凸関数の最適化

$$\theta f(x) + (1-\theta) f(y) \geq f(\theta x + (1-\theta) y) \quad (\forall x, y \in \mathbb{R}^p, \theta \in [0,1])$$



局所最適解＝大域的最適解



局所最適解　　大域的最適解

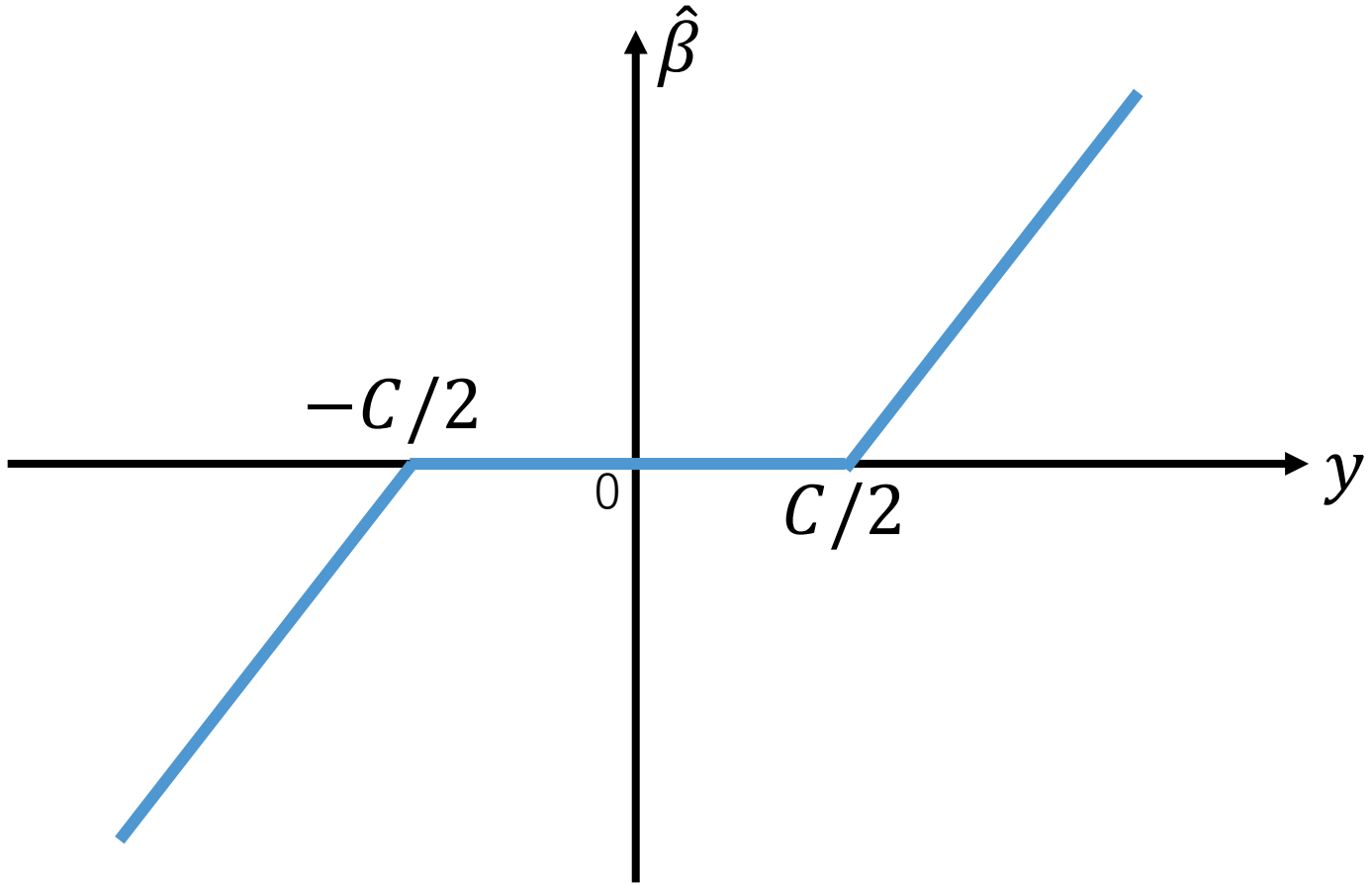
凸関数は局所最適解が大域的最適解

→ 効率的な最適化が可能な場合が多い

簡単な例

1 次元の場合

$$\min_{\beta \in \mathbb{R}} (y - \beta)^2 + C|\beta|$$



Lassoのスパース性

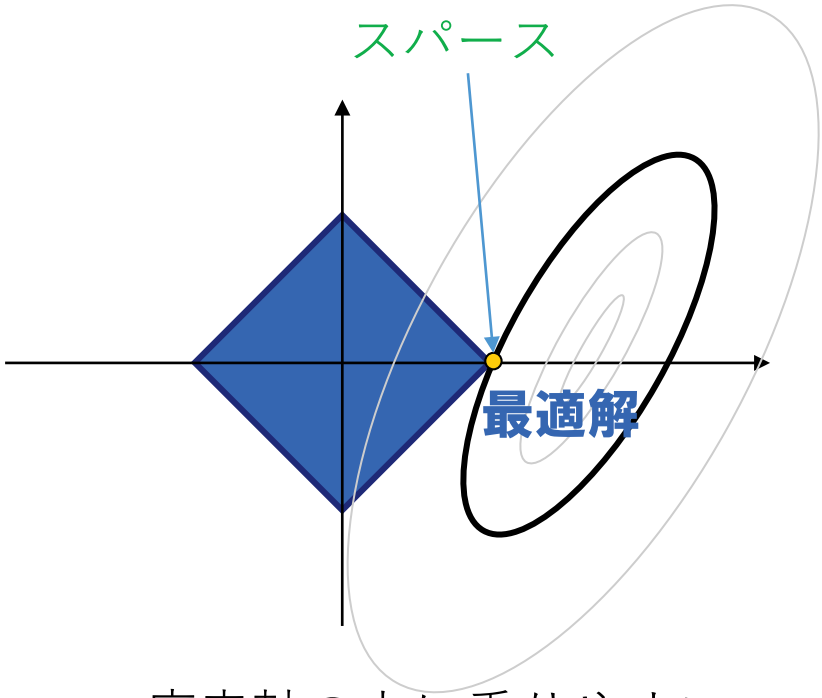
$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \|X\beta - Y\|_2^2 + \lambda \sum_{j=1}^{p} |\beta_j|. \quad \Leftrightarrow \quad \min_{\beta \in \mathbb{R}^p} \frac{1}{n} \|X\beta - Y\|^2 \text{ s.t. } \sum_{j=1}^{p} |\beta_j| \leq C$$

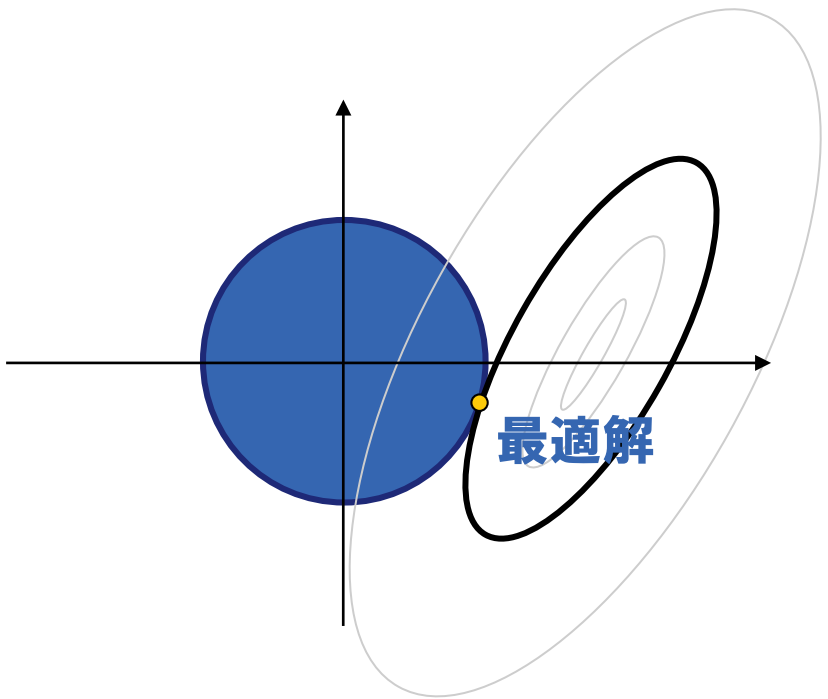
L1正則化

$$\|\beta\|_1 = |\beta_1| + \cdots + |\beta_p|$$

L２正則化（リッジ正則化）

$$\|\beta\|_2^2 = \beta_1^2 + \cdots + \beta_p^2$$





座表軸の上に乗りやすい

スパース推定によって予測に必要な変数が自動的に選ばれる

スパース性の恩恵

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \|X\beta - Y\|_2^2 + \lambda_n \sum_{j=1}^{p} |\beta_j|$$

d = 真のベクトルβ^*の非ゼロ要素の数 (予測に寄与する変数の数)

定理 (Lassoの収束レート (Bickel et al., 2009; Zhang, 2009))

ある条件のもと（制限等長性など），ある定数Cが存在して,

$$\|\hat{\beta} - \beta^*\|_2^2 \leq C \frac{d \log(p)}{n}$$

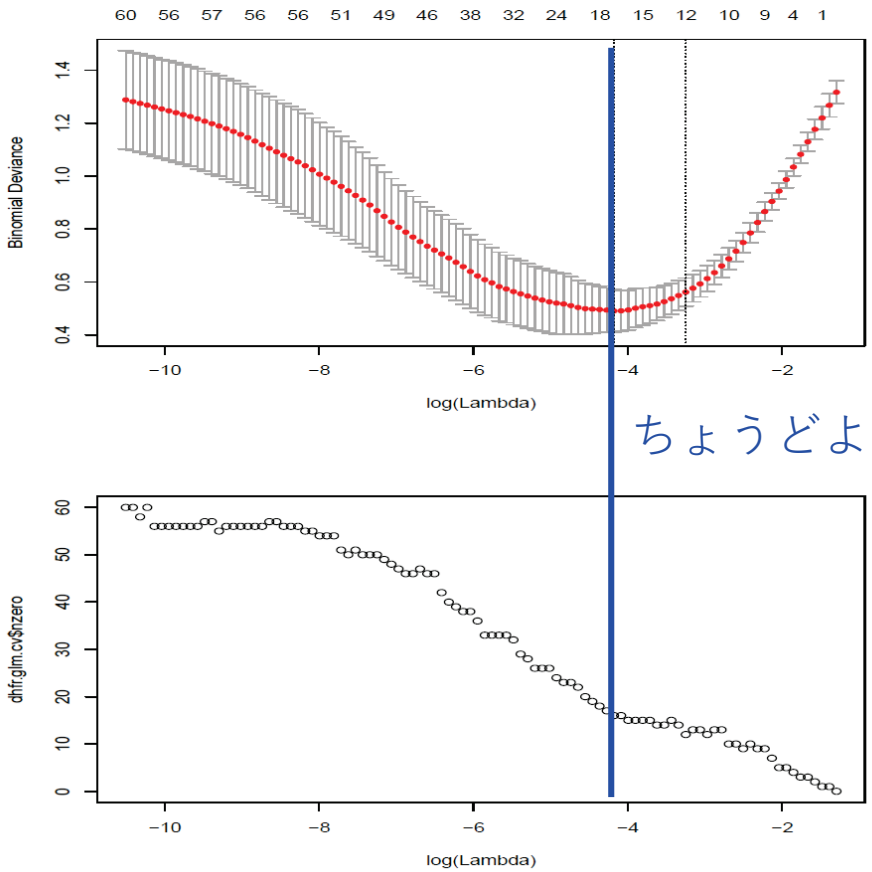
- 全体の次元p はたかだか$O(\log(p))$でしか影響しない!
- 実質的次元d が支配的.
- 高次元スパースな問題を精度よく解くことができる.

} **過学習を防止**

推定誤差 $\dfrac{d \log(p)}{n} \ll \dfrac{p}{n}$ (最小二乗法)
過学習してしまう

低次元性（スパース性）をうまく利用できている.

ジヒドロ葉酸レダクターゼデータにおける実験



CVスコア
(予測精度)

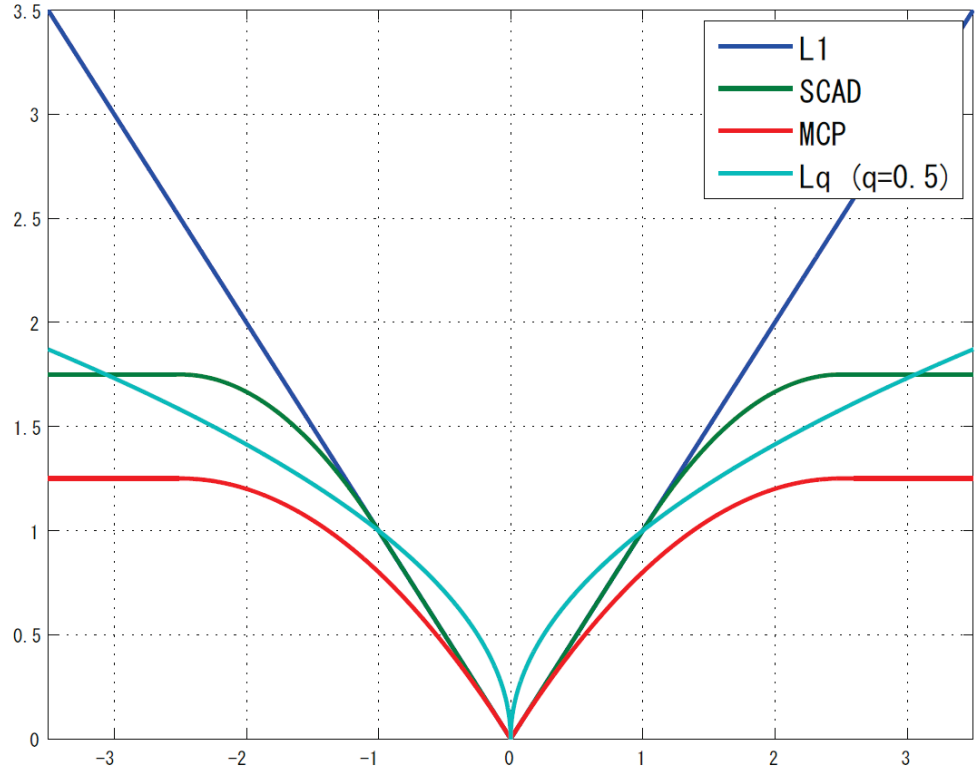
ちょうどよい正則化

非ゼロ要素の個数

スパース性と汎化誤差

横軸：正則化パラメータ．縦軸：(上段) CV スコア, (下段) 非ゼロ要素個数

非凸正則化



SCAD

$$\rho(|\beta|,\lambda) = \begin{cases} \lambda|\beta| & (|\beta| \le \lambda) \\ \frac{-|\beta|^2+2a\lambda|\beta|-\lambda^2}{2(a-1)}| & (\lambda < |\beta| \le a\lambda) \\ \frac{(a+1)\lambda^2}{2}| & (|\beta| \le a\lambda) \end{cases}$$

MCP

$$\rho(|\beta|;\lambda) = \lambda \int_0^{|\beta|} (1 - x/(\gamma\lambda))_+ \mathrm{d}x$$

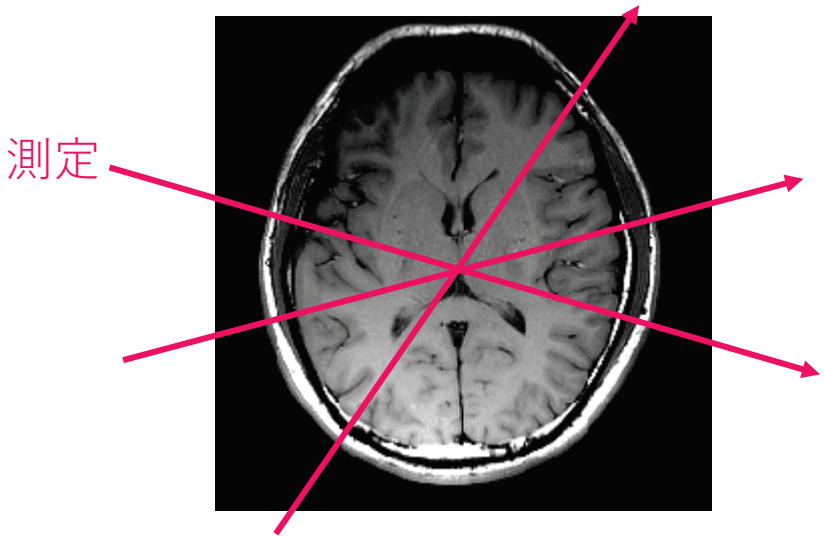
- SCAD (Smoothly Clipped Absolute Deviation) (Fan and Li, 2001)
- MCP (Minimax Concave Penalty) (Zhang, 2010)
- Lq 正則化(q < 1), Bridge 正則化(Frank and Friedman, 1993)

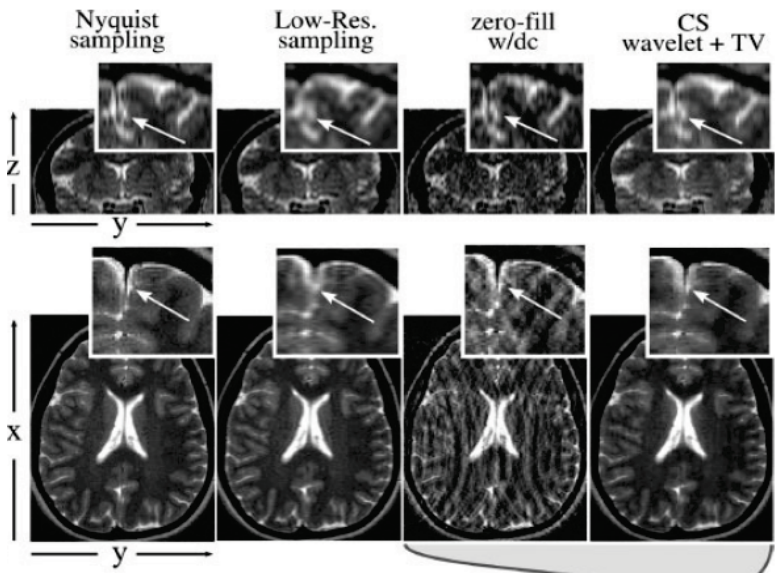
よりスパースな解．その代わり最適化は難しくなる．

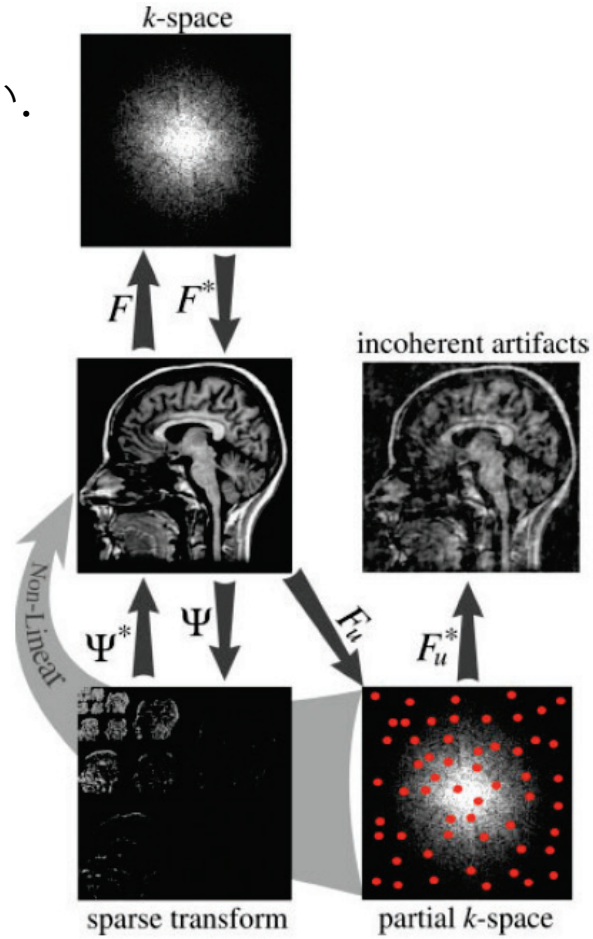
ただし，最近は局所最適解でも統計的性質は良いことが示されている．

MRIへの応用

なるべく測定時間（観測回数）を減らしたい.







画像はwavelet基底に関してスパース
→少数の観測（サンプル）でも大丈夫

[Lustig, Donoho and Pauly: Sparse MRI: The application of compressed sensing for rapid MR imaging, 2007]

スパース共分散選択

$$x_k \sim N(0,\Sigma)\ \ (\text{i.i.d.}, \Sigma \in \mathbb{R}^{p\times p}), \qquad \widehat{\Sigma} = \frac{1}{n}\sum_{k=1}^{n} x_k x_k^\top$$

$$\hat{S} = \underset{S:\text{半正定対称}}{\arg\min} \left\{ -\log(\det(S)) + \mathrm{Tr}[S\widehat{\Sigma}] + \lambda \sum_{i,j=1}^{p} |S_{i,j}| \right\}.$$

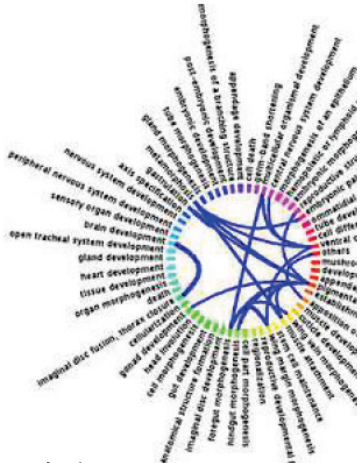
データへの当てはまり（正規分布の負対数尤度）

L1正則化

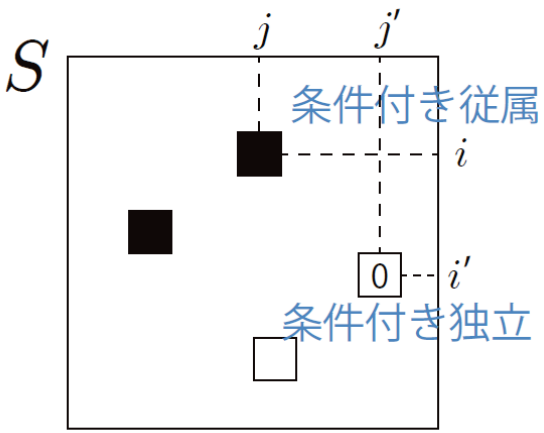
[Meinshausen and Buhlmann, 2006, Yuan and Lin, 2007, Banerjee et al., 2008]

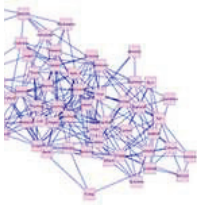
- の逆行列Sを推定
- $S_{ij}=0$ ⇔ 「X_iとX_jが条件付き独立」
- $S_{ij}=0$なら変数X_iと変数X_jは直接的に相互作用しないという意味

グラフィカルモデルが凸最適化で推定可能



遺伝子ネットワ

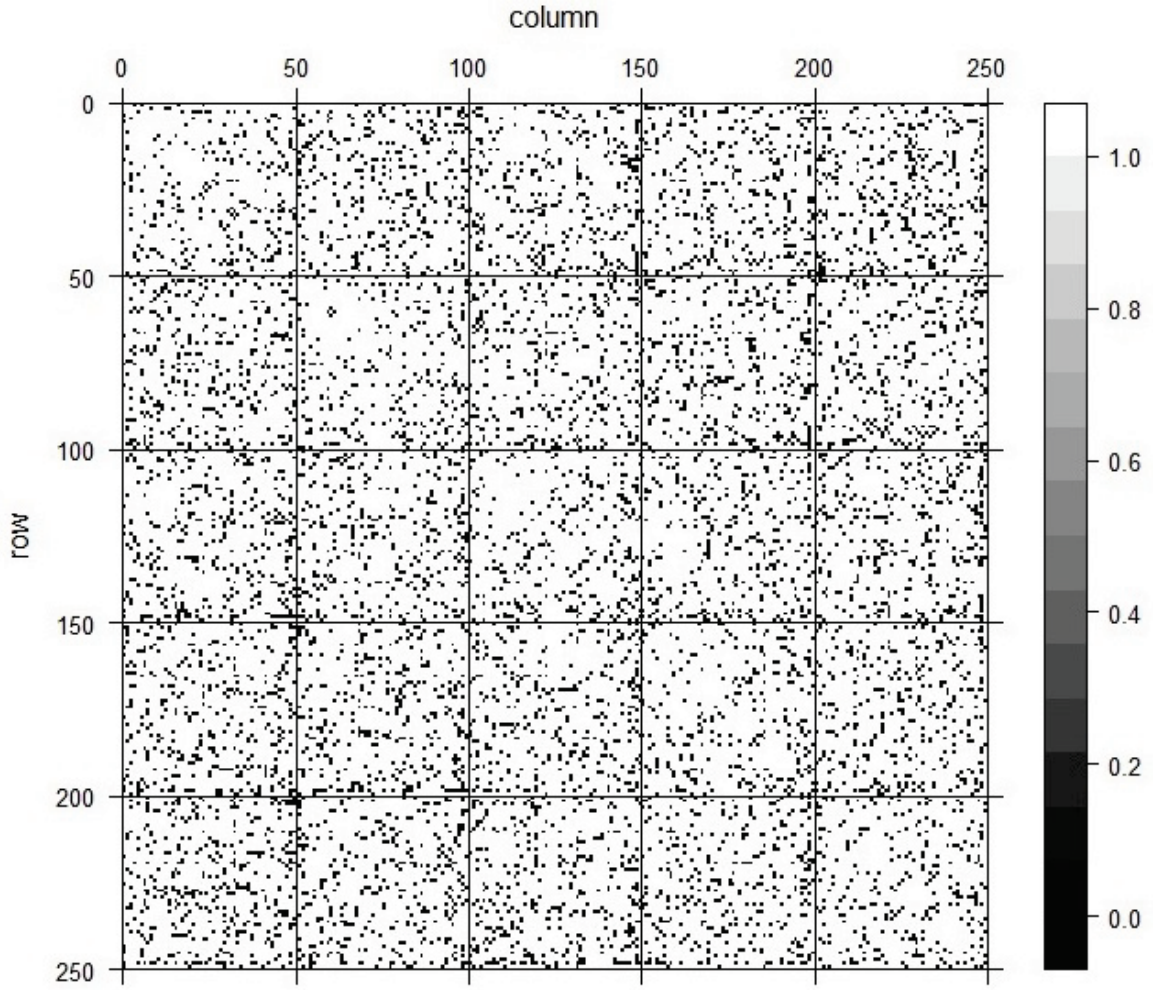




員間の関係

から推定）

[Kolar+etal,2010]



NASDAQ 銘柄からランダム抽出した50 銘柄.
株価データを用いた分散共分散選択. 時間差も考慮.
(2011 年1 月4 日から2014 年12 月31 日まで)
(Lie Michael, Bachelor thesis)

その他のスパース性

スパース正則化はL1正則化だけではない.

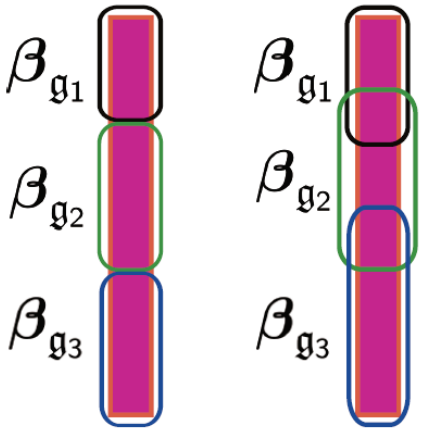
他にも以下のようなより構造を持った正則化がある.

構造的正則化

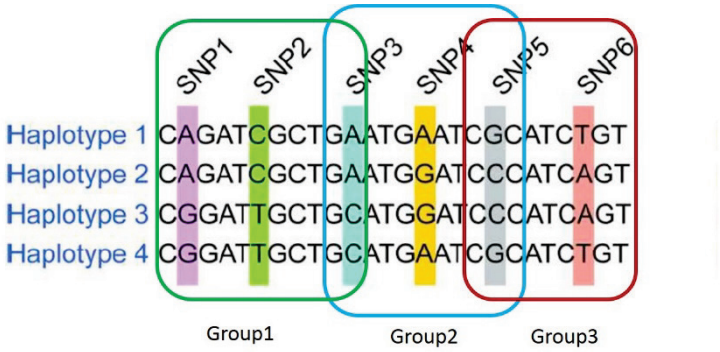
- グループ正則化：変数のグループごと 0 にする.
- 一般化連結正則化
- トレースノルム正則化

グループ正則化

$$\psi(\beta) = C \sum_{\mathfrak{g} \in \mathfrak{G}} \|\beta_{\mathfrak{g}}\|$$



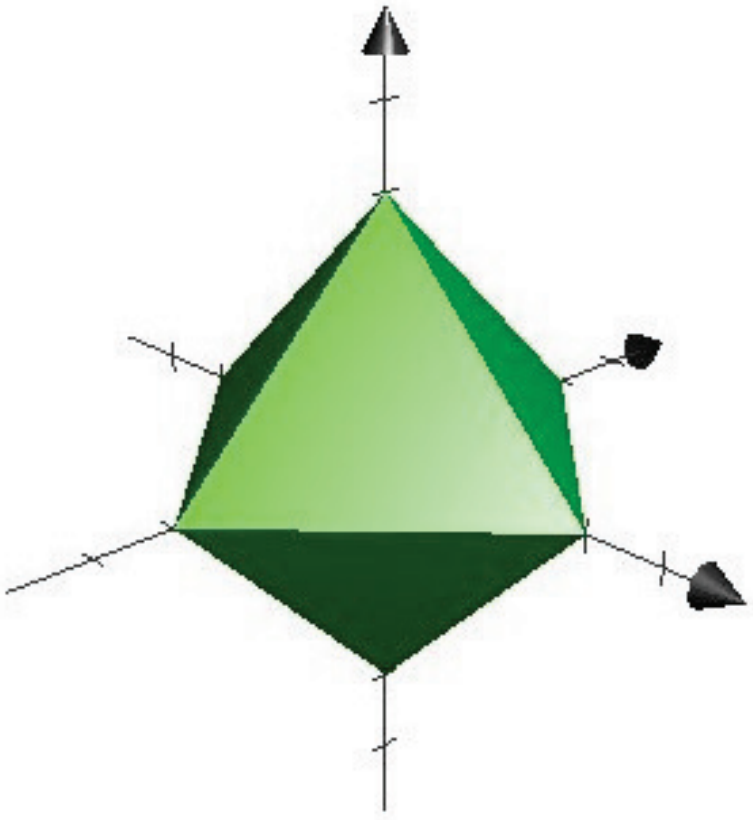
- グループごとに正則化
- グループ全体が0になりやすい.



Genome Wide Association Study (GWAS)
[Balding `06, McCarthy et al. `08]

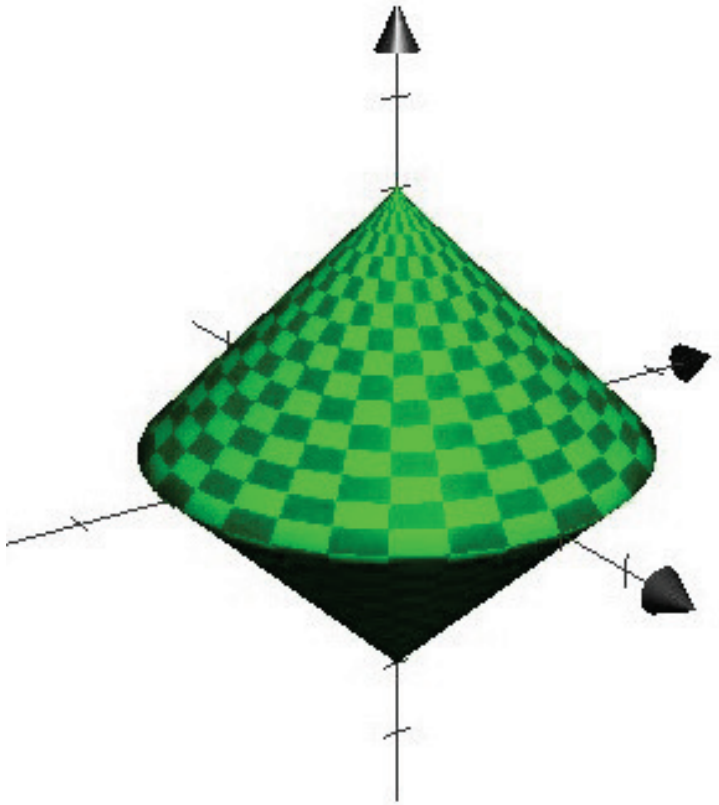
グループ正則化の概形

Lasso



$$|\beta_1| + |\beta_2| + |\beta_3| \leq 1$$

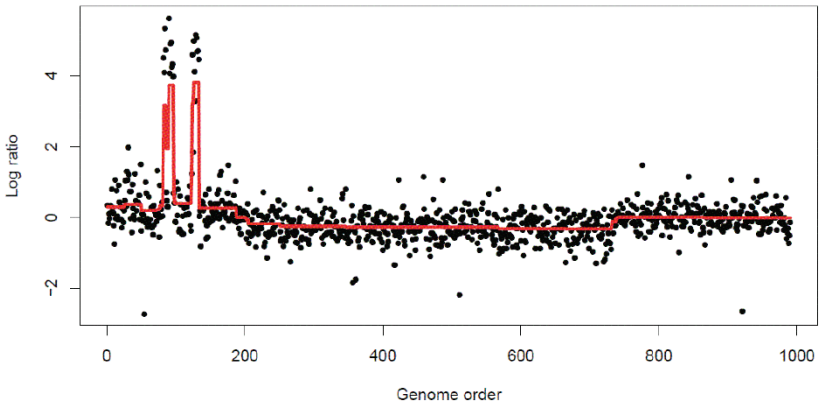
Group Lasso



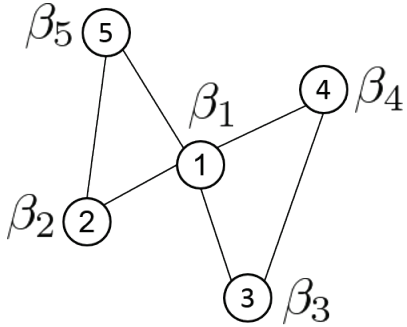
$$\sqrt{\beta_1^2 + \beta_2^2} + |\beta_3| \leq 1$$

一般化連結正則化（Fused Lasso）

$$\psi(\beta) = \sum_{(i,j)\in E} |\beta_i - \beta_j|$$



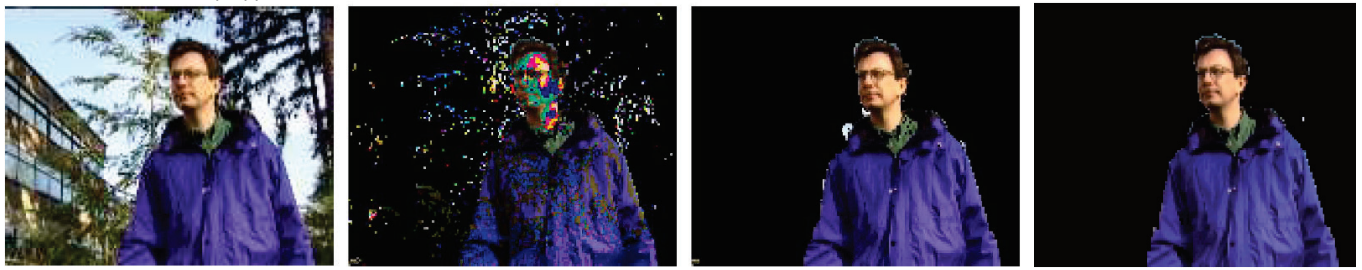
Fused lasso による遺伝子データ解析
[Tibshirani and Taylor `11]



TVデノイジング
（パッチを使わないデノイジング）
[Chambolle `04, Mairal et al., 2009]

背景切り出し [Mairal et al.: 2011]

テスト画像　L1正則化　L1/L2グループ正則化　一般化連結正則化



低ランク行列補完

ベクトルから**行列**の学習へ

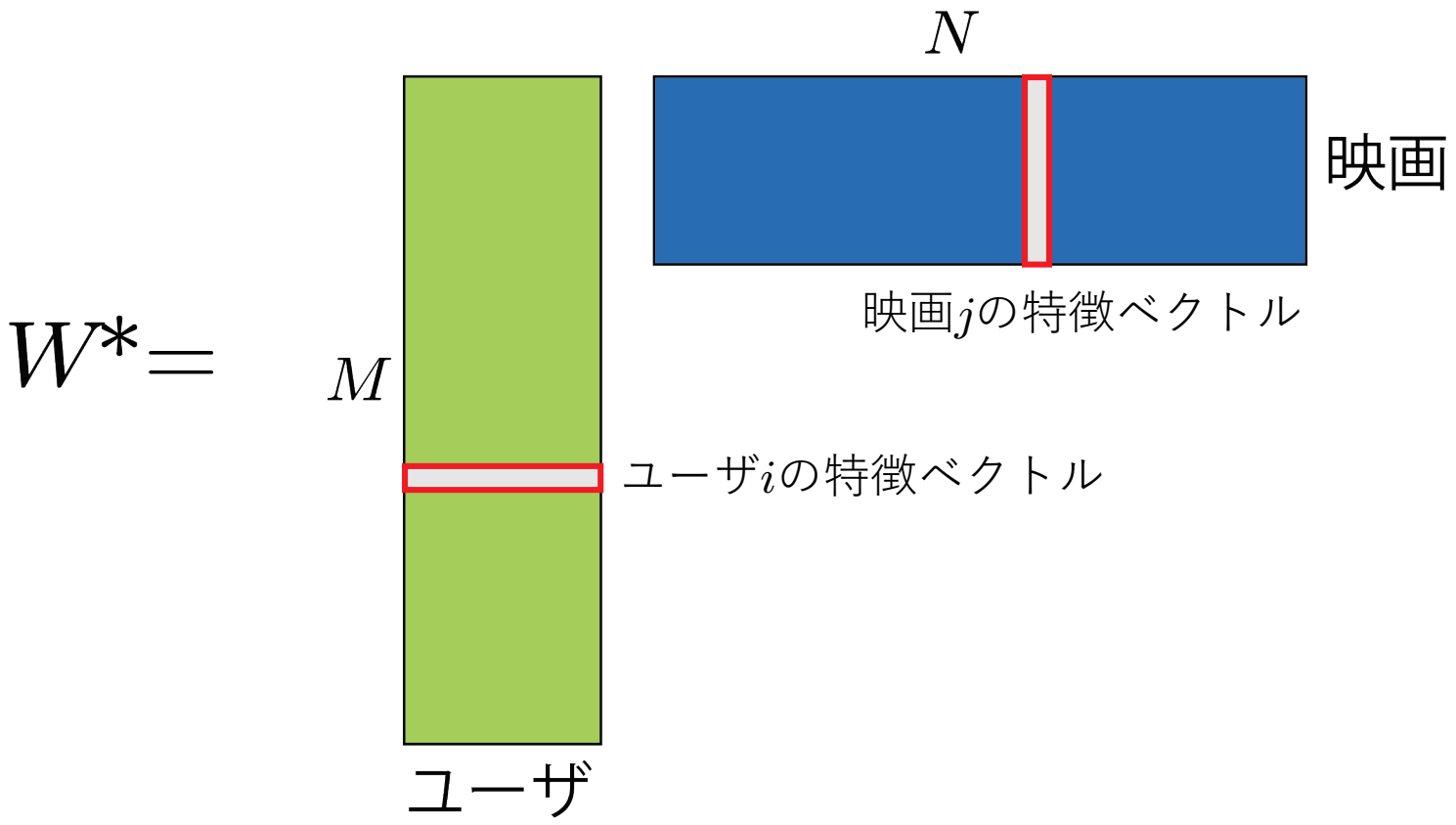
- 推薦システム

	映画 A	映画 B	映画 C	・・・	映画 X
ユーザ 1	4	8	4	・・・	2
ユーザ 2	2	4	2	・・・	1
ユーザ 3	2	4	2	・・・	1
⋮					

ランク 1 と仮定

各ユーザーが各映画をどれだけ好むかという部分的情報がある.
→　残りの部分（*の部分）を埋めたい.
　　低ランク行列補完で可能.

e.g., Netflix prize (100万ドルの賞金,　48万ユーザ×１万8千映画)

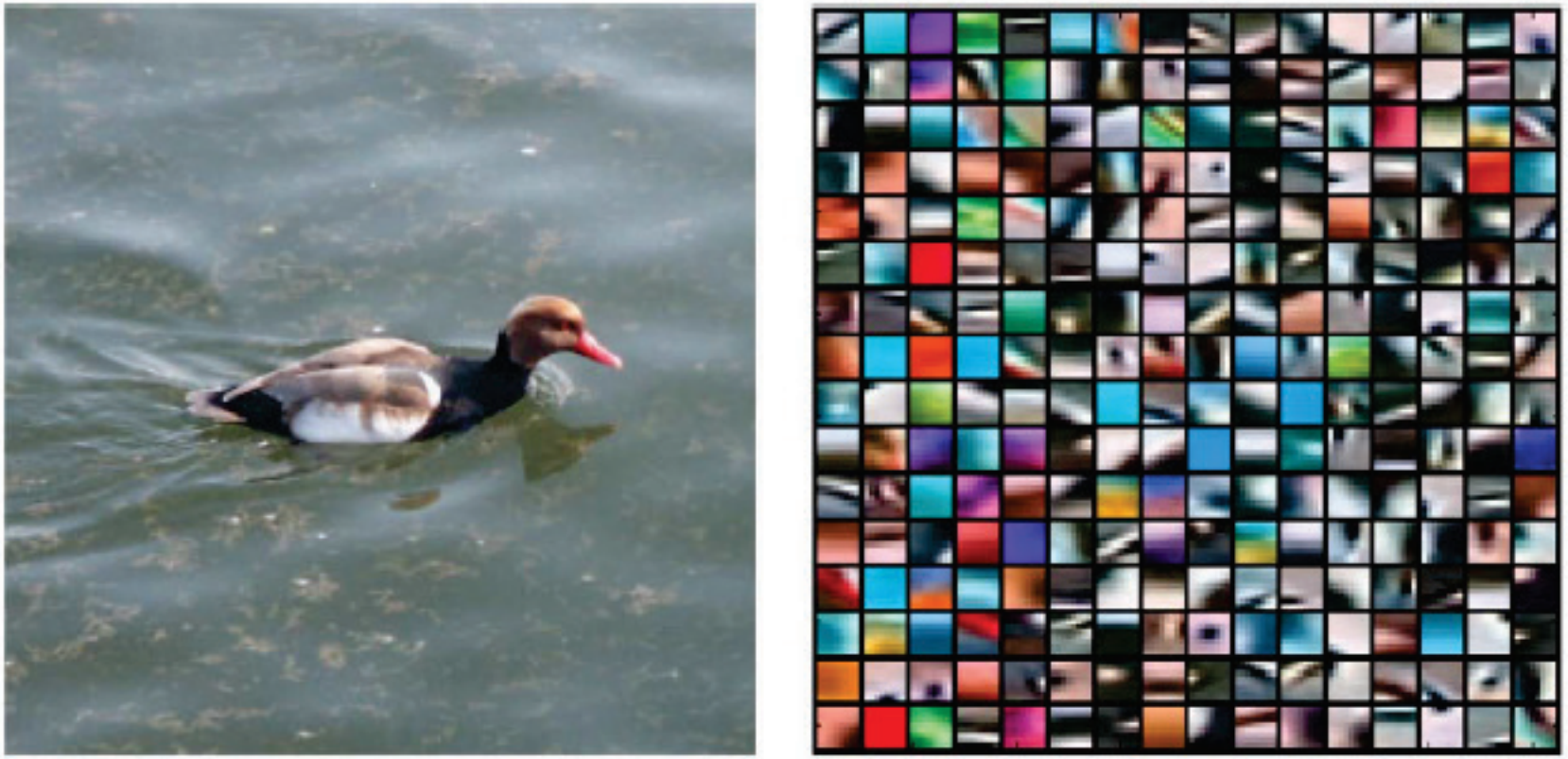


低ランク行列の学習は「ユーザ」と「映画」の低次元表現を学習することに他ならない.
→交互最適化法やトレースノルム正則化法で学習可能

推定誤差 $O\left(\frac{r(M+N)}{n}\right) \ll O\left(\frac{MN}{n}\right)$ (低ランク性を利用しない最小二乗法)

r:ランク

スパース表現，辞書学習



(a) (b)

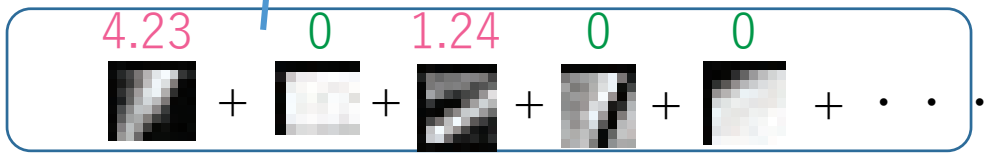
Mairal, Elad and Sapiro: Sparse Representation for Color Image Restoration. *IEEE Transactions on Image Processing*, Vol. 17, No. 1, 2008.

低ランク行列推定による辞書学習

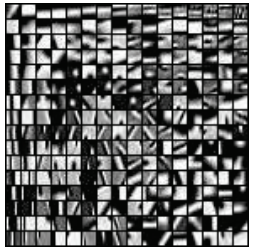
スパースコーディング

辞書（イメージパッチ）　スパースな係数　ノイズ

観測画像 $x = D\alpha + \xi$



学習された辞書



$$(\hat{D}, \hat{\alpha}) = \underset{D \in \mathbb{R}^{p \times k}, \alpha_i \in \mathbb{R}^k}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \|x_i - D\alpha_i\|^2 + \lambda \sum_{i=1}^{n} \|\alpha_i\|_1$$

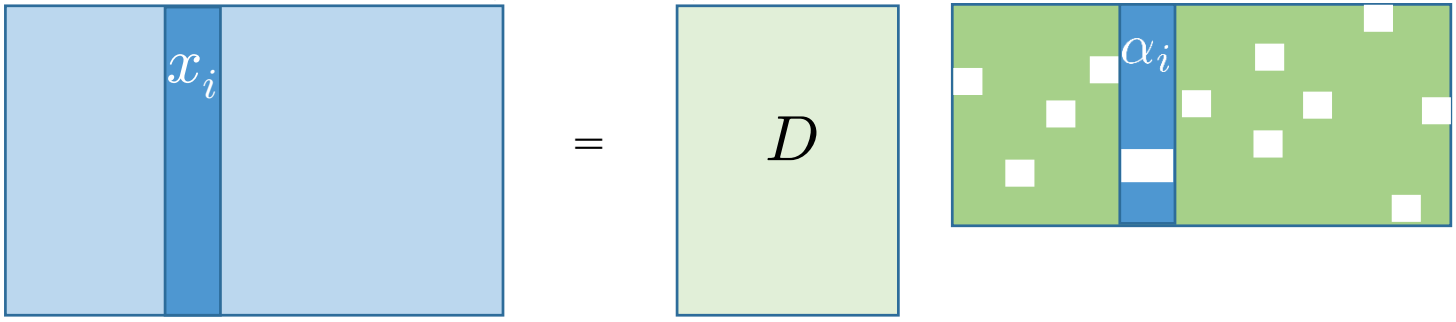
$$\text{s.t.} \quad \|D_{:,j}\| \leq 1 \quad (j = 1, \ldots, k)$$

各画像がスパースな係数で表現できるような辞書を構成

x_i $(i=1, \quad ,n)$ ：n枚の画像

α_i $(i=1, \quad ,n)$ ：n枚の画像それぞれの係数（学習対象）

D：全画像共通の辞書（学習対象）



実際はイメージパッチ（D）と係数（α）を交互に最適化して学習.

スパースコーディングを用いたデノイジング

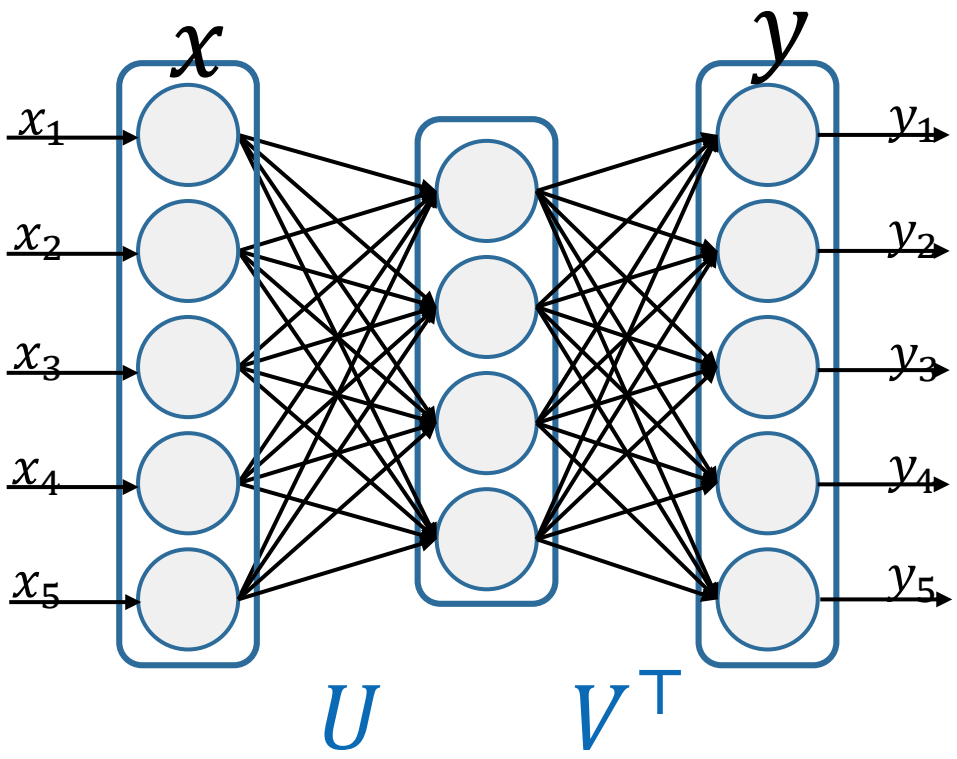
This image is taken from MLSS2012 tutorial by F. Bach.

Mairal et al.: Non-local sparse models for image restoration.
In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

スパース表現を用いた画像補完

Mairal, Elad and Sapiro: Sparse Representation for Color Image Restoration. *IEEE Transactions on Image Processing*, Vol. 17, No. 1, 2008.

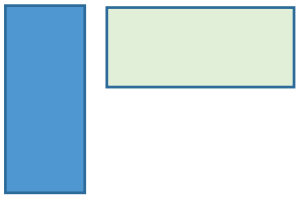
3層ニューラルネットワーク



$$f(x) = V^\top U x$$

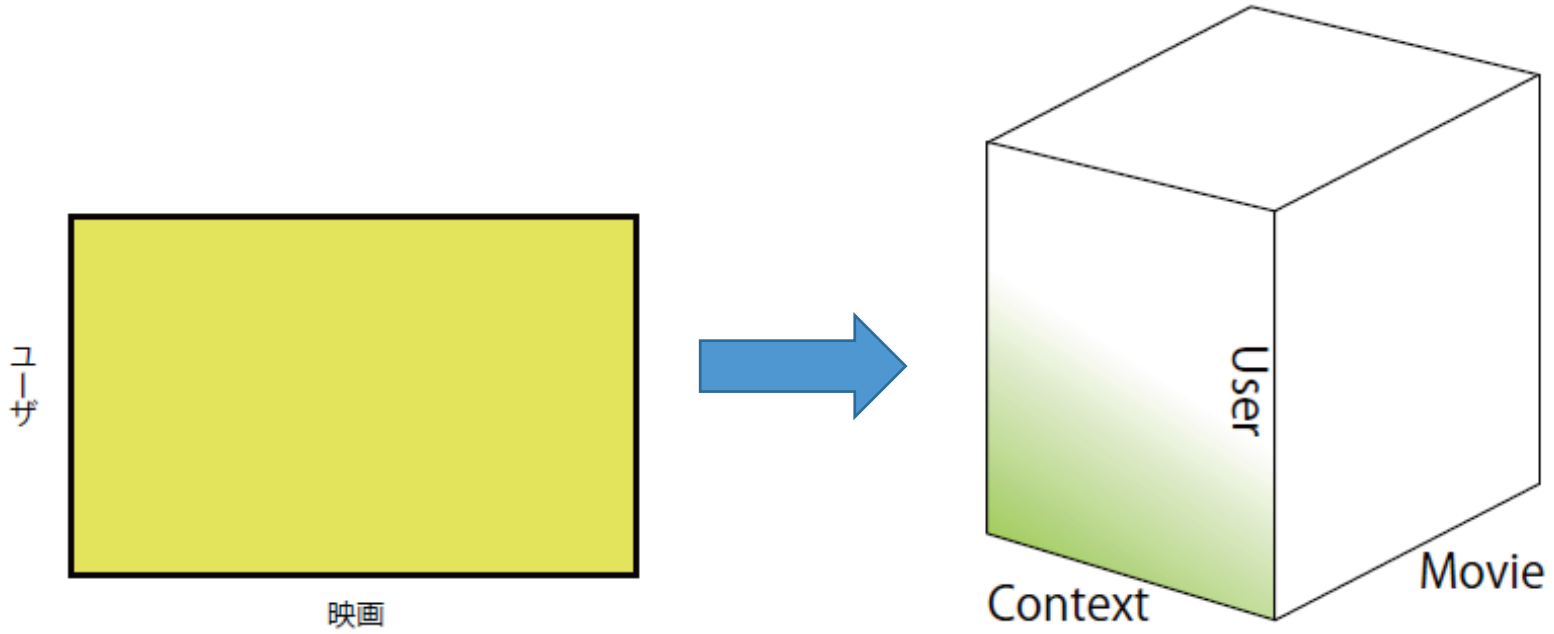
$$f(x) = V^\top h(Ux)$$

低ランク行列



- 縮小ランク回帰
- マルチタスク学習

テンソルへの拡張



行列

$$X_{ij} = \sum_{r=1}^{d} u_{r,i}^{(1)} u_{r,j}^{(2)}$$

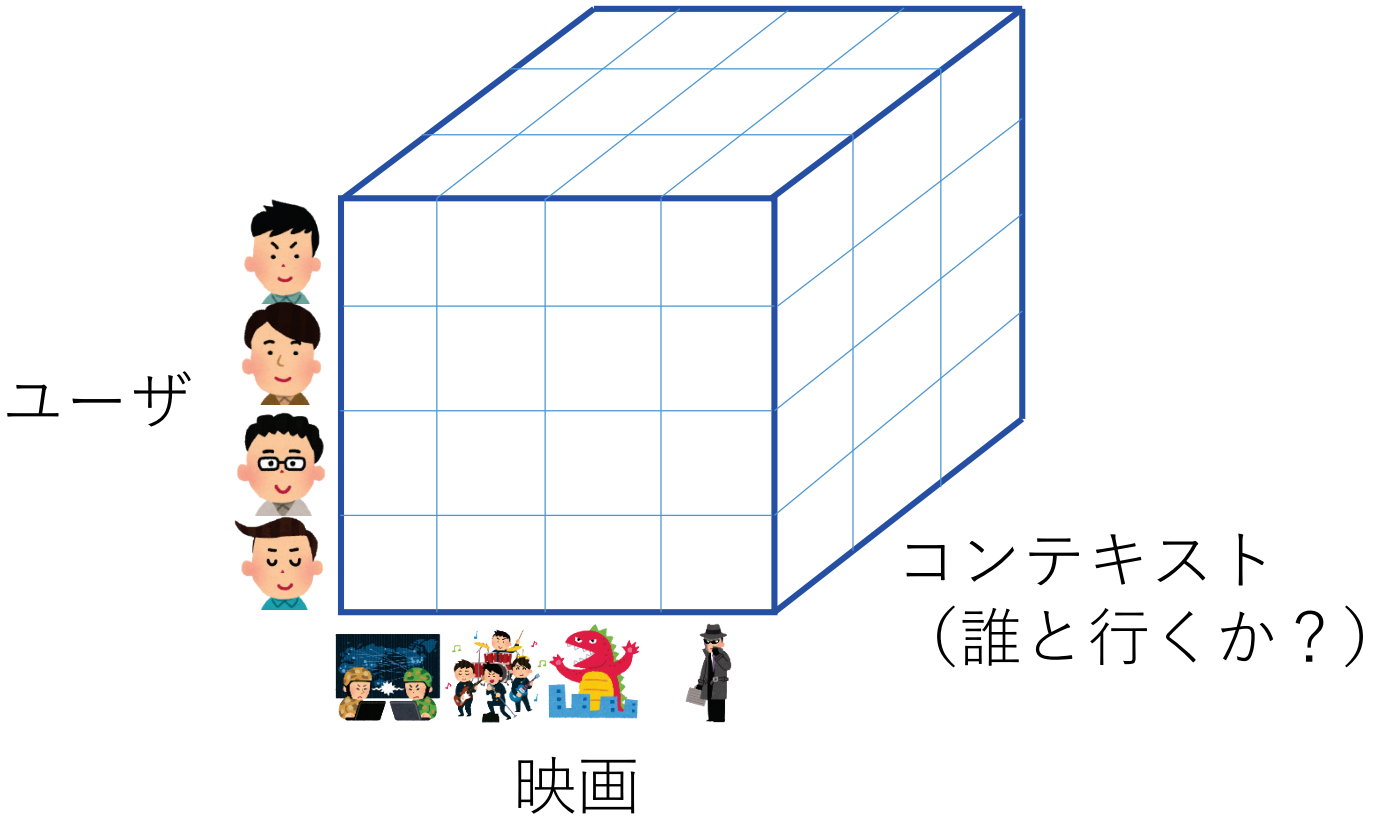
テンソル

$$X_{ijk} = \sum_{r=1}^{d} u_{r,i}^{(1)} u_{r,j}^{(2)} u_{r,k}^{(3)}$$

応用

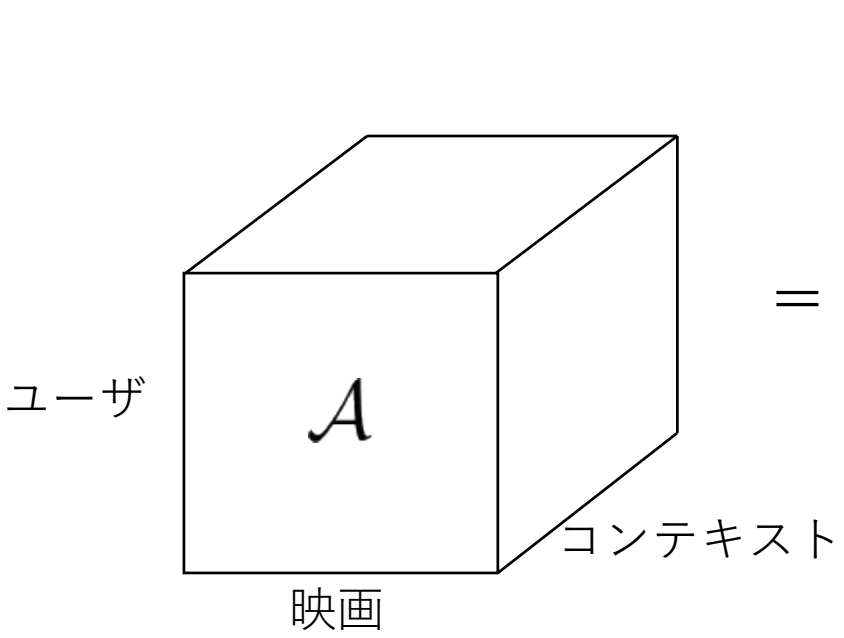
- 推薦システム
- 自然言語処理（単語のベクトル表現）
- 時空間データ解析
- 関係データ解析
- マルチタスク学習

補助情報も入れた推薦

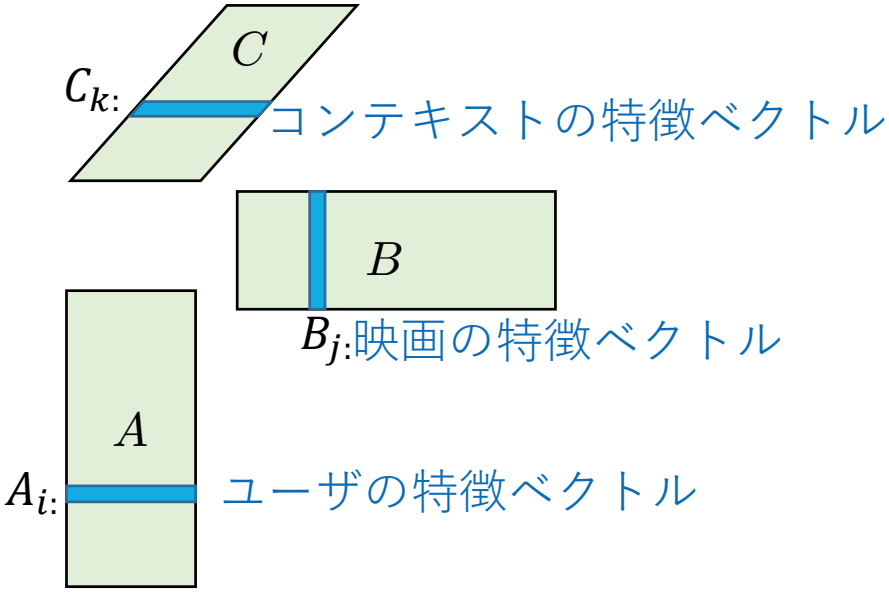


推薦システム：ユーザー×商品×季節
広告モデル：ユーザー×サイト×広告
バイクシェアリング：会員種類×時間×位置

低ランクテンソルモデル



$=$



$$\mathcal{A}_{ijk} = A_{i1}B_{j1}C_{k1} + A_{i2}B_{j2}C_{k2} + \cdots + A_{id}B_{jd}C_{kd}$$

ユーザiが持つ因子1の重み

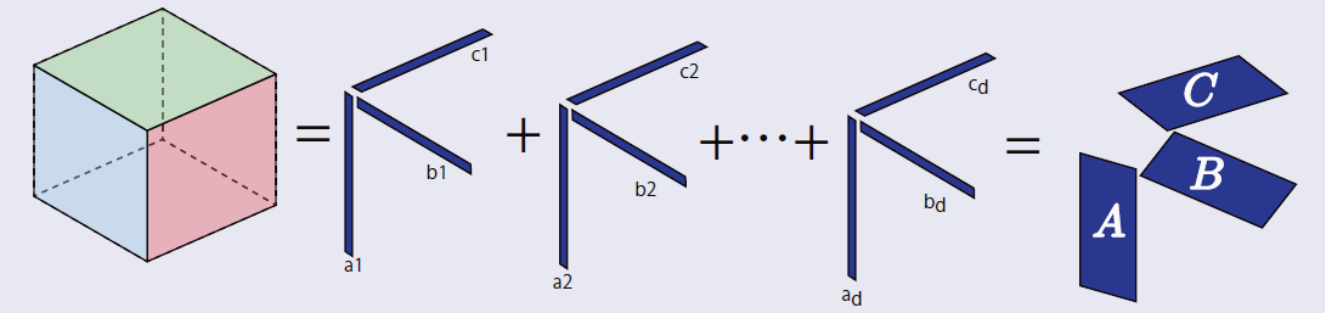
映画jが持つ因子1の重み

コンテキストkが持つ因子1の重み

A,B,Cを観察することで学習結果の解釈も可能

テンソル分解

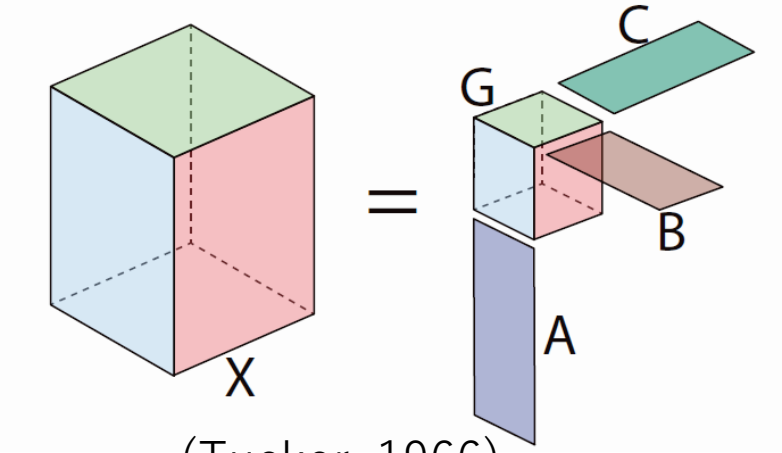
CP-分解/ランク



Canonical Polyadic 分解(Hitchcock, 1927; Hitchcock, 1927)
CANDECOMP/PARAFAC (Carroll & Chang, 1970; Harshman, 1970)

計算はNP困難，ある条件の元で分解の一意性あり

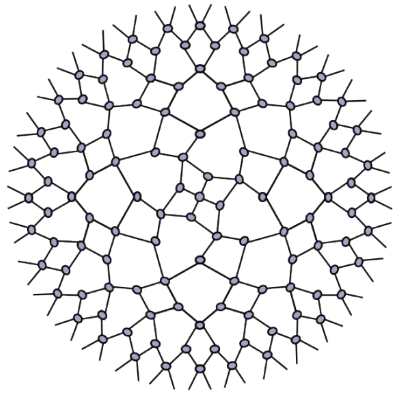
Tucker-分解/ランク



(Tucker, 1966)

特異値分解で計算可能

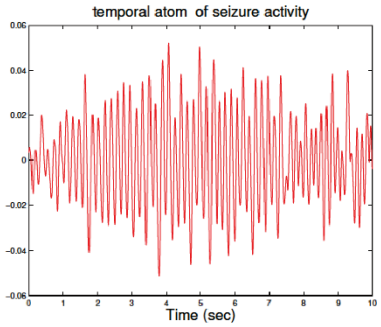
テンソルネットワーク

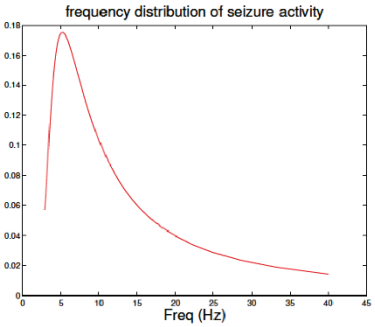


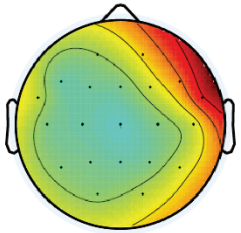
(物理学で発展)

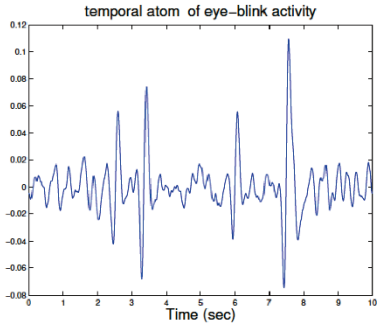
時空間解析への応用例

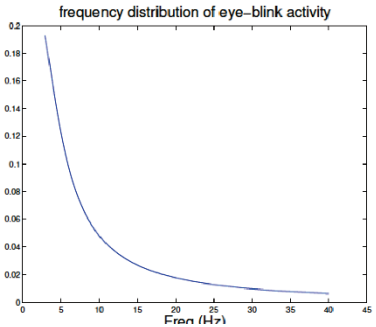
EEGデータ解析

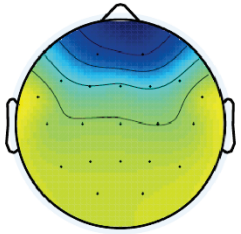










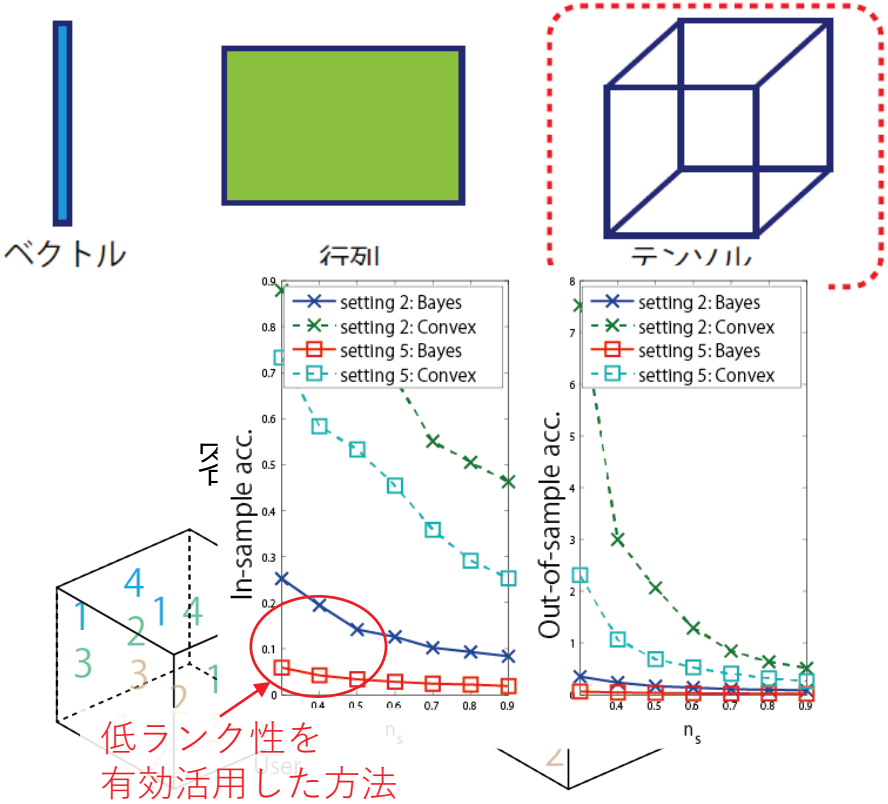


time × frequency × space

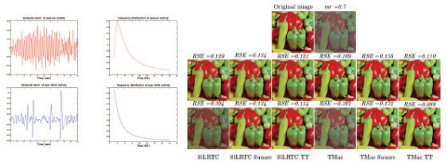
CP分解

EEG monitoring: Epileptic seizure onset localization (De Vos et al., 2007)

テンソルの学習



他の応用例：
-時空間データ解析
-画像処理
-自然言語処理
-深層学習



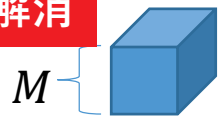
通常(最小二乗法)

M^K/n

低ランク性を利用した方法
(ベイズ推定法等)

dKM/n

次元の呪いを解消

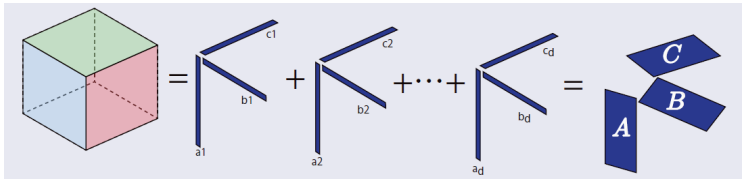


K:次元
d:ランク($\ll M$)

[Suzuki,ICML2015;Kanagawa+et al.,ICML2016;Suzuki+et al.,NIPS2016]

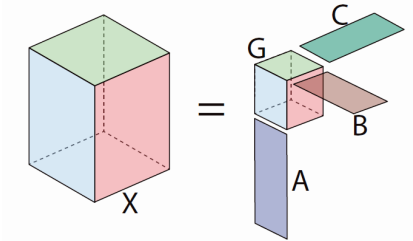
テンソルの“ランク”

CP-分解/ランク



Canonical Polyadic 分解(Hitchcock, 1927; Hitchcock, 1927)
CANDECOMP/PARAFAC (Carroll & Chang, 1970; Harshman, 1970)

Tucker-分解/ランク



(Tucker, 1966)

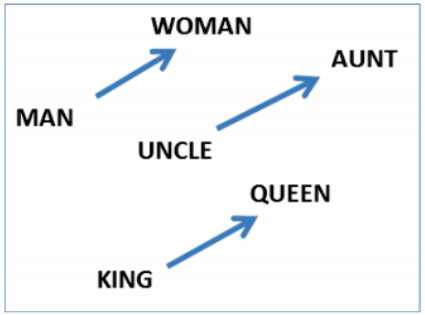
自然言語処理に現れる低ランク性

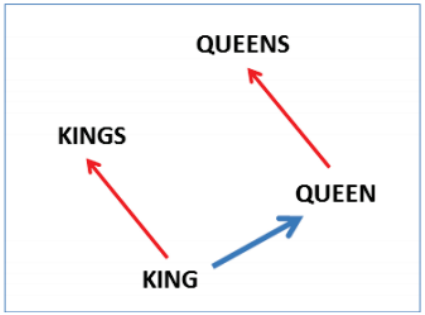
Word2vec [Mikolov et al., 2013]

- 単語のベクトル表現を得る方法

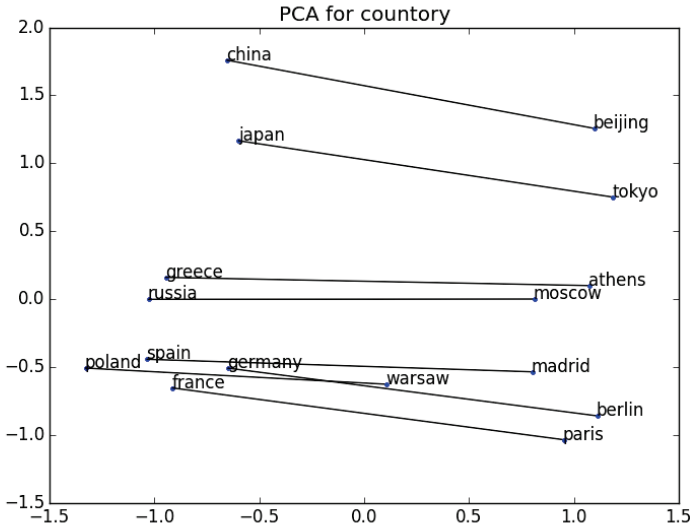
“King” – “Man” + “Woman” = “Queen”
“Tokyo” – “Japan” + “China” = “Beijing”

意味を足し引きできるような表現が得られる.



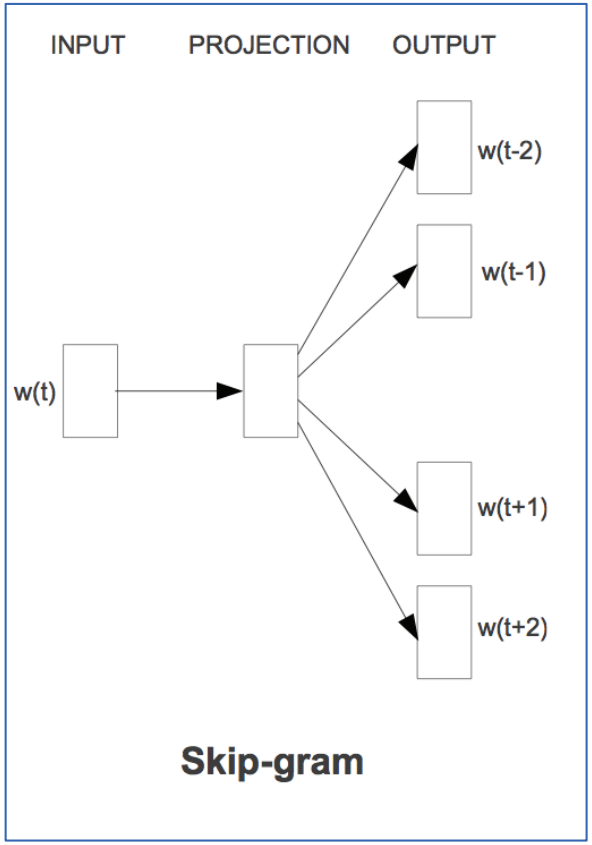


(Mikolov et al., NAACL HLT, 2013)



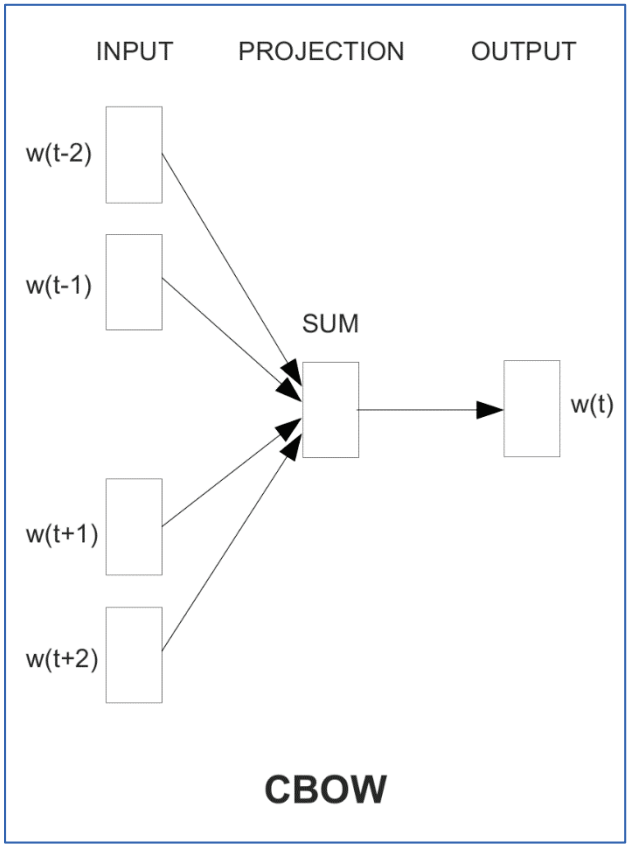
表現学習

skip-gramとContinuous Bag-of-Words (CBOW)



skip-gramモデル

ある単語のまわりに出現する
単語の確率分布をモデル化



CBOWモデル

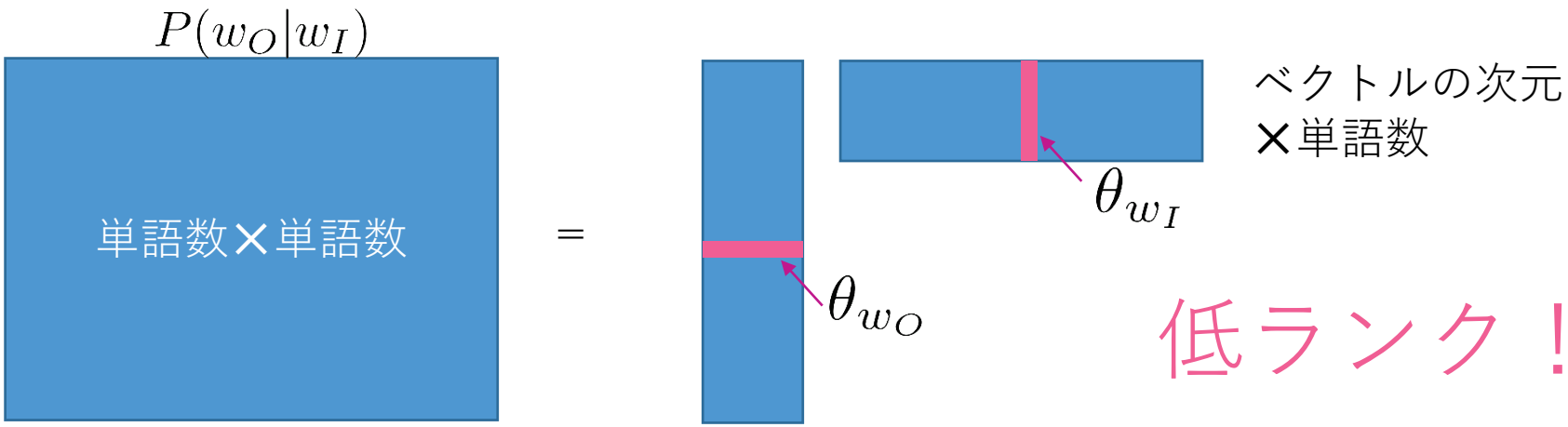
まわりの単語からその場所に
ある単語が出現する確率をモデル化

Skip-gramモデル

単語w_Oがw_Iの周辺（前後１０単語ほど）に現れる確率

$$P(w_O|w_I) = \frac{\exp(\langle \theta_{w_O}, \theta_{w_I} \rangle)}{\sum_{w'} \exp(\langle \theta_{w'}, \theta_{w_I} \rangle)}$$
$$\propto \exp(\langle \theta_{w_O}, \theta_{w_I} \rangle)$$

- 単語のベクトル表現　θ_{w_O} と θ_{w_I} の内積で表現.
- ベクトルの次元はせいぜい500ほど



実際の挙動

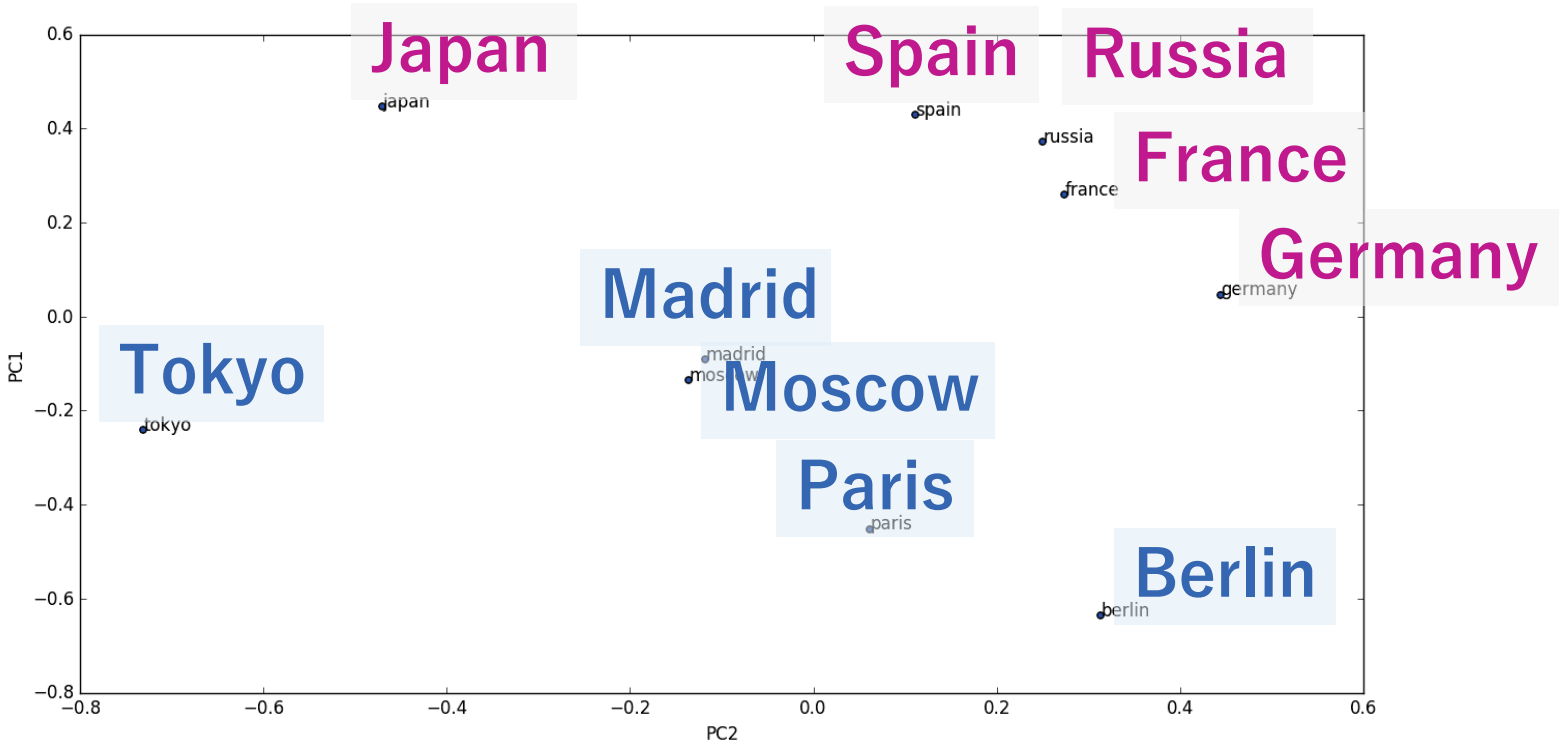

```
from gensim.models import word2vec

train_file = "./mldata/text8"

data = word2vec.Text8Corpus(train_file)
model = word2vec.Word2Vec(size=100, window=5, min_count=5, workers=7)

model.build_vocab(data)
model.train(data)
```

次元１００，前後５単語の出現頻度をモデル化，５回以下の出現単語は無視

“Queen” + “Man” - “Woman” = “King” ?

```
>>> model.most_similar(positive=['queen','man'],negative=['woman'])
[('king', 0.6050819158554077), ('scotland', 0.587989091873169), ('prince', 0.573
6681222915649), ('elizabeth', 0.571208119392395), ('lord', 0.5638244152069092),
('duchess', 0.5520190000534058), ('duke', 0.5498123168945312), ('crown', 0.54618
62087249756), ('sir', 0.544183969497 6807), ('lorraine', 0.5441141128540039)]
```

国と首都



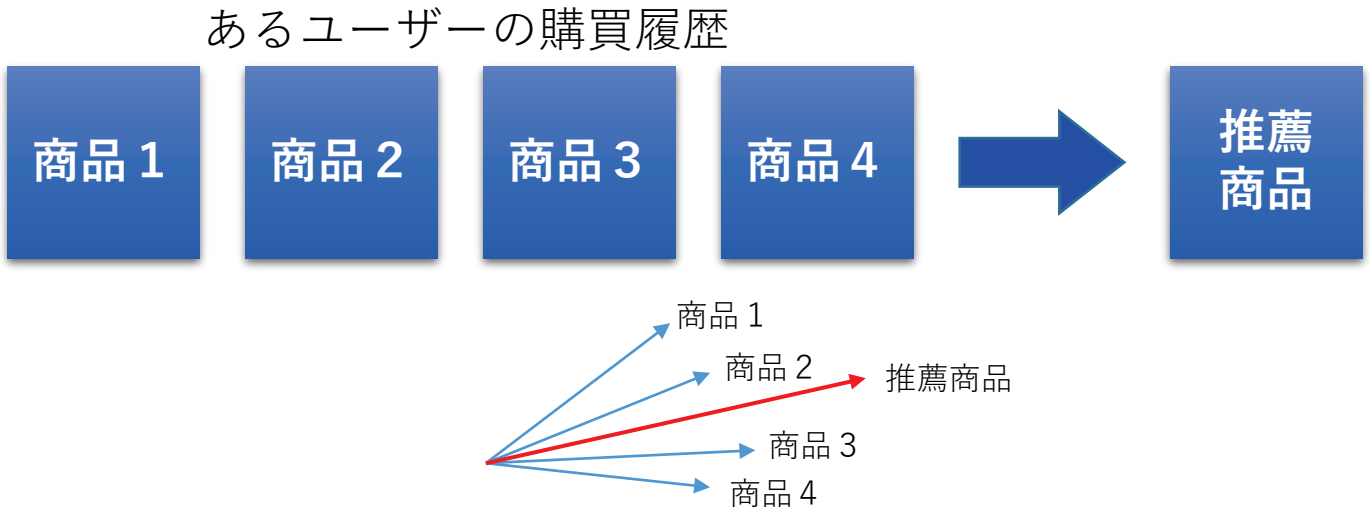
word2vecの貢献：

データの「意味」を低次元ベクトルとして表現できることを実験的に示した.

→ 深層学習にもつながる考え方.

Word2vecの応用

- 商品タグから関連した商品を推薦



- 口コミの要約

<u>文章</u>をベクトル化する手法：skip-thought, text2vec

- 記事から関連広告を推薦
- 感情分析

[Kiros, Zhu, Salakhutdinov, Zemel, Torralba, Urtasun, and Fidler: Skip-Thought Vectors. *arXiv preprint arXiv:1506.06726 (2015).*]

まとめ

- 機械学習の歴史
- 機械学習の考え方
  - 複雑な規則をデータから学ぶ
- モデルと損失
  - 学習：期待損失最小化
- 過学習の問題
  - 複雑なモデルを当てはめれば良いわけではない.
  - 正則化
  - 変数選択
- スパース高次元データ解析
  - Lasso
- 低ランク行列/テンソル分解

→深層学習にもつながる

補足資料

最適化法

$$\min_{\beta \in \mathbb{R}^p} \quad \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, x_i^\top \beta) \; + \; \psi(\beta)$$

手元にあるデータへの当てはまり（損失関数）

正則化項

記法を簡略化

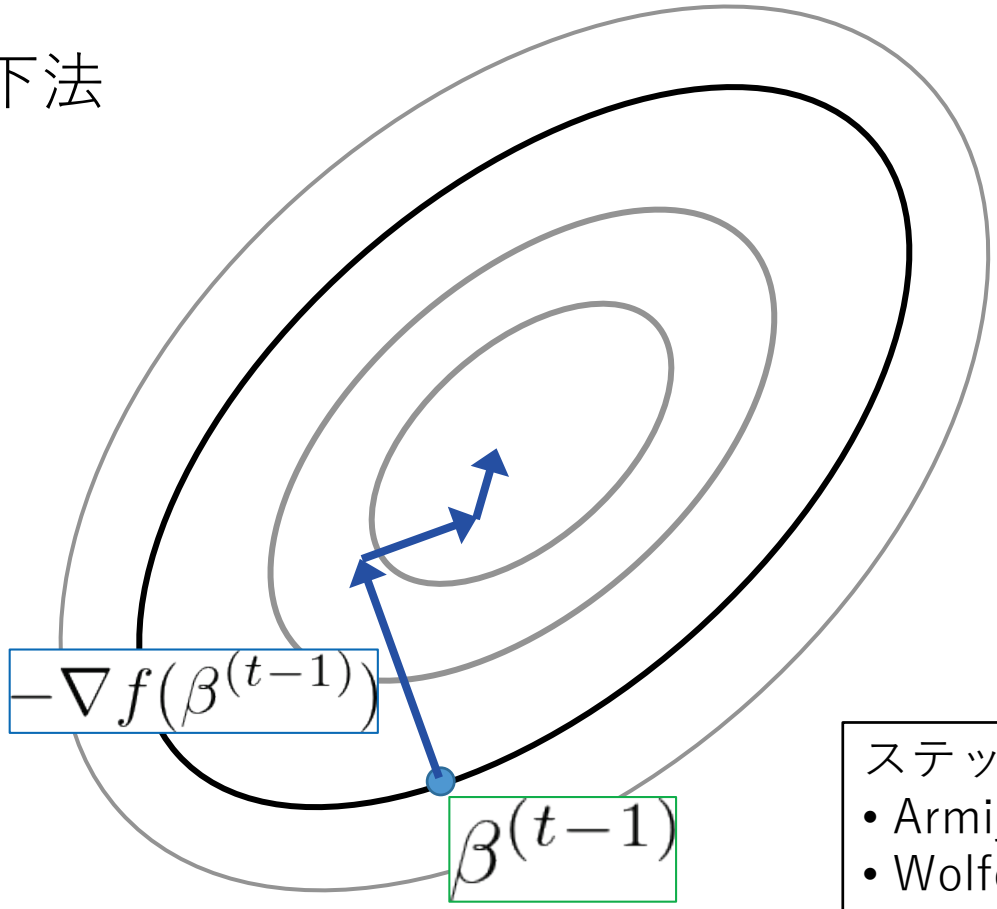
$$\min_{\beta \in \mathbb{R}^p} \quad f(\beta) + \psi(\beta)$$

最適化法：どうやってこの最適化問題を解く？

- 勾配法
- 座標降下法
- 交互方向乗数法

正則化項がない場合の最適化

- 最急降下法



ステップサイズの決定には
- Armijoの規準
- Wolfeの規準

等がある.

$$\beta^{(t)} = \beta^{(t-1)} - \alpha_t \nabla f(\beta^{(t-1)})$$

正則化項がある場合：近接勾配法

$$f(\beta) + \psi(\beta)$$

線形近似

正則化項はそのまま
（正則化項は微分不可能）

$$g_t = \nabla f(\beta^{(t-1)})$$

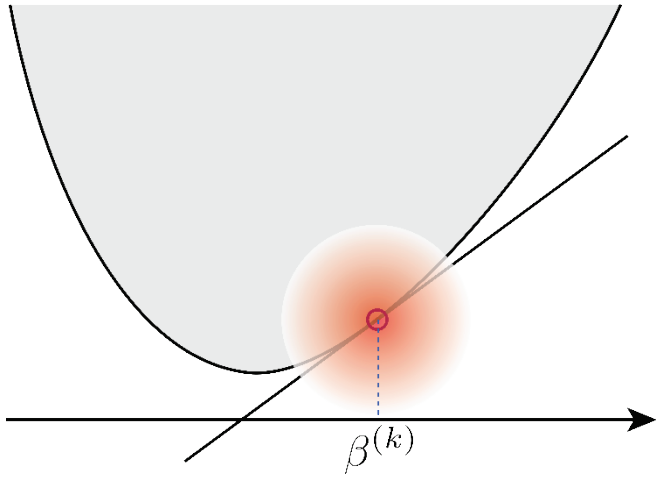
$$\beta^{(t)} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left\{ g_t^\top \beta + \psi(\beta) + \frac{\eta_t}{2} \|\beta - \beta^{(t-1)}\|^2 \right\}$$

遠くへ離れないようにする項

鍵となる計算：**近接写像**

$$\mathrm{prox}(\mathbf{q}|\psi) := \underset{\mathbf{x}}{\arg\min} \left\{ \psi(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \mathbf{q}\|^2 \right\}$$

→　L1正則化なら簡単に計算可能.
（Soft-thresholding関数）



近接写像を用いた表記

$$g_t = \nabla f(\beta^{(t-1)})$$

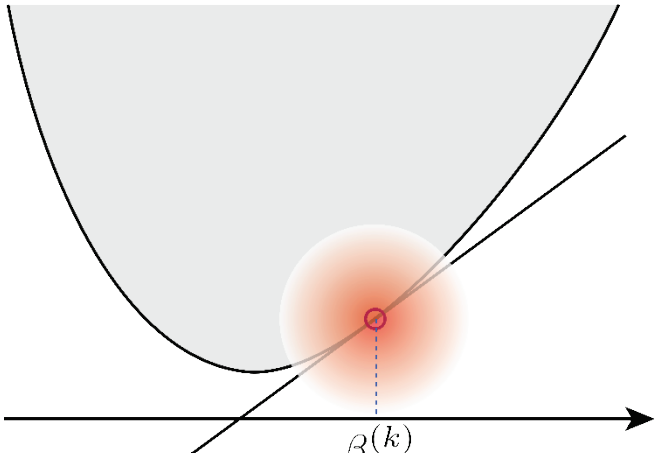
$$\begin{aligned}\beta^{(t)} &= \arg\min_{\beta\in\mathbb{R}^p}\left\{g_t^\top\beta + \psi(\beta) + \frac{\eta_t}{2}\|\beta-\beta^{(t-1)}\|^2\right\}\\ &= \arg\min_{\beta\in\mathbb{R}^p}\left\{\frac{\psi(\beta)}{\eta_t} + \frac{1}{2}\left\|\beta-\left(\beta^{(t-1)}-\frac{g_t}{\eta_t}\right)\right\|^2\right\}\\ &= \mathrm{prox}(\beta^{(t-1)} - g_t/\eta_t|\psi/\eta_t)\end{aligned}$$

- Ψ=0なら単なる最急降下法
- Ψがある凸集合の標示関数なら射影勾配法

鍵となる計算：**近接写像**

$$\mathrm{prox}(\mathbf{q}|\psi) := \arg\min_{\mathbf{x}}\left\{\psi(\mathbf{x}) + \frac{1}{2}\|\mathbf{x}-\mathbf{q}\|^2\right\}$$

→ L1正則化なら簡単に計算可能.
（Soft-thresholding関数）



具体例：L1正則化

座標ごとにわかれている！

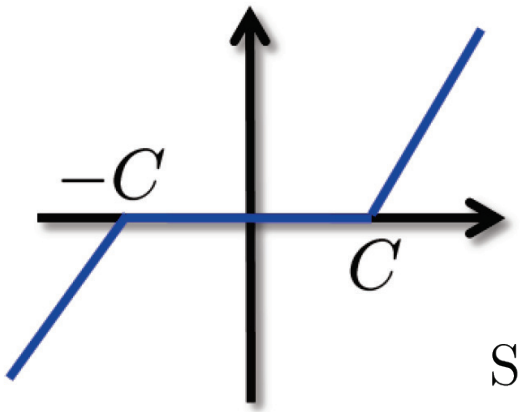
$$\beta_j^{(t)} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \left\{ g_{t,j} + \lambda |\beta_j| + \frac{\eta_t}{2} (\beta_j - \beta_j^{(t-1)})^2 \right\}$$

$$\begin{aligned} \mathrm{prox}(\mathbf{q}|C\|\cdot\|_1) &= \underset{\mathbf{x}}{\arg\min} \left\{ C\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}-\mathbf{q}\|^2 \right\} \\ &= (\mathrm{sign}(q_j)\max(|q_j| - C, 0))_j. \end{aligned}$$

$\mathrm{sign}(q_j)\max(|q_j| - C, 0)$ を $\mathrm{ST}_C(q_j)$ とおく

$$\beta_j^{(t)} = \mathrm{ST}_{\lambda/\eta_t}\left(\beta_j^{(t-1)} - \frac{g_{t,j}}{\eta_t}\right)$$

解が陽に書ける！



$\mathrm{ST}_C(q_j)$のグラフ

近接勾配法の収束速度

$$\beta^{(t)} = \mathrm{prox}(\beta^{(t-1)} - g_t/\eta_t | \psi/\eta_t)$$

f の性質	μ-強凸	非強凸
L-平滑	$\exp\left(-t\frac{\mu}{L}\right)$	$\frac{L}{t}$
非平滑	$\frac{1}{\mu t}$	$\frac{1}{\sqrt{t}}$

- 滑らかなほど・強凸なほど速い.
- Nesterov の加速法を用いれば滑らかな場合に速くなる (Nesterov, 2007, Zhang et al., 2010).
- 上のオーダーは勾配情報のみを用いる方法 (First order method) の中で最適.

η_t の設定	強凸	非強凸
平滑	L	L
非平滑	$\frac{\mu t}{2}$	$\sqrt{t}$

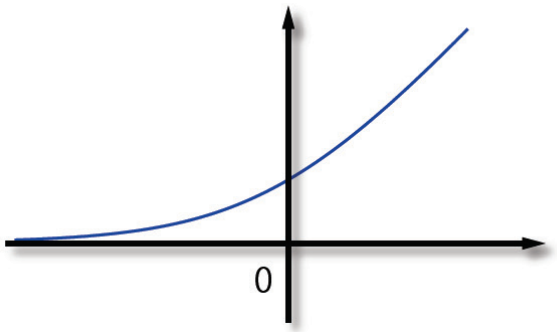
強凸性と平滑性

- 平滑性： 勾配の変化がゆっくり

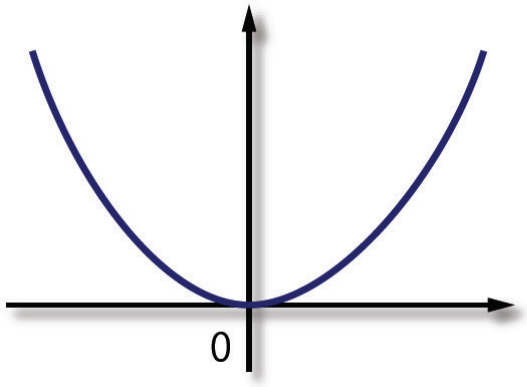
$$\|\nabla f(x) - \nabla f(x')\| \leq L\|x - x'\|$$

- 強凸性： 2 次関数以上に曲がっている

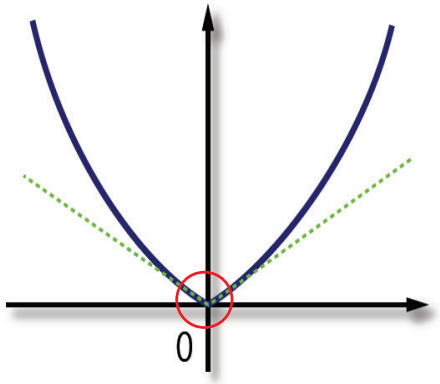
$$\frac{\mu}{2}\theta(1-\theta)\|x-y\|^2 + f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$$



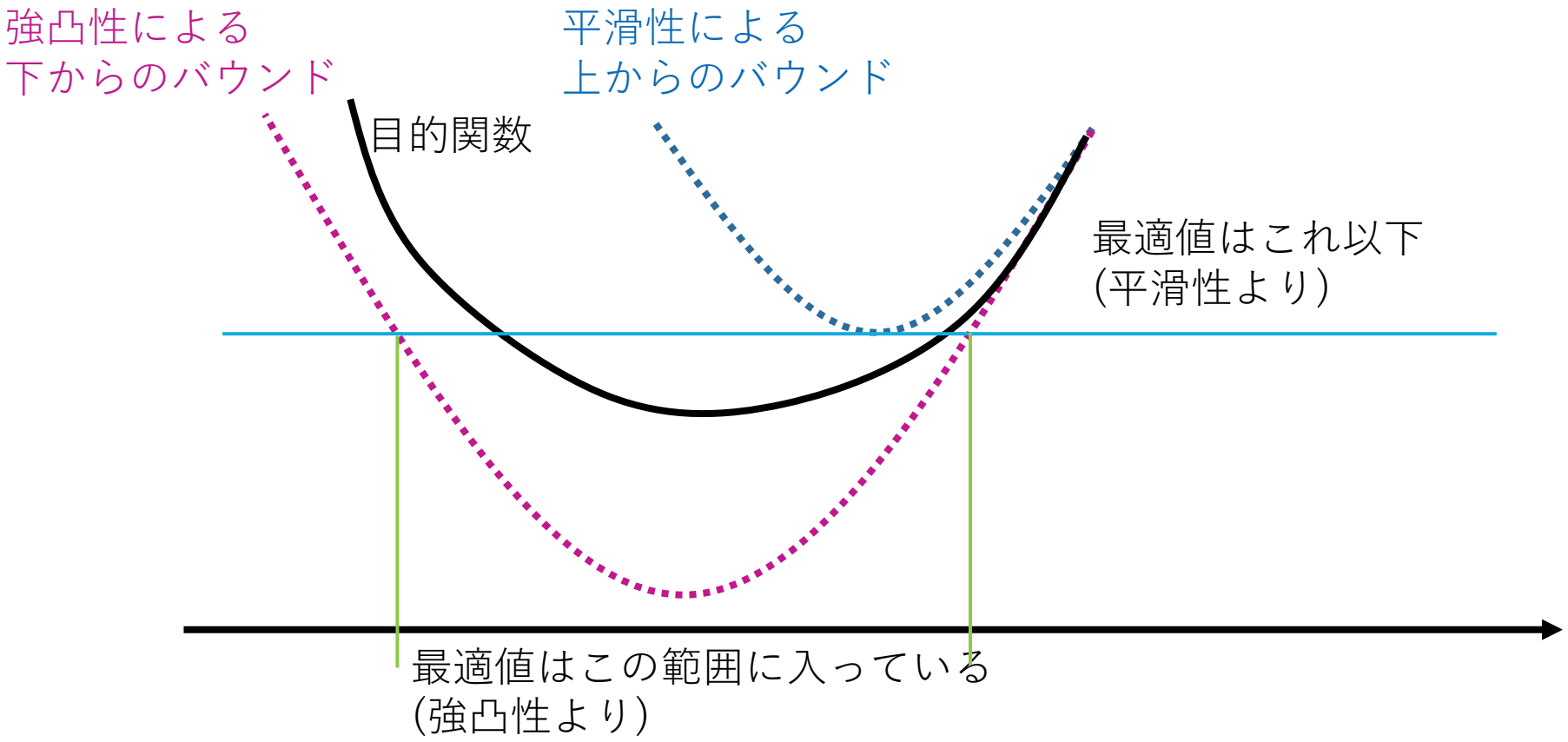
平滑だが強凸ではない



平滑かつ強凸



強凸だが平滑ではない



平滑性→最適値を上から抑えられる.
強凸性→最適値の範囲を限定できる.

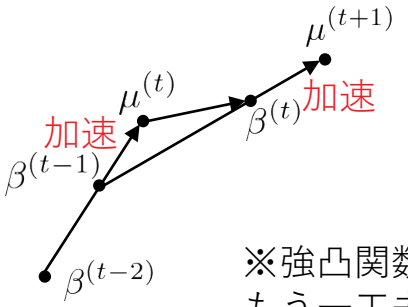
近接勾配法の収束速度

Nesterovの加速法

$$\beta^{(t)} = \mathrm{prox}(\mu^{(t)} - g_t/\eta|\psi/\eta)$$

$$s_{t+1} = \frac{1+\sqrt{1+4s_t^2}}{2}$$

$$\mu^{(t+1)} = \beta^{(t)} + \left(\frac{s_t - 1}{s_{t+1}}\right)(\beta^{(t)} - \beta^{(t-1)})$$



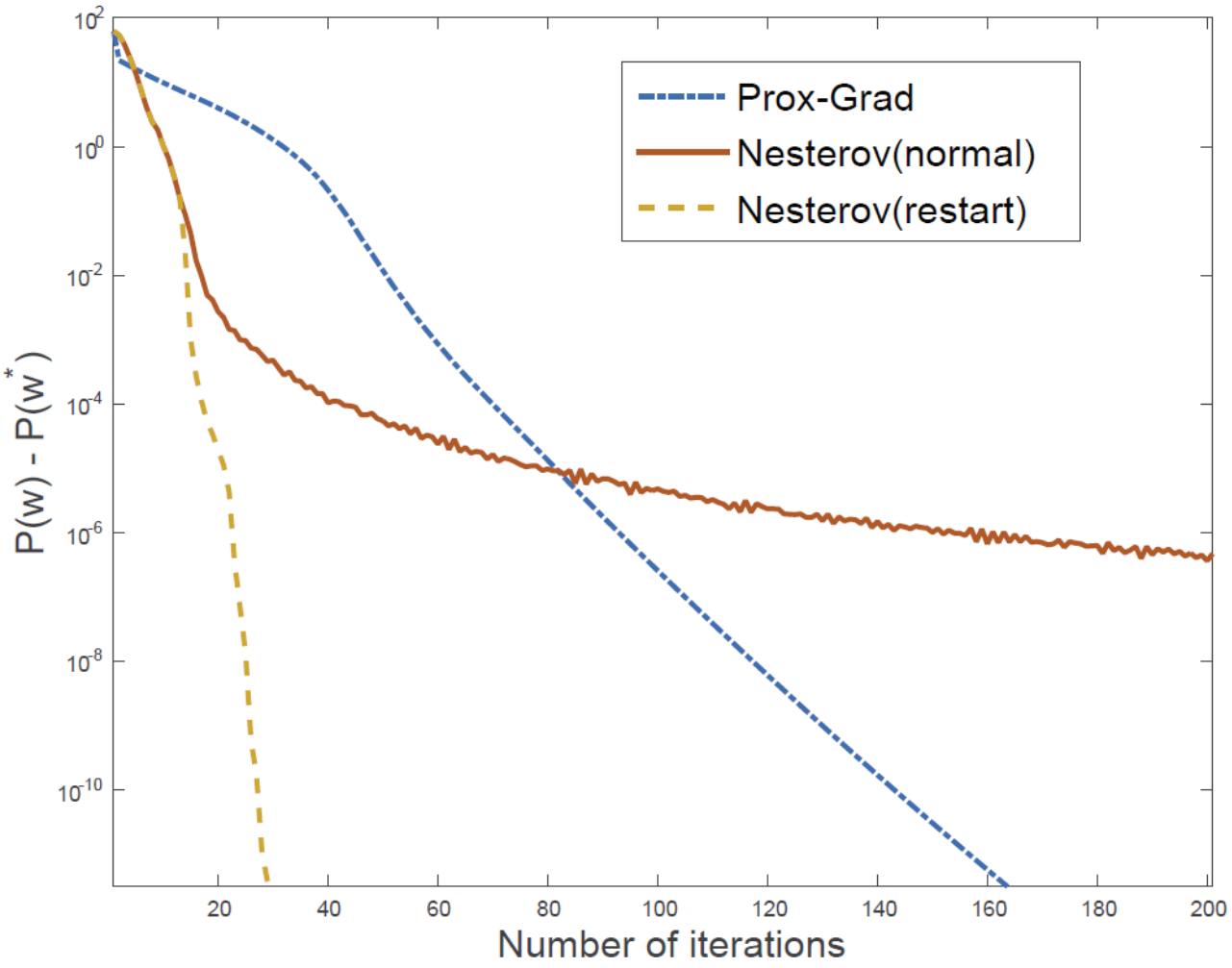
※強凸関数にはもう一工夫必要

f の性質	μ-強凸	非強凸
L-平滑	$\exp\left(-t\sqrt{\frac{\mu}{L}}\right)$	$\frac{L}{t^2}$
非平滑	$\frac{1}{\mu t}$	$\frac{1}{\sqrt{t}}$

- 滑らかなほど・強凸なほど速い.
- Nesterov の加速法を用いれば滑らかな場合に速くなる (Nesterov, 2007, Zhang et al., 2010).
- 上のオーダーは勾配情報のみを用いる方法 (First order method) の中で最適.

η_t の設定	強凸	非強凸
平滑	L	L
非平滑	$\frac{\mu t}{2}$	$\sqrt{t}$

数値実験例



サンプルサイズ$n = 700$，次元$p = 1000$，100成分のみ非ゼロ

総計算量

経験損失はO(1/n)まで下げる必要がある．（汎化誤差ミニマックスレートがO(1/n)）

$\kappa = \frac{L}{\mu}$ ：条件数

一回の更新にかかる計算量
$O(n)$

$\nabla f(\beta) = \sum_{i=1}^{n} \nabla_\beta \ell(z_i, \beta)$：$n$個の和

×

1/nまで下げるのに必要な更新回数
$O\left(\sqrt{\kappa}\log(n)\right)$

Nesterovの加速法

＝

総計算量
$O(n\sqrt{\kappa}\log(n))$

- サンプルサイズnが強く効いてくる
- 大規模データの最適化が難しい

→ 確率的最適化が有用