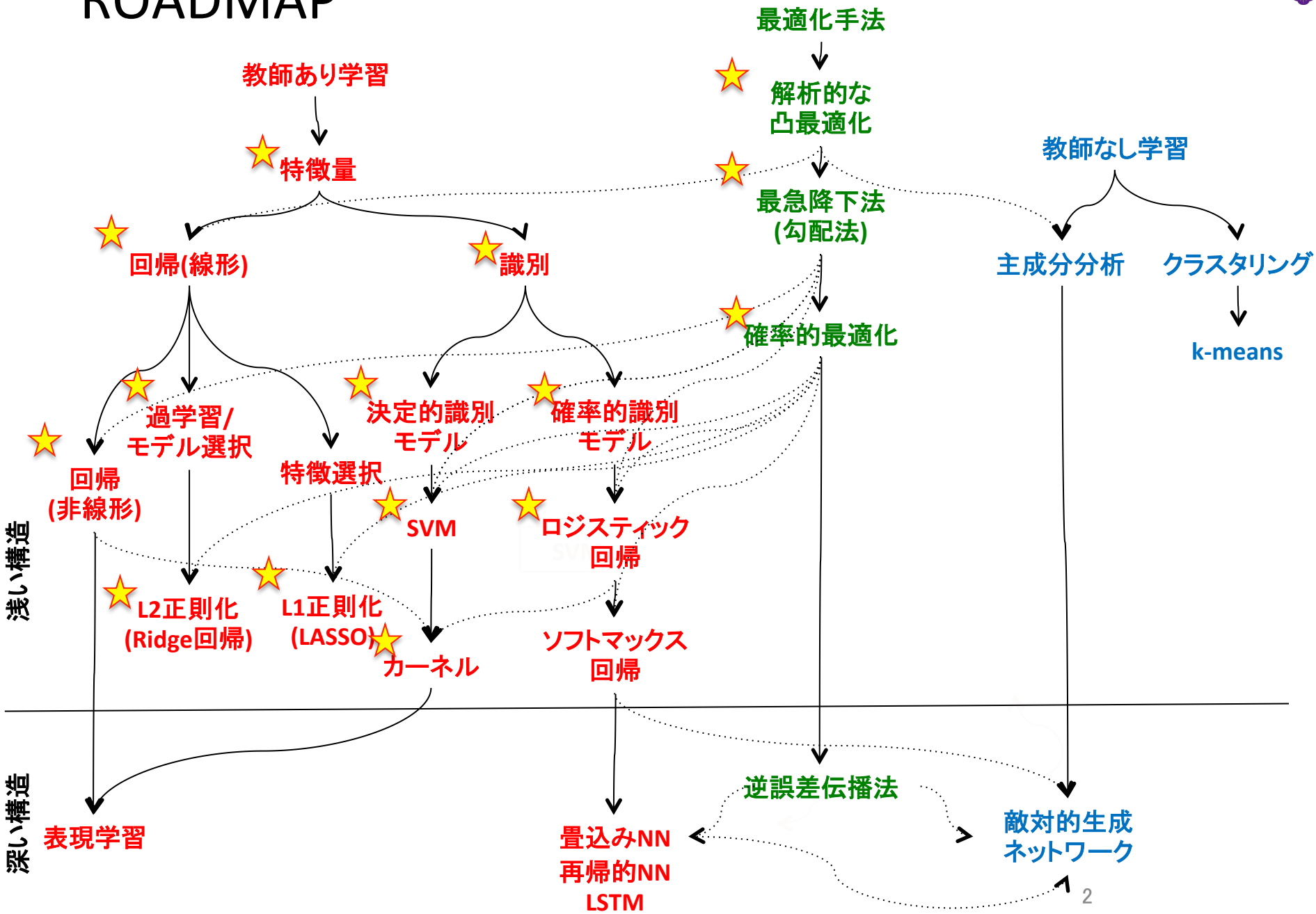
機械学習(7)
確率的識別モデル2/経験損失最小化

情報科学類　佐久間　淳

ROADMAP


最適化手法
教師あり学習
解析的な
凸最適化
特徴量
教師なし学習
最急降下法
(勾配法)
回帰(線形)
識別
主成分分析
クラスタリング
確率的最適化
k-means
過学習/
モデル選択
決定的識別
モデル
確率的識別
モデル
回帰
(非線形)
特徴選択
SVM
ロジスティック
回帰
浅い構造
L2正則化
(Ridge回帰)
L1正則化
(LASSO)
カーネル
ソフトマックス
回帰
逆誤差伝播法
表現学習
畳込みNN
再帰的NN
LSTM
敵対的生成
ネットワーク
深い構造

再掲

凸計画問題

- 凸計画問題
  - 目的関数が凸関数
  - 実行可能領域が凸集合

凸計画問題	制約なし	等式制約	不等式制約
fが微分可能	fの微分を0にする点が解析的に求まる ⇒その点が最適解 求まらない ⇒最急降下法など	ラグランジュの未定乗数法	ラグランジュの未定乗数法 今回は扱わない (SVM導出などで出てくる)
fが微分不可能	2,3週後に触れます	今回は扱わない	今回は扱わない

交差エントロピー誤差関数の最小化

- 尤度関数最大化の代わりに負の対数尤度関数(この場合, 交差エントロピー誤差関数と呼ぶ)を最小化

$$E(\boldsymbol{w}) = -\log L(\boldsymbol{w}) = -\sum_{n=1}^{N} \{t_n \log \hat{t}_n + (1 - t_n) \log(1 - \hat{t}_n)\}$$

$$\hat{t}_n = \sigma(\boldsymbol{w}^T \boldsymbol{x}_n)$$

- まずは最急降下法で最適化

演習

7.1　ロジスティック回帰の最急降下法

$\boldsymbol{x}_i = \begin{pmatrix} 1 \\ x_{i1} \\ \vdots \\ x_{iD} \end{pmatrix}, \boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_N^T \end{pmatrix}, \boldsymbol{t} = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{pmatrix}, \boldsymbol{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_D \end{pmatrix}$ とする。 $t_i \in \{0, 1\}$ である。事例群 $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^N$ を使ってロジスティック回帰モデル $t = \sigma(\boldsymbol{w}^T \boldsymbol{x})$ を最急降下法を使って求めることを考える。目的関数 (交差エントロピー損失) は以下の通り。

$$E(\boldsymbol{w}) = -\log L(\boldsymbol{w}) = -\sum_{n=1}^{N} \{t_n \log \hat{t}_n + (1 - t_n) \log(1 - \hat{t}_n)\} \tag{3}$$

ここで $\hat{t}_n = \sigma(\boldsymbol{w}^T \boldsymbol{x}_n)$ である。

1. 交差エントロピー損失の $\boldsymbol{w}$ についての勾配が以下であることを示せ

$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} (\hat{t}_n - t_n) \boldsymbol{x}_n = \boldsymbol{X}^T (\hat{\boldsymbol{t}} - \boldsymbol{t}) \tag{4}$$

2. ステップサイズパラメータが η であるとき, 最急降下法の更新式を示せ

最急降下法をもう少し真面目に

- 目的関数 $f(\boldsymbol{x})$ の最小化を考える
- 一次のテイラー展開
  - $f(\boldsymbol{x}+\boldsymbol{\delta}) \simeq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{\delta}$
  - x周辺ではfは線形関数だと考える
- 最急降下法 $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{x}^{(t)} - \eta \nabla f(\boldsymbol{x}^{(t)})$

$$
\begin{aligned}
f(\boldsymbol{x}^{(t+1)}) &= f(\boldsymbol{x}^{(t)} - \eta \nabla f(\boldsymbol{x}^{(t)})) \\
&\simeq f(\boldsymbol{x}^{(t)}) - \eta \nabla f(\boldsymbol{x}^{(t)})^T \nabla f(\boldsymbol{x}^{(t)}) \\
&< f(\boldsymbol{x}^{(t)})
\end{aligned}
$$

ηが十分小さければ最急降下法による更新は目的関数値を減少させる

ヘッセ行列

- 勾配：点 $\boldsymbol{x}$ において,最も大きく $f(\boldsymbol{x})$ が減る方向
  - D次元関数の勾配はD次元ベクトル

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_D} \right)^T$$

- ヘッセ行列(ヘシアン)
  - D次元関数の二階微分
  - D次元関数のヘッセ行列はD×D行列
  - $\boldsymbol{x}$ でのヘシアン：$H_{\boldsymbol{x}}$ や $\nabla\nabla_{\boldsymbol{x}} f$ などと書く

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

http://ja.wikipedia.org/wiki/%E3%83%98%E3%83%83%E3%82%BB%E8%A1%8C%E5%88%97

テイラー展開による近似をより精密にしたら？

- 二次のテイラー展開
  - $f(\boldsymbol{x}+\boldsymbol{\delta}) \simeq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T \boldsymbol{H}_{\boldsymbol{x}} \boldsymbol{\delta}$
  - $\boldsymbol{H}_{\boldsymbol{x}}$ はヘッセ行列 (多変数関数の二階微分)
  - x周辺ではfは凸二次関数だと考える

$$g(\boldsymbol{x}) = f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^T \boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^T \boldsymbol{H}_{\boldsymbol{x}} \boldsymbol{\delta}$$

- この近似を信じる場合、x周辺のfの最小化

$$\nabla g(\boldsymbol{x}) = \boldsymbol{0}$$
$$\Longleftrightarrow \nabla f(\boldsymbol{x}) + \boldsymbol{H}_{\boldsymbol{x}} \boldsymbol{\delta} = \boldsymbol{0}$$
$$\Longleftrightarrow \delta = -\boldsymbol{H}_{\boldsymbol{x}}^{-1} \nabla f(\boldsymbol{x})$$

よってこのように更新すればx周辺の最小値に更新される(はず)

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{\delta}$$
$$= \boldsymbol{x} - \boldsymbol{H}_{\boldsymbol{x}}^{-1} \nabla f(\boldsymbol{x})$$

ニュートン法

- 更新式 $\boldsymbol{x}^{(t+1)} \leftarrow \boldsymbol{x}^{(t)} - H_{\boldsymbol{x}^{(t)}}^{-1} \nabla_x f(\boldsymbol{x}^{(t)})$

$$(H_{\boldsymbol{x}})_{ij} = \frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}$$

  - 逆行列計算が入っているので計算は重い
  - でも二階微分まで考慮しているので効率が良い

演習

7.2　ロジスティック回帰のニュートン法

定義と記法は 7.1 節に準じます。

1. 交差エントロピー損失の $\boldsymbol{w}$ についての ヘシアン行列が以下であることを示せ

$$\boldsymbol{H} = \boldsymbol{X}^T \boldsymbol{R} \boldsymbol{X} \tag{5}$$

ただし, $\boldsymbol{R}$ は $N \times N$ 対角行列で、第 n 対角要素は $\boldsymbol{R}_{nn} = \hat{t}_n(1 - \hat{t}_n)$.

2. ロジスティック回帰のニュートン法によるパラメータ $\boldsymbol{w}$ の更新式を $\boldsymbol{X}$ および $\boldsymbol{t}, \hat{\boldsymbol{t}}, \boldsymbol{R}$ の式で表せ.

ロジスティック回帰のニュートン法による最適化

- 交差エントロピー損失
$$E(\boldsymbol{w}) = -\log L(\boldsymbol{w}) = -\sum_{n=1}^{N} \{t_n \log \hat{t}_n + (1-t_n)\log(1-\hat{t}_n)\}$$
- 勾配
$$\nabla E(\boldsymbol{w}) = \sum_{n=1}^{N} (\hat{t}_n - t_n)\boldsymbol{x}_n = \boldsymbol{X}^T(\hat{\boldsymbol{t}} - \boldsymbol{t})$$
- ヘッセ行列 $\nabla\nabla E(\boldsymbol{w}) = \boldsymbol{X}^T\boldsymbol{R}\boldsymbol{X}$
ただし$\boldsymbol{R}$ は対角で $R_{nn} = \hat{t}_n(1-\hat{t}_n)$

- よってニュートン法による更新式は
$$\boldsymbol{w}' = \boldsymbol{w} - (\boldsymbol{X}^T\boldsymbol{R}\boldsymbol{X})^{-1}\boldsymbol{X}^T(\hat{\boldsymbol{t}} - \boldsymbol{t})$$

7.3　ロジスティック回帰における勾配法とニュートン法の比較

プログラム iris_LR_GD.ipynb および iris_LR_Newton.ipynb を実行して以下の問題に答えよ。IRIS データは分類問題用のデータであり、特徴量はがく片長 (Sepal Length)，がく片幅 (Sepal Width)，花びら長 (Petal Length)，花びら幅 (Petal Width) の 4 次元ベクトルである。いずれも連続変数である。目標値はセトナ (setosa)、バーシクル (versicolor)、 バージニカ (virginica) という 3 種類のあやめである。ここでは 2 値分類とするため、virginica か、そうでないか、を予測することとする。

プログラムは Iris データのロジスティック回帰における分類を、勾配法 (GD) とニュートン法 (Newton) で学習し、その学習曲線 (更新回数 vs 目標関数値) を表示する。GD では更新前後のコストの改善が 0.00000001 を下回った時に学習を終了する。GD の一つ目のセクションではステップサイズが 0.0001 の場合に学習曲線を表示している。GD の二つ目のセクションではステップサイズを [0.1, 0.01, 0.008, 0.006, 0.004, 0.003, 0.002, 0.001, 0.0005] と変化させた時の学習曲線を表示している。

1. 勾配法 (GD) において、ステップサイズが 0.003 より大きいとき、目的関数が増加するのはなぜか

2. 勾配法 (GD) において、ステップサイズが 0.002 より小さい場合の、更新回数、実行時間の関係を説明しなさい。ステップサイズをある以上小さくすると、最終的な目的関数値 (final cost) が増加するのはなぜか。

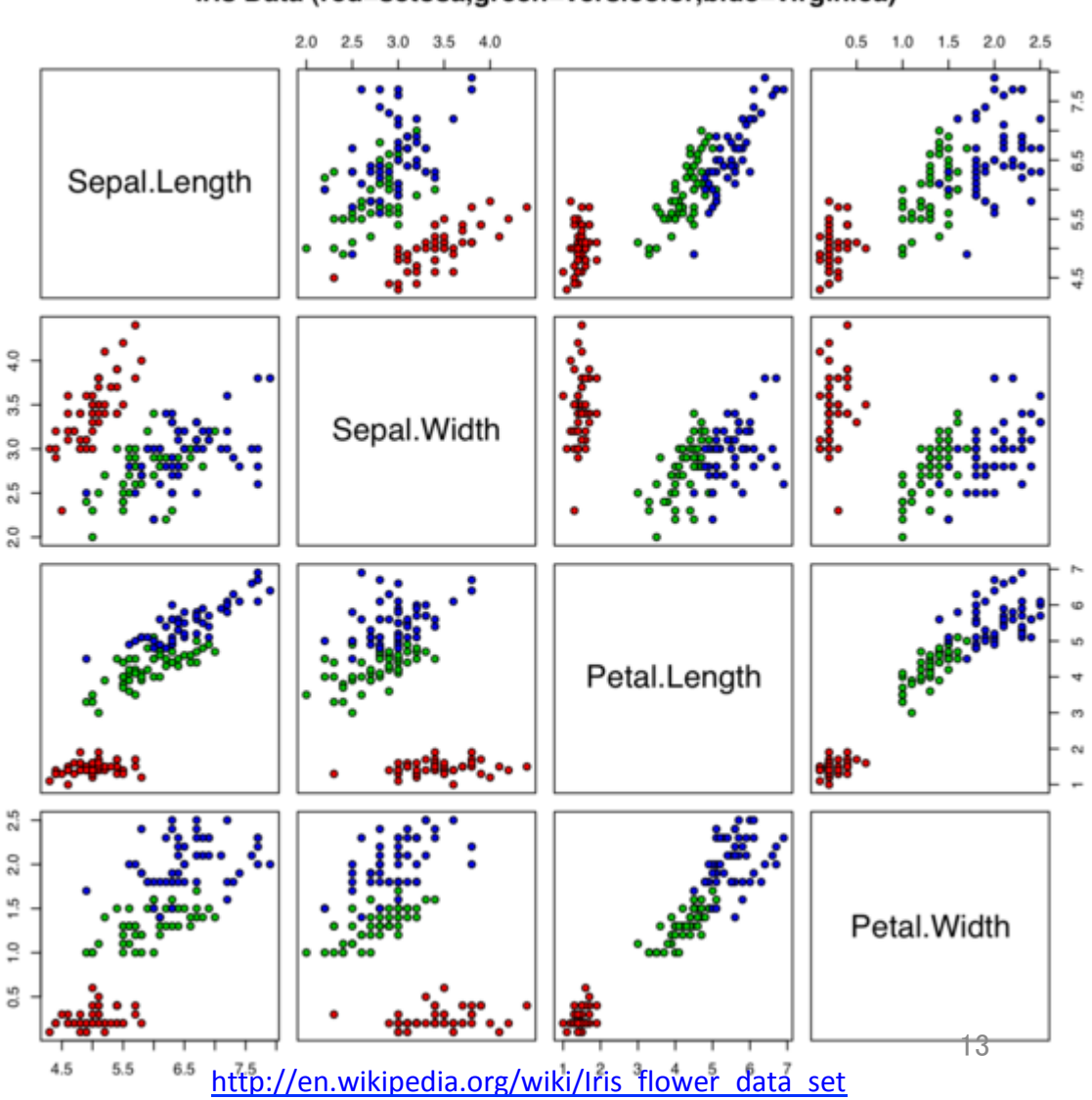
3. 勾配法とニュートン法を、最終的な目的関数値、更新回数、実行時間の観点から比較せよ

4. ニュートン法はステップサイズパラメータを持たないがこのデータにおいて最適化は適切に実行されている。なぜか。

5. 最適化においてニュートン法が適しているケースと最急降下法が適しているケースについて考察しなさい。

参考：IRISデータ

- IRISデータ
- *setosa*，*versicolor*, *virginica,* 3種のあやめ
- 萼片長，萼片幅，花弁長,花弁幅，4属性値



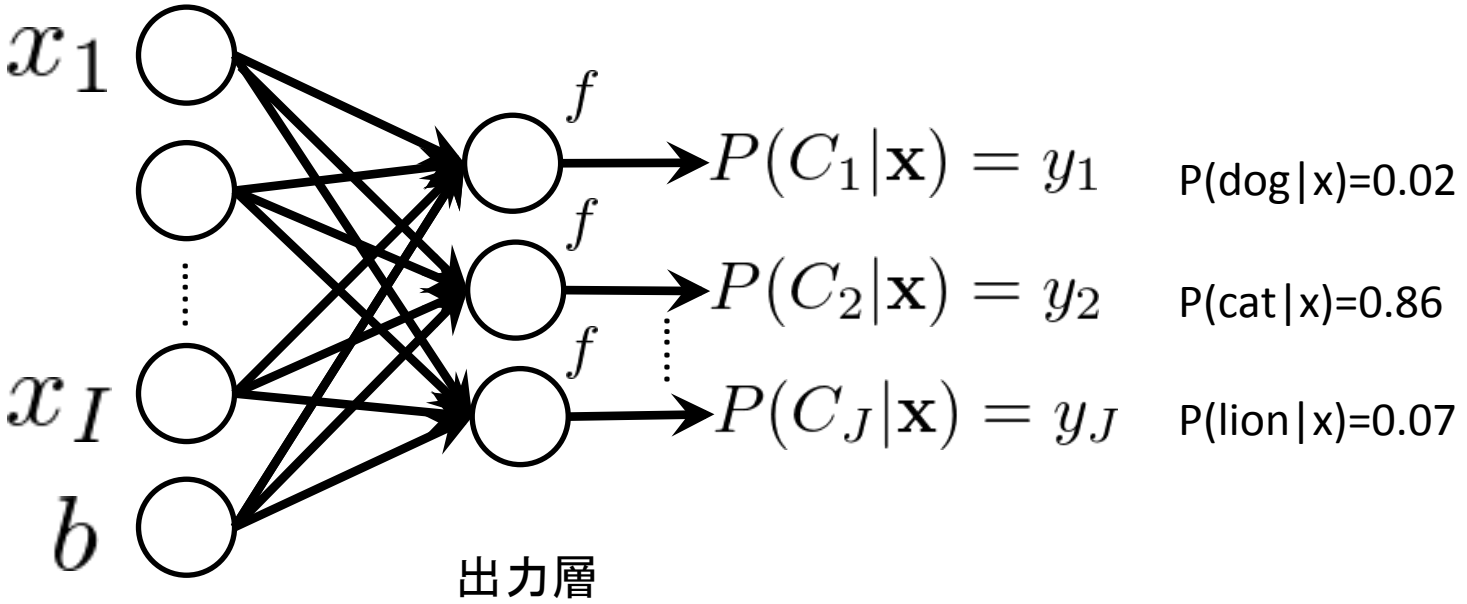
http://en.wikipedia.org/wiki/Iris_flower_data_set

確率的識別モデルまとめ

- 確率的識別モデルではラベルを {0,1}とする
- 確率的識別モデルではラベルそのもの{0,1}の代わりにP(0|x), P(1|x)を予測する
- [0,1]を値域とする確率を予測したいので、w^Txの代わりに$\sigma(w^Tx)$を予測モデルとする
- [0,1]を値域とする確率について誤差を定義したいので、二乗誤差の代わりに交差エントロピー損失を最小化する
- 交差エントロピー損失は凸関数で微分はできるけど，勾配がゼロになる点を解析的に求めることができないので，最急降下法(or 二階微分まで考えたニュートン法)で最適化する

多クラス分類(softmax回帰)

- 二値分類から多クラス分類へ



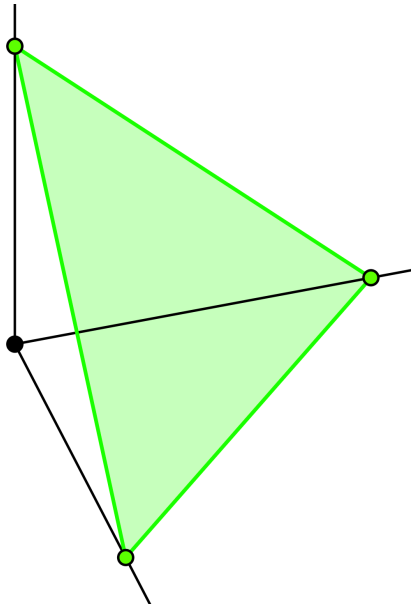
P(dog|x)=0.02

P(cat|x)=0.86

P(lion|x)=0.07

From wikipedia, **CC BY-SA 3.0, CC BY 2.0**

softmax関数



- Sigmoid関数の多ラベル拡張
- softmax関数のj番目の出力　(j=1,...,K)

$$\sigma(\boldsymbol{z})_j = \frac{\exp(z_j)}{\sum_d \exp(z_d)}$$

- $\mathbb{R}^D$をK-1シンプレクスに写像する
  - K-1シンプレクス上の任意の点は、K次元確率ベクトル

$$\sum_k \sigma(\boldsymbol{z})_k = 1$$

- スムーズなmax関数と考えて良い

$$\boldsymbol{x}^T = (1, 2, 3, 4, 1, 2, 3)$$

$$\sigma(\boldsymbol{x})^T = (0.024, 0.064, 0.175, 0.475, 0.024, 0.064, 0.175)$$

https://en.wikipedia.org/wiki/Softmax_function

softmax回帰

- 訓練データの目標値
  - E.g. 手書き数字 0,1,…,9 の10クラス
  - 1-ok-k表記 (1 hot vector) $\boldsymbol{t}_n = [0\,0\,1\,0\,0\,0\,0\,0\,0\,0]^T$
- 予測 $\hat{t}_{nk} = \Pr(k|\boldsymbol{x}_n) = \dfrac{\exp(\boldsymbol{w}_k^T \boldsymbol{x}_n)}{\sum_{k=1}^{K} \exp(\boldsymbol{w}_k^T \boldsymbol{x}_n)}$
- 訓練データ集合に対する尤度

$$L(\boldsymbol{w}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \Pr(k|\boldsymbol{x}_n)^{t_{n,k}} = \prod_{n=1}^{N} \prod_{k=1}^{K} (\hat{t}_{n,k})^{t_{n,k}}$$

- 誤差関数(負の対数尤度)は交差エントロピー

$$E(\mathbf{w}) = -\ln L(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln \hat{t}_{nk}$$

Softmax回帰と交差エントロピー

- 2クラス分類の場合

ロジスティック回帰 $\hat{t}_n = \sigma(\boldsymbol{w}^T \boldsymbol{x}_n)$

$$E(\boldsymbol{w}) = -\ln L(\boldsymbol{w}) = -\sum_{n=1}^{N} \left\{ t_n \ln \hat{t}_n + (1 - t_n) \ln(1 - \hat{t}_n) \right\}$$

- Kクラス分類の場合 同様にニュートン法などで最小化する

$$E(\mathbf{w}) = -\ln L(\mathbf{w}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln \hat{t}_{nk}$$

softmax回帰

$$\hat{t}_{nk} = \Pr(k|\boldsymbol{x}_n) = \frac{\exp(\boldsymbol{w}_k^T \boldsymbol{x}_n)}{\sum_{k=1}^{K} \exp(\boldsymbol{w}_k^T \boldsymbol{x}_n)}$$

演習

7.4 softmax 回帰

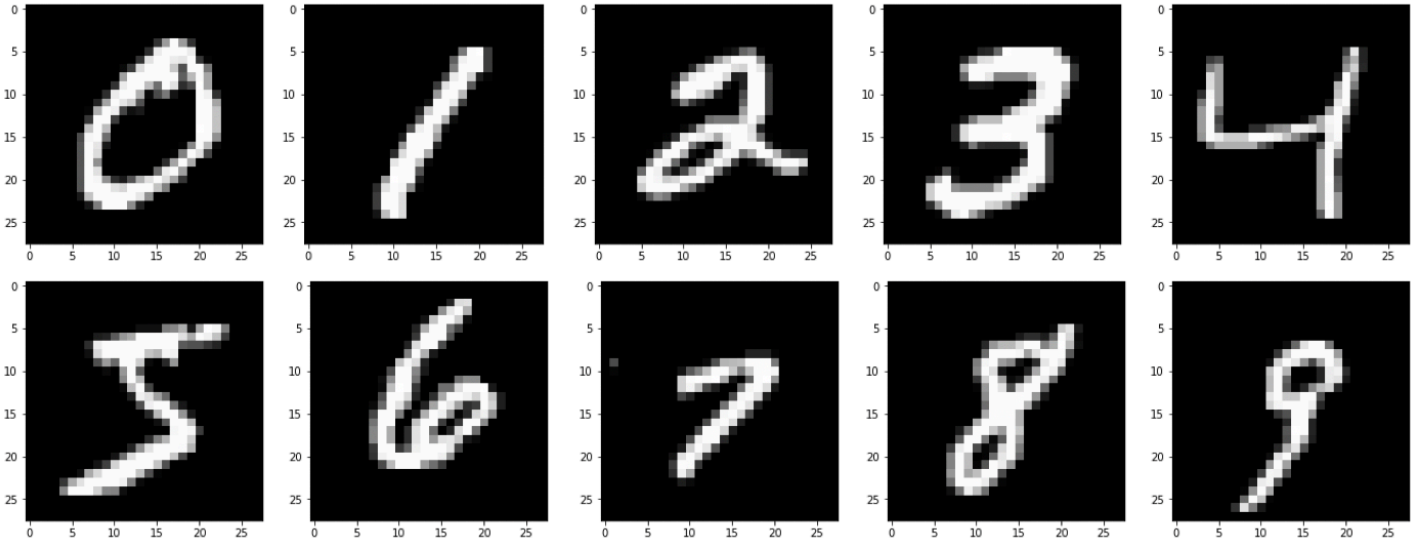
プログラム mnist_softmax.ipynb を実行して以下の問題に答えよ。プログラムは Mnist データ (0-9 の手書き文字分類) を読み込み、10 クラス分類の交差エントロピーを最小化による softmax 回帰を学習する。

1. 3 番目のセクションでは、任意のテストデータのインデックスを指定し、その画像と、softmax 回帰への入力の出力を見ることができる。index=6 の画像が, 4 と識別される確率と, 8 と識別される確率をそれぞれ求めよ。ここで u_k は softmax 回帰における $\boldsymbol{w}^T\boldsymbol{x}$ の値を示している。

2. 4 番目のセクションでは、クラスを指定したときに、学習した softmax 回帰において、テストデータの画像とそのデータが指定したクラスとに識別される確率を示している。条件付き確率が閾値 θ を超えた時 ($\Pr(t=5|\boldsymbol{x}) > \theta$), 画像 $\boldsymbol{x}$ をクラス t として認識するものとする。プログラム実行結果の「5」の認識において、精度を 0.9 以上にするもっとも小さい θ はいくらか。小数点以下第 4 位までの範囲で答えよ

(参考)MNISTデータ

- MNISTデータ(0～9の手書き数字)
- データサイズは28×28, 各要素は[0,255]の輝度
- Softmax回帰: 784次元ベクトルに変換
- 畳み込みNN: 28*28次元行列のまま扱う(あとでやる)



ロジスティック損失

- 交差エントロピー損失
  - これを全てのサンプルについて足し合わせると負の対数尤度

$$\ell(t, \hat{t}) = -t \log \hat{t} + (1 - t) \log(1 - \hat{t}), t \in \{0, 1\}$$

- ロジスティック損失
  - $\ell(t_i, \boldsymbol{w}^T \boldsymbol{x}_i) = \log(1 + \exp(-t_i \boldsymbol{w}^T (\boldsymbol{x}_i)))$

ロジスティック回帰は、ロジスティック損失の総損失最小化ともみなせる →演習

演習

7.5　ロジスティック損失とクロスエントロピー損失

定義と記法は 7.1 節に準じる。ただし混乱を防ぐためにラベルが $\{0, 1\}$ で与えられる場合 (クロスエントロピー損失) には t_i, ラベルが $\{-1, 1\}$ で与えられる場合 (ロジスティック損失) には y_i と表記する。クロスエントロピー損失に基づく目的関数は

$$E_{\text{cross}}(\boldsymbol{w}) = \sum_{i=1}^{N} -t_i \log \hat{t}_i - (1 - t_i) \log(1 - \hat{t}_i) \tag{6}$$

ロジスティック損失に基づく目的関数は

$$E_{\text{logistic}}(\boldsymbol{w}) = \sum_{i=1}^{N} \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}) \tag{7}$$

で与えられる。両者を最適化した解は一致することを示せ。

(ヒント) ラベル $t_i = 0$ と $y_i = -1$, $t_i = 1$ と $y_i = 1$ がそれぞれ対応している。

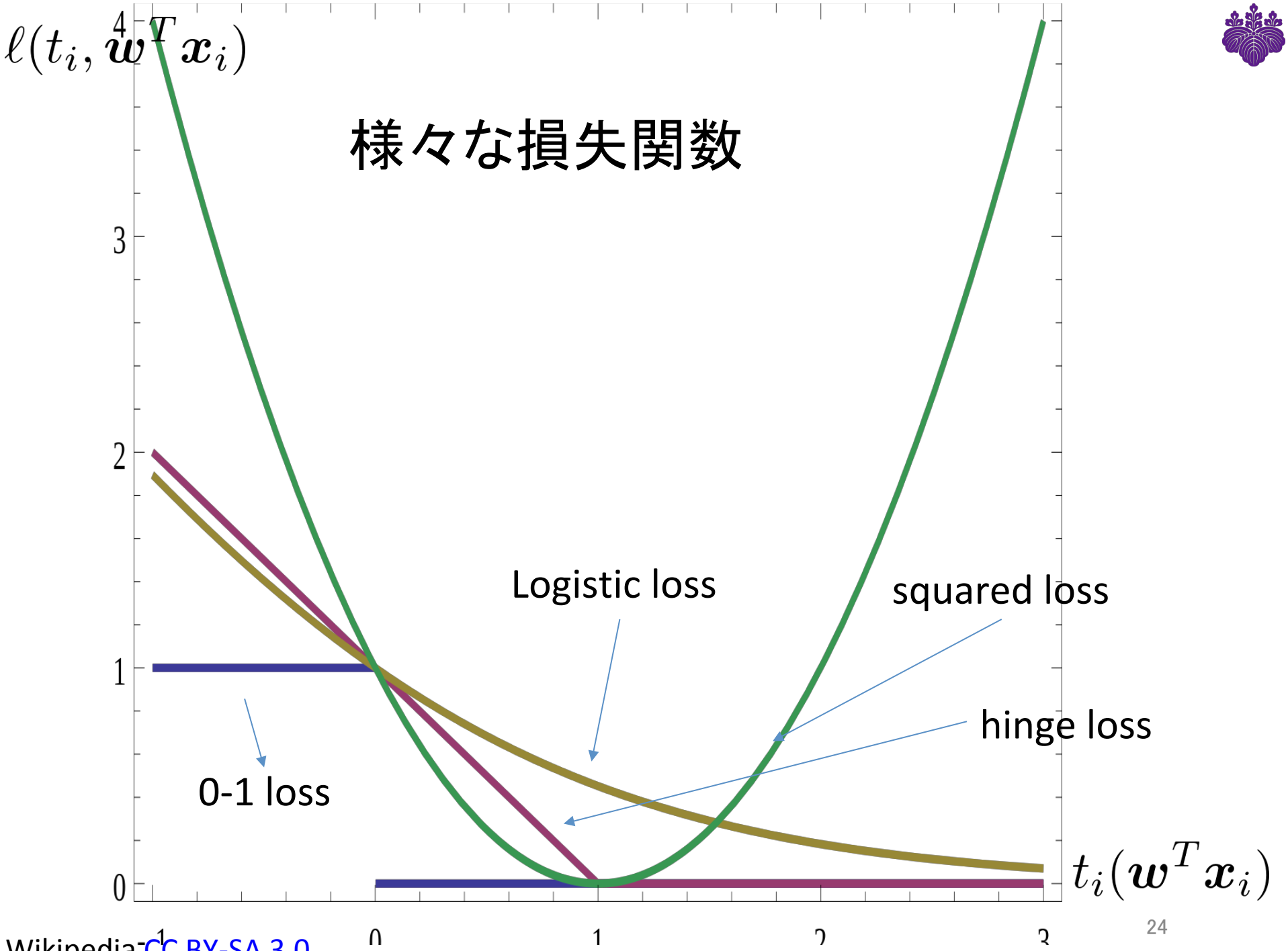
経験損失最小化
Empirical risk minimization (ERM)

- 目的関数

$$L(\mathbf{w}) = \sum_{n=1}^{N} \ell(y, f(\mathbf{x})) + \lambda \Omega(\mathbf{w})$$

回帰→二乗誤差
分類→ヒンジ損失(SVM)
ロジスティック損失
(ロジスティック回帰)

L2正則化→複雑さを抑える
L1正則化→ スパース性導入



Wikipedia CC BY-SA 3.0

様々なERM

- リッジ回帰 $$E(\boldsymbol{w}) = \sum_{i=1}^{N}(t_i - \boldsymbol{w}^T\boldsymbol{x})^2 + \lambda \boldsymbol{w}^T\boldsymbol{w}$$
- Lasso $$E(\boldsymbol{w}) = \sum_{i=1}^{N}(t_i - \boldsymbol{w}^T\boldsymbol{x})^2 + \lambda \sum_{i=1}^{D}|w_i|$$
- SVM $$\min_{\boldsymbol{w}} \sum_{i=1}^{N} \max\{0, 1 - t_i(\boldsymbol{w}^T\boldsymbol{x_i})\}$$
- L2正則化ロジスティック回帰 $$\sum_{i} \log(1 + \exp(-t_i \boldsymbol{w}^T(\boldsymbol{x}_i))) + \lambda \boldsymbol{w}^T\boldsymbol{w}$$
- スパースロジスティック回帰 $$\sum_{i} \log(1 + \exp(-t_i \boldsymbol{w}^T(\boldsymbol{x}_i))) + \lambda \sum_{d}|w_d|$$

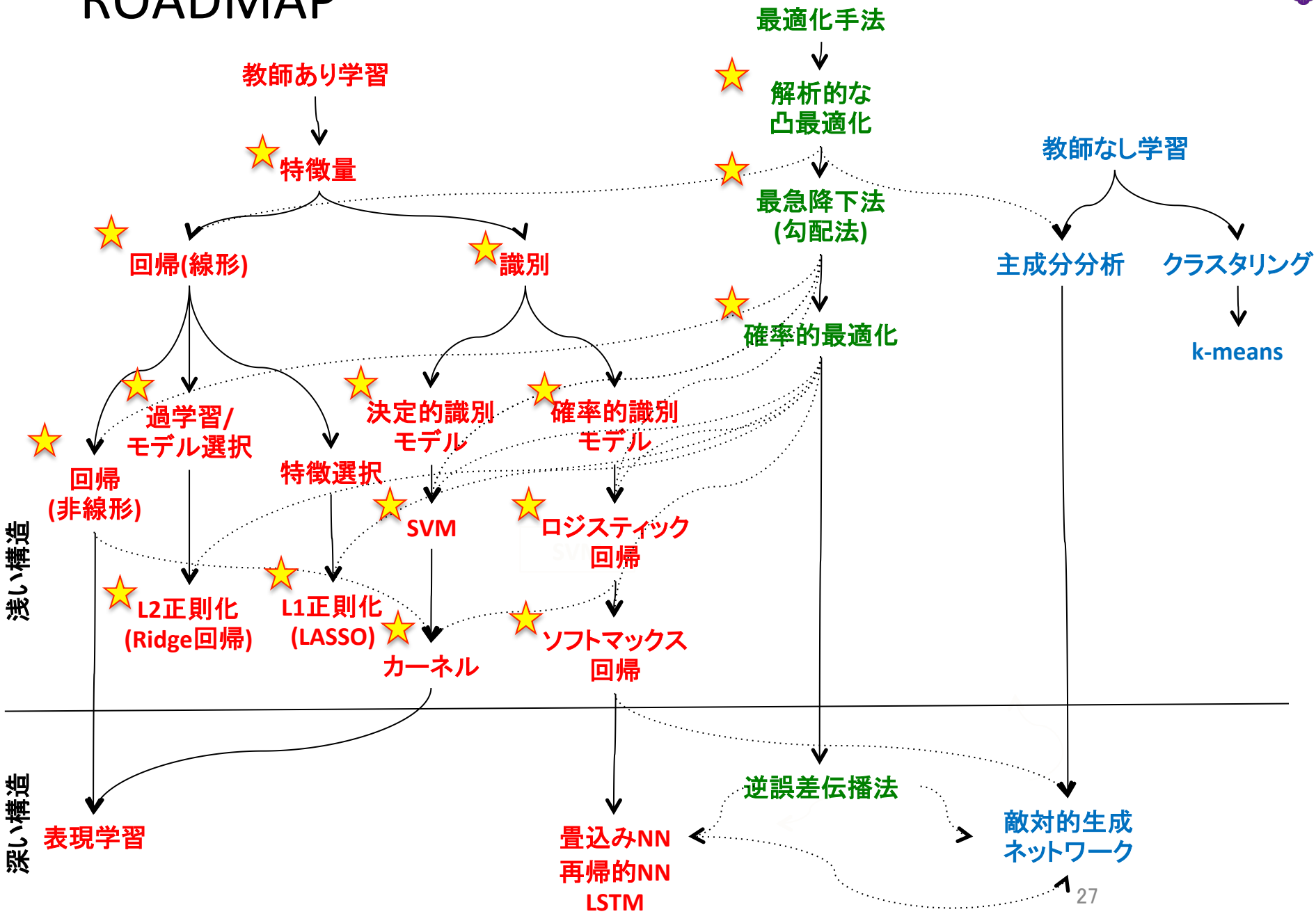
まとめ

[目的関数]

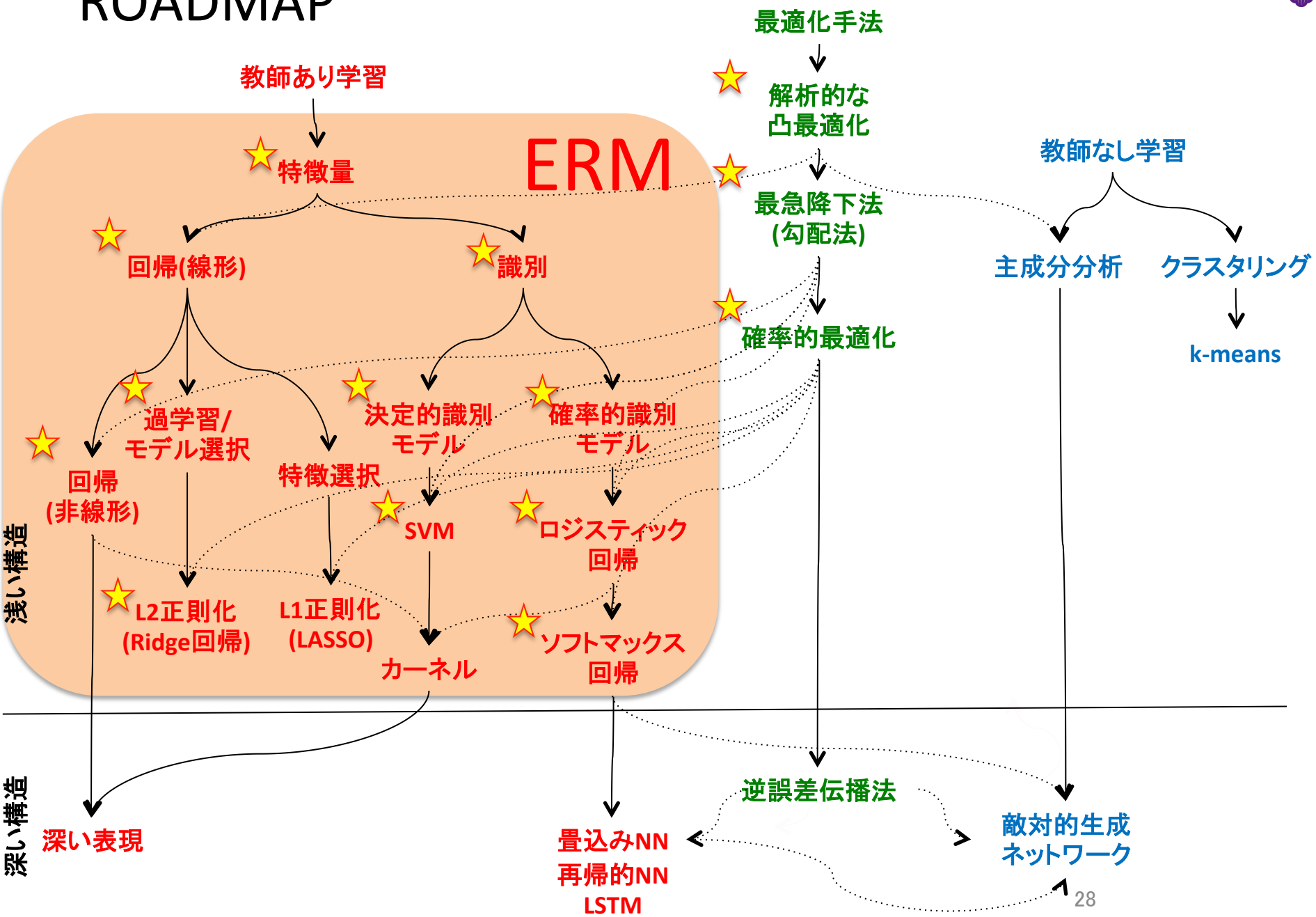
損失/正則化	正則化なし	L2正則化 （複雑さのコントロール）	L1正則化 (特徴選択)	そのほか
回帰 (連続値予測) 二乗損失	回帰	リッジ回帰	ラッソ	エラスティックネット L1+L2
決定的識別 ヒンジ損失		SVM		L2-SVM (ヒンジ損失の二乗を損失関数とする)
確率的識別 ロジスティック損失	ロジスティック回帰	L2正則化ロジスティック回帰	スパースロジスティック回帰	

[特徴]　生データ→非線形特徴量→カーネル→深層表現

ROADMAP


最適化手法
教師あり学習
解析的な
凸最適化
特徴量
教師なし学習
最急降下法
(勾配法)
回帰(線形)
識別
主成分分析
クラスタリング
確率的最適化
k-means
過学習/
モデル選択
決定的識別
モデル
確率的識別
モデル
回帰
(非線形)
特徴選択
SVM
ロジスティック
回帰
浅い構造
L2正則化
(Ridge回帰)
L1正則化
(LASSO)
カーネル
ソフトマックス
回帰
逆誤差伝播法
表現学習
畳込みNN
再帰的NN
LSTM
敵対的生成
ネットワーク
深い構造

ROADMAP


最適化手法
教師あり学習
解析的な
凸最適化
ERM
特徴量
教師なし学習
最急降下法
(勾配法)
回帰(線形)
識別
主成分分析
クラスタリング
確率的最適化
k-means
過学習/
モデル選択
決定的識別
モデル
確率的識別
モデル
回帰
(非線形)
特徴選択
SVM
ロジスティック
回帰
浅い構造
L2正則化
(Ridge回帰)
L1正則化
(LASSO)
カーネル
ソフトマックス
回帰
逆誤差伝播法
深い構造
深い表現
畳込みNN
再帰的NN
LSTM
敵対的生成
ネットワーク