

機械学習エンジニアコース Sprint

ー ロジスティック回帰 ー



DIVE INTO CODE



今回のモチベーション

目的はなにか

1. **統計モデルを知る**
スクラッチを通してロジスティック回帰を理解する
2. **分類問題とそうでないものの違いを知る**
分類問題についての基礎を学ぶ



このスライドは?

ここでは、ロジスティック回帰の基本的な知識を学びましょう



ロジスティック回帰とはなにか

ロジスティック回帰(Logistic regression)とは、ベルヌーイ分布 / 二項分布(1)に従う分類モデルの一種である。

ロジスティック回帰の出力は、「ある出来事が発生する**確率**」であり、この出力によって**クラス**の分類を行う。

この確率は、平均へと回帰する線形結合を「関数で変換する」(2)ことによって導かれるため、ロジスティック回帰は**分類問題のタスクに用いられる**にも関わらず、「回帰」と名付けられている。

具体的には、線形回帰の出力をあとで紹介するシグモイド関数に通して二項分布の生起確率へと変換したものがロジスティック回帰の出力である。

ゆえに、ロジスティック回帰も一方の変数を他方の変数(要因ごとに重みをかけた上で)によって説明しようとする点は線形回帰と変わりはない。



ロジスティック回帰とはなにか

(1) 二項分布(Binomial Distribution)とは、互いに独立したベルヌーイ試行(2種類のみの結果しか得られないような試行。例えば、「サイコロを投げた場合1なのか、それ以外なのか?」というのを考える場合はベルヌーイ試行だが、「サイコロを投げてどの目が出るか?」というのを考えるのはベルヌーイ試行ではない)を独立に n 回行ったときに、ある事象が何回起こるか(成功回数)を表す離散確率分布のことである。パラメータとして試行回数と成功確率を持つ。二項分布は n が十分に大きいとき、平均 np 、分散 $np(1-p)$ の正規分布に近づく(p は成功確率のこと)。試行回数が1回の二項分布は、ベルヌーイ分布に等しい。

<https://to-kei.net/distribution/binomial-distribution/#i>

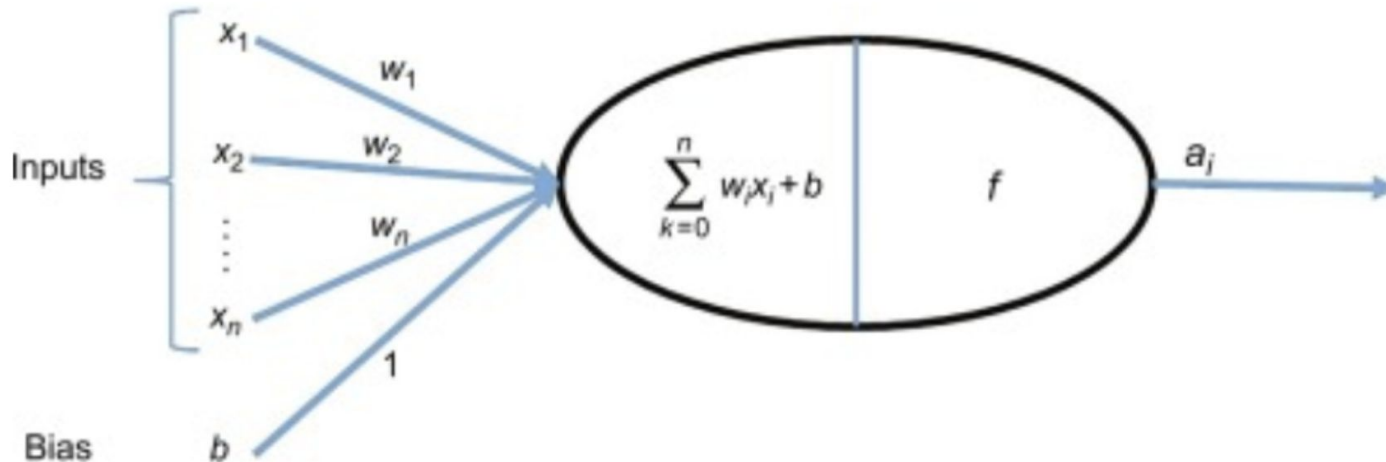
(2) シグモイド関数を用い、確率と線形結合 ($\theta_0 + \theta_1 x$) を関連づける。一般に、目的変数が正規分布に従わないとき、仮定関数は $\mu = \theta_0 + \theta_1 x$ (μ : 平均値)ではなく、 $G(\mu) = \theta_0 + \theta_1 x$ のような G (リンク関数) によって、モデルとデータを関連づけることができる。こうしたモデルを一般化線形モデル(GLM)と呼ぶ。



ロジスティック回帰とはなにか

ロジスティック回帰というテーマについて

ロジスティック回帰は、隠れ層のないニューラルネットワークと同じ構造であることから、ロジスティック回帰の理解は、深層学習への橋渡しとなる。





与えられた条件は何か

ロジスティック回帰においては以下が仮定されている。

- ①説明変数(x)は連続値 or 離散値で、目的変数(y)は離散値である。
- ②予測値(\hat{y})は、ベルヌーイ分布 / 二項分布(平均 np 、分散 $np(1-p)$)に従う。



この課題の対象者

①scikit-learnの分類モデルを用いて、学習、推定するコードが書ける方

②機械学習モデルの計算過程(仮定関数、目的関数、最急降下法)を知っている方(1)

(1) sprint3 線形回帰スクラッチを解いた方



この後の流れ

ロジスティック回帰の問題設定を知る

- ① シグモイド関数を用いて予測値を導く式(仮定関数)をたてる
- ② 同時確率を最大化する式(尤度関数)をたてる
- ③ 最小化する問題に再設定し、式(目的関数)をたてる
- ④ 目的関数の最適解を探索的に求める(最急降下法)
- ⑤ 最適解に至るとき、仮定関数の最適なパラメータが求まる

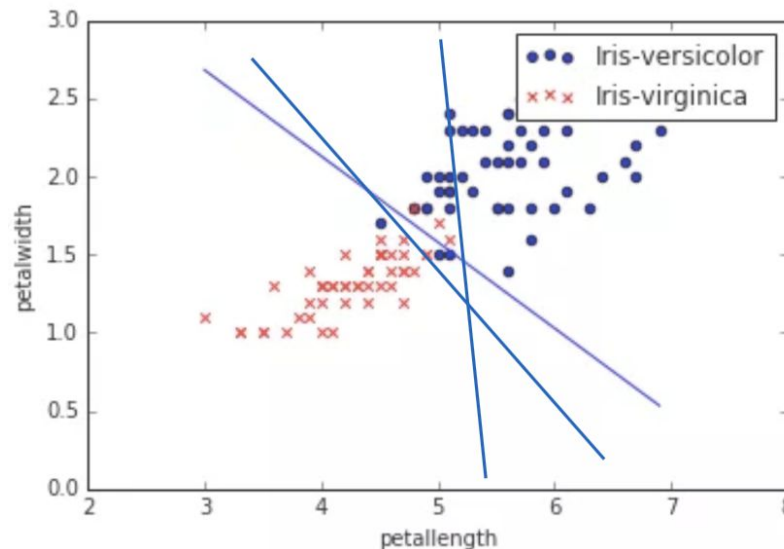


この後の流れ

Iris data

いまここにIrisデータセットがあるとしよう。データ点はあらかじめクラスごとに色分けされている。ある特徴量X1(petallength)と特徴量X2(petalwidth)を選び、二変数間の関係をプロットしてみよう。

Iris-versicolorとIris-virginicaのクラスを分類するための決定境界(境界線)が引けると嬉しいが、そもそもどのようにして決定境界を引けば良いだろうか。
ここで線形回帰の直線(線形結合)を利用してみよう。





この後の流れ

下の式にある**シグモイドという関数**を用いると、
線形結合の出力を確率に変換することができる。

sigmoid function:
$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$

しきい値(0.5)をもち、特定の分類の確率を返す関数

式中の z に線形結合 ($\theta_0 + \theta_1 x_i$) を代入すると、あるクラスに属する確率を返してくれる。つまり、 i 番目の x_i が、クラス1である確率を返す。

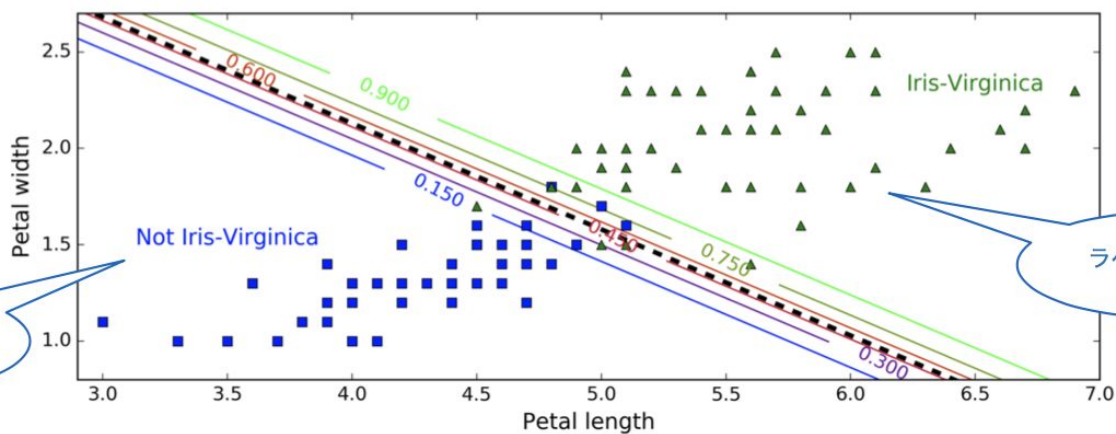
すると、確率が0.5になる地点は、ちょうど決定境界とみなすことができる。
直線を引いて分離する、というよりは、0.5をしきい値として領域の座標に確率を付与した結果、0.5地点に直線が引けている、とみなすことができそうだ。



この後の流れ

sigmoid function: $\sigma(z) = \frac{1}{1 + \exp(-z)}$

決定境界（黒点線）: $0.5 = \frac{1}{1 + \exp(-z)}$



ラベル1の領域では
決定境界から遠ざかるほど確率が
高く、ラベル0の領域では
決定境界から遠ざかるほど確率が
低くなる



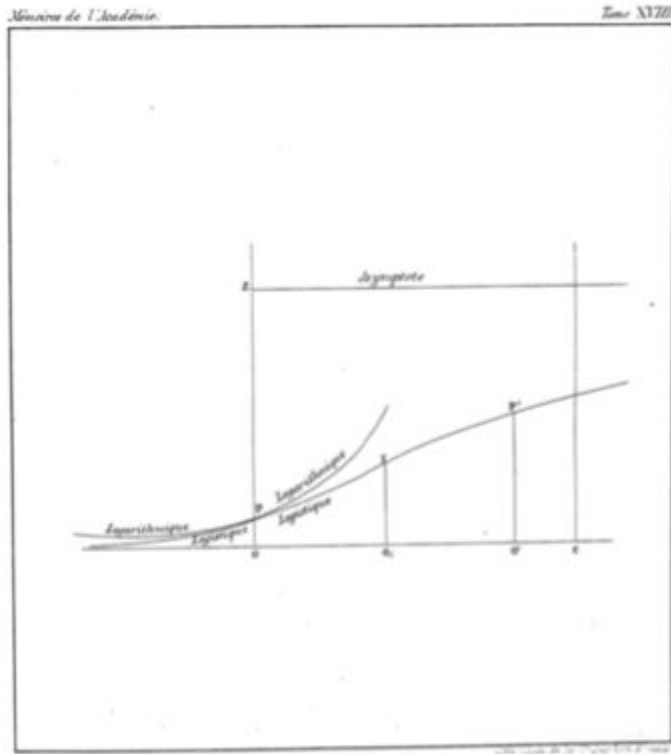
この後の流れ

シグモイド関数とは

ロジスティックシグモイド関数とも呼ばれる。もともと人口の成長をモデル化するために開発された。成長の初期段階はほぼ指数関数的に増加し(幾何学的)、次に飽和が始まると、成長は線形(算術)まで遅くなり、成熟すると成長が停止することを説明している。

※Wikipedia「Logistic function」より

https://en.wikipedia.org/wiki/Logistic_function#In_ecology:_modeling_population_growth



Mémoire sur la population par M. P. Verhulst.

Original image of a logistic curve, contrasted with a logarithmic curve



この後の流れ

指数関数とシグモイド関数

「生物の連続的繁殖モデルにおけるロジスティック方程式の解は、資源制約がない場合は指数関数的増加を示し、資源制約がある場合はシグモイド型増加を示す。」

※ note シングularityが「出来ない理由」より

ある時点まで指数関数曲線を描いても、ある点で安定したり、減少することをシグモイド関数は表す、と見做すことができそうだ。

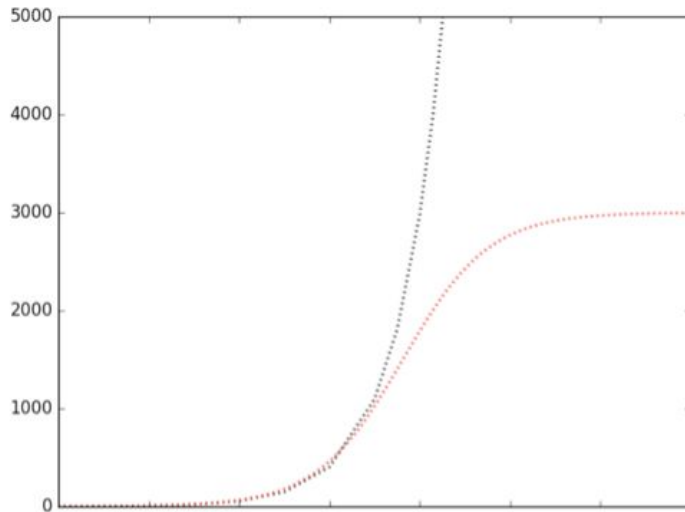


図1: 指数関数曲線 (黒) とシグモイド曲線 (赤)

<http://skeptics.hatenadiary.jp/entry/2017/06/27/232828>



この後の流れ

ロジスティック回帰の問題設定を知る

- ① シグモイド関数を用いて予測値を導く式(仮定関数)をたてる
- ② 同時確率を最大化する式(尤度関数)をたてる
- ③ 最小化する問題に再設定し、式(目的関数)をたてる
- ④ 目的関数の最適解を探索的に求める(最急降下法)
- ⑤ 最適解に至るとき、仮定関数の最適なパラメータが求まる



この後の流れ

今回の仮定関数

シグモイド関数に線形結合を入力した出力:

z :線形結合($\theta_0 + \theta_1 x$)

$$\hat{y} = \sigma(z) = \frac{1}{1 + \exp(-z)}$$

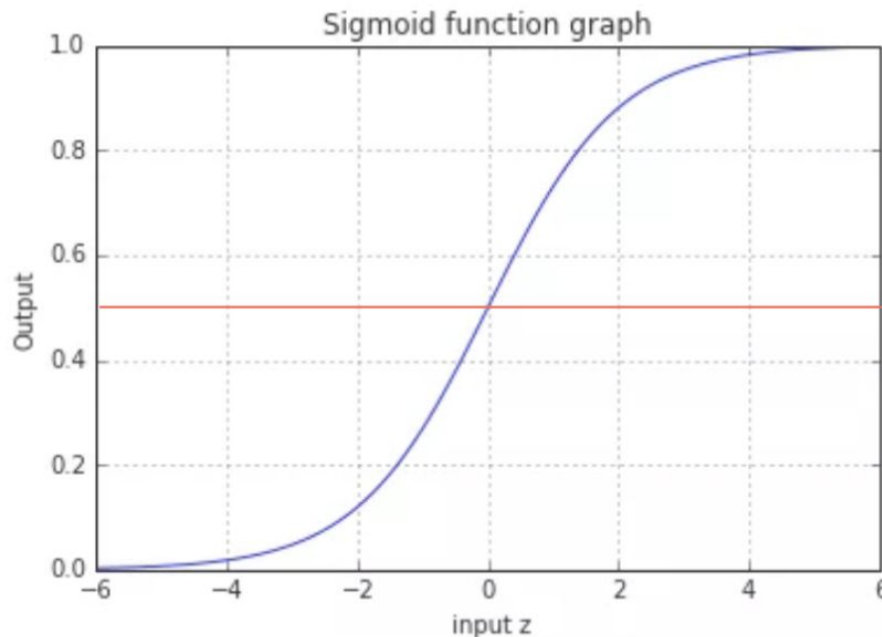


この後の流れ

今回の仮定関数

$(\theta_0 + \theta_1 x)$ がどんな値をとろうとも、その出力はsigmoidの出力範囲 $0 \leq y \leq 1$ におさまる。

この性質のおかげで、2値分類の正解ラベル = $\{0, 1\}$ を予測することができる。
なぜなら、一方のクラスを $y = 1$ 、他方のクラスを $y = 0$ で表すとき、出力される確率を「クラス1が起きる確率」と見做すことができるからである。



$$y = 1 \left(\frac{1}{1 + \exp(-z)} > 0.5 \right)$$

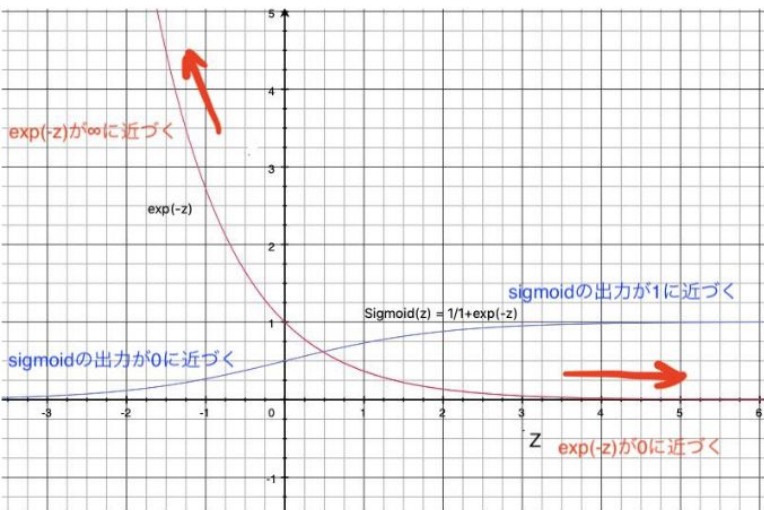
$$y = 0 \left(\frac{1}{1 + \exp(-z)} < 0.5 \right)$$



この後の流れ

シグモイド関数の入力と出力の値の関係

入力される線形結合の値が増加する場合と、減少する場合で、シグモイドからの出力結果がどう変化するか確認しよう。



入力(z): $(\theta_0 + \theta_1 x) > 0$ のとき

$\exp(-(\theta_0 + \theta_1 x))$ は 0 に近づく

出力: $\text{sigmoid}(\theta_0 + \theta_1 x) \rightarrow 1$ (1に近づく)

入力(z): $(\theta_0 + \theta_1 x) < 0$ のとき

$\exp(-(\theta_0 + \theta_1 x))$ は ∞ に近づく

出力: $\text{sigmoid}(\theta_0 + \theta_1 x) \rightarrow 0$ (0に近づく)



この後の流れ

シグモイド関数と線形結合の関係

予測値 \hat{y} と線形結合を関連づける関数をロジット(logit)と呼ぶ。
以下のように関係を表すことができる。

$$\begin{aligned}\hat{y} &= \frac{1}{1 + \exp(-\theta^T x)} \\ \text{logit}(\hat{y}) &= \theta^T x \\ &= \log \frac{\hat{y}}{1 - \hat{y}}\end{aligned}$$

$$\begin{aligned}\log \frac{\hat{y}}{1 - \hat{y}} &= \theta^T x \\ \Leftrightarrow \frac{\hat{y}}{1 - \hat{y}} &= \exp(\theta^T x) \\ \Leftrightarrow \frac{1 - \hat{y}}{\hat{y}} &= \frac{1}{\exp(\theta^T x)} \\ \Leftrightarrow \frac{1}{\hat{y}} - 1 &= \frac{1}{\exp(\theta^T x)} \\ \Leftrightarrow \frac{1}{\hat{y}} - 1 &= \frac{1}{\exp(\theta^T x)} \\ \Leftrightarrow \frac{1}{\hat{y}} &= \frac{1 + \exp(\theta^T x)}{\exp(\theta^T x)} \\ \Leftrightarrow \hat{y} &= \frac{\exp(\theta^T x)}{1 + \exp(\theta^T x)} \\ \Leftrightarrow \hat{y} &= \frac{1}{1 + \exp(-\theta^T x)}\end{aligned}$$



この後の流れ

ロジット関数と線形結合の関係式を変形していくと、シグモイド関数と線形結合の一見して複雑な関係を導くことができる。

ロジット関数は対数オッズとも呼ばれ、この対数オッズは、線形結合を評価している。

$\frac{y}{1-y}$: できごとが生じる確率 / できごとが生じない確率

$y = 0$ のとき、 $\frac{y}{1-y}$ は 0 となり、対数オッズはマイナス ∞ になる

$y = 0.5$ のとき、対数オッズは 0 になる



この後の流れ

ロジスティック回帰の問題設定を知る

- ① シグモイド関数を用いて予測値を導く式(仮定関数)をたてる
- ② 同時確率を最大化する式(尤度関数)をたてる
- ③ 最小化する問題に再設定し、式(目的関数)をたてる
- ④ 目的関数の最適解を探索的に求める(最急降下法)
- ⑤ 最適解に至るとき、仮定関数の最適なパラメータが求まる



この後の流れ

いま $y_i = 1$ である確率を p_i とし、 $y_i = 0$ である確率を $1 - p_i$ とすると、以下のように尤度(1)の式が書ける。

式中の p_i を、 $\sigma(\theta^T \mathbf{x}_i)$ と入力したと置き換えると以下の式になる。

得られ $l_i = \sigma(\theta^T \mathbf{x}_i)^{y_i} (1 - \sigma(\theta^T \mathbf{x}_i))^{1-y_i}$ 独立した確率とみなし、**同時確率**を求めたい。
これは積で表すことができる。

$$L = l_1 \times l_2 \times \dots \times l_i = \prod_{i=1}^N l_i$$

同時確率を表す (ゆうど)

(1) 尤度は確率(密度)を標本個数分掛けたものだが、それが意味するところは「事象の確率」のように見えるがそうではなく、「仮定したパラメータによる確率分布が観測データ(y)にどれほどよく当てはまるか」を示す。

尤度関数は、 $0 \leq l_i \leq 1$ の積なので、極めて小さい値になりうる。アンダーフローを避けるためと、計算を簡単にする理由から、尤度関数は対数変換されることが多い。よって、先ほどの式は

$$L = \prod_{i=1}^N \sigma(\theta^T \mathbf{x}_i)^{y_i} (1 - \sigma(\theta^T \mathbf{x}_i))^{1-y_i}$$

$$\log L = \sum_{i=1}^N \log \sigma(\theta^T \mathbf{x}_i)^{y_i} (1 - \sigma(\theta^T \mathbf{x}_i))^{1-y_i}$$

のように書き換えることができる。積が和になったことで、値が小さくなることを回避できた。対数変換した尤度関数を対数尤度関数と呼ぶ。また(θ というパラメータを仮定し) \mathbf{x}_i の値が観測できたとき、クラスが y_i である確率を求めることは、以下のような条件付き確率を求めることに等しい。

(C : クラス)

$$P(C = y_i | \mathbf{x}_i; \theta)$$



この後の流れ

「尤度を**最大化**する問題(最尤推定法)」
→「損失を**最小化**する問題」

$$\begin{aligned} L(\theta) &= - \prod_{i=1}^N P(C = y_i | \mathbf{x}_i; \theta) \\ \log L(\theta) &= - \sum_{i=1}^N \log P(C = y_i | \mathbf{x}_i; \theta) \\ &= - \sum_{i=1}^N \log \sigma(\theta^T \mathbf{x}_i)^{y_i} (1 - \sigma(\theta^T \mathbf{x}_i))^{1-y_i} \\ &= - \sum_{i=1}^N y_i \log \sigma(\theta^T \mathbf{x}_i) + (1 - y_i) \log(1 - \sigma(\theta^T \mathbf{x}_i)) \end{aligned}$$

ここまで登場した条件付き確率と対数尤度関数を左式に示す。先程までと違うのは、先頭に**マイナス**が付いている点である。条件付き確率や対数尤度関数はその**最大化**を目的にしているが、ここではマイナスをつけて逆転させ、**最小化問題**とする。式変形して導かれた最後の式が、今回の**目的関数**である。

※式変形で用いられている積の対数、累乗の対数の公式は以下を参照

<https://sci-pursuit.com/math/logarithm-formulae-and-calculation.html>



この後の流れ

ロジスティック回帰の問題設定を知る

- ① シグモイド関数を用いて予測値を導く式(仮定関数)をたてる
- ② 同時確率を最大化する式(尤度関数)をたてる
- ③ 最小化する問題に再設定し、式(目的関数)をたてる
- ④ 目的関数の最適解を探索的に求める(最急降下法)
- ⑤ 最適解に至るとき、仮定関数の最適なパラメータが求まる



この後の流れ

今回の目的関数

クロスエントロピー損失関数 + 正則化項

先ほど、マイナス付きの対数尤度関数の式変形から導かれた目的関数をサンプル数で割ったものを、今回の目的関数とする。この式を**クロスエントロピー損失関数**と呼ぶ。
機械学習では、k個のうちどのクラスに該当するか判断する場合に、目的関数としてクロスエントロピー損失関数を用いることが多い。

$$h_{\theta}(x) = \theta^T x$$

errorをサンプル数で割る

正則化項

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N [-y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] + \frac{\lambda}{2N} \sum_{j=1}^m \theta_j^2$$



この後の流れ

この目的関数で何を実現する？

真の確率分布 $y^{(i)}$ と、予測した確率分布 $\log(h_{\theta}(x^{(i)}))$ (情報量とも言う) の差が小さくなるように、つまり、予測した確率分布の形状を $y^{(i)}$ へ近づくように学習することができる。

$$h_{\theta}(x) = \theta^T x$$

① 0を返す if $y = 0$ ② 0を返す if $y = 1$

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N [-y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] + \frac{\lambda}{2N} \sum_{j=1}^m \theta_j^2$$



この後の流れ

クロスエントロピー損失関数の式の意味を確認する。

前半部分(①)のデータ $x^{(i)}$ のクラス $y^{(i)}$ が0の場合、①全体の計算が0となり、後半部分(②)のみが残る。逆に、②のデータ $x^{(i)}$ のクラス $y^{(i)}$ が1の場合、②全体の計算が0となり、①が残る。①か②のいずれかの値分だけマイナスを加算していき、全体としてより小さくなるような θ を求めることを目指す。

$$h_{\theta}(x) = \theta^T x$$


① 0を返す if $y = 0$ ② 0を返す if $y = 1$

$$J(\theta) = \frac{1}{N} \sum_{i=1}^N [-y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] + \frac{\lambda}{2N} \sum_{j=1}^m \theta_j^2$$



この後の流れ

正則化項とはなにか

$$\text{クロスエントロピー損失関数} + \text{正則化項} \left(+ \frac{\lambda}{2N} \sum_{j=1}^m \theta_j^2 \right)$$


正則化はデータセットへの過剰適合を回避し、汎化誤差を減らすために追加情報を導入するテクニックである。

機械学習の分野では、「重み減衰(weight decay)」と呼ばれる。モデルの当てはめに必要ないと判断される重みが徐々に減衰して0に近づいていくためである。

正則化項はまた、「ノルムを利用したペナルティ」と捉えることができるが、用途に応じてノルムの種類を使い分ける⁽¹⁾。機械学習において最も一般的なものは、L1ノルム(マンハッタン距離)を利用したL1正則化($p=1$)と、L2ノルム(ユークリッド距離)を利用したL2正則化($p=2$)である。

L1正則化の特徴は、正則化パラメータの λ より小さい重みを0とし、それ以外は λ 分だけ0に近づけることで、特徴量選択を行うことである。L2正則化の方は、重みを相対的な重要度に応じて0に近づけ、大きな重みほど制御し、全体的に平す仕組みである。いずれも正則化パラメータは正の定数で、これを大きくするほど正則化の効果が高まるが、大きすぎると適合不足になる。

(1) <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.642.3159&rep=rep1&type=pdf>



この後の流れ

$p = 1$ のときは、重みの
絶対値の和なので、ひし形
状の領域が目的関数に対する
制約となる

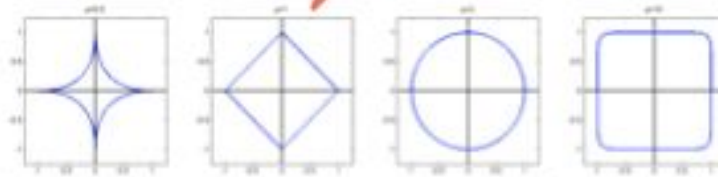


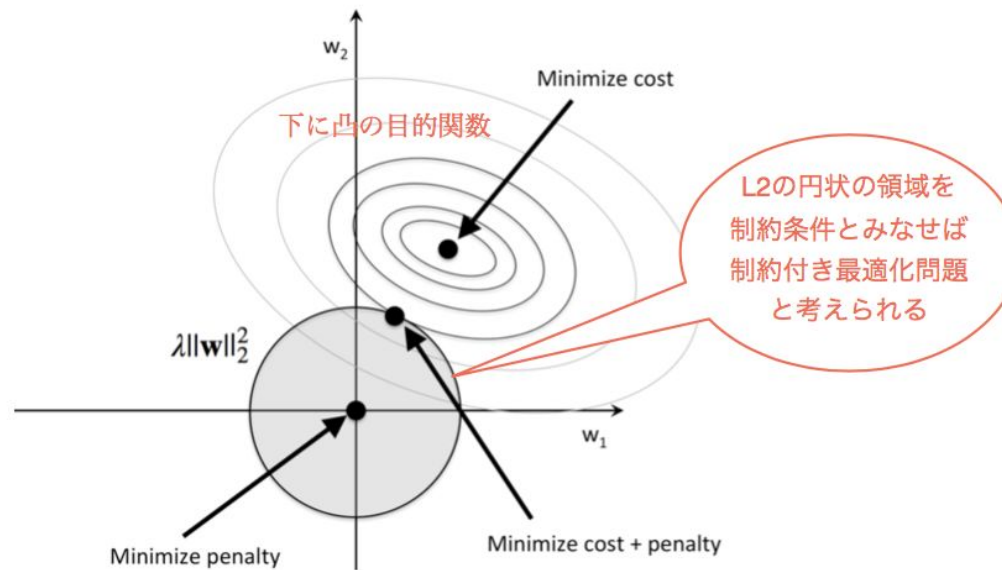
Figure 1: Unit circles for several Minkowski- p -norms $\|\mathbf{x}\|_p$: from left to right $p = 0.5$, $p = 1$ (Manhattan), $p = 2$ (Euclidean), $p = 10$.



この後の流れ

正則化項をつける意味

なぜロジスティック回帰の目的関数に**正則化項**を用いるのか？



<http://robonchu.hatenablog.com/entry/2017/10/15/112724>



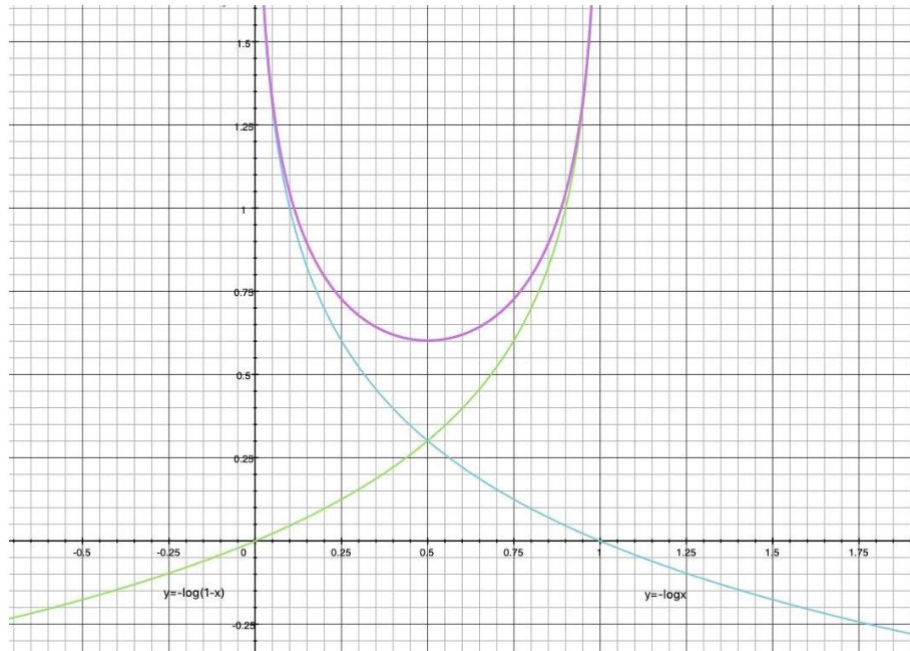
この後の流れ

ロジスティック回帰の問題設定を知る

- ① シグモイド関数を用いて予測値を導く式(仮定関数)をたてる
- ② 同時確率を最大化する式(尤度関数)をたてる
- ③ 最小化する問題に再設定し、式(目的関数)をたてる
- ④ 目的関数の最適解を探索的に求める(最急降下法)
- ⑤ 最適解に至るとき、仮定関数の最適なパラメータが求まる



この後の流れ



最急降下法で最適解が見つかるか

今回の最適解を探索すべき目的関数（クロスエントロピー損失関数）は二つのグラフから構成されている。

$-y^{(i)}\log(h_{\theta}(x^{(i)}))$ (青いグラフ)と

$-(1 - y^{(i)})\log(1 - h_{\theta}(x^{(i)}))$ (緑のグラフ)

の2つの凸関数を足し合わせると下に凸の関数(ピンクのグラフ)になる（左図のイメージ）。

この目的関数ならば、最急降下法で探索的に最適解を見つけることができ、ひいては仮定関数のパラメータの更新も行うことができる。



この後の流れ

ロジスティック回帰の問題設定を知った

- ① シグモイド関数を用いて予測値を導く式(仮定関数)をたてる
- ② 同時確率を最大化する式(尤度関数)をたてる
- ③ 最小化する問題に再設定し、式(目的関数)をたてる
- ④ 目的関数の最適解を探索的に求める(最急降下法)
- ⑤ 最適解に至るとき、仮定関数の最適なパラメータが求まる

ロジスティック回帰 完