MachineLearning and Statistics

ari

2018/1202

Contents

1 Introduction

世の中最新手法も様々出ていますが、基礎知識が全くない状況での最新手法を調べるのは理解の上で非効率だと思っているので、機械学習、統計の基本を一人で振り返るアドベントカレンダーをする。これだけできれば機械学習の数理の基本はできているという状態を目指す。

1.1 Logistic Regression

ロジスティック回帰はよく使われる機械学習の手法の 1 つで以下の特徴がある。

- 分類
- 確率 (確信度) から結果を分類

回帰と名前がついていますが、確率を元に分類することに注意せよ。もちろん確率的な分布が知りたい時に使って問題はない。ロジスティック回帰の"気持ち"について説明する. ロジスティック回帰も基本的には二値分類を考えており, 気持ちとしては y が 1 である確率を予測している.

これは予測に使う関数が 0 から 1 までの値をを取り、なおかつ, 二値 $\{0, 1\}$ で 1 が正解である確率かのように判断するためである.(そしてこの確率が一定の値を超えた時に 1, そうでない時 0 と判定する.

Remark. ロジスティック回帰は数理モデル等で出てくるロジスティック方程式と関係がある。何が一番最初なのかもはやわからないので、関係があることだけ注意しておく。

具体的な問題設定について説明する。

- データ $D := \{(x_i, y_i)\}, x_i \in \mathbb{R}^n, y_i \in \{0, 1\}$
- パラメータ $w \in \mathbb{R}^n$

- 関数の取りうる範囲 f
- 損失関数 $L(D)$
- パラメータ変更アルゴリズム ニュートン法

1.1.1 Parameter

今回考えるパラメーター全体, つまり、考える関数全体の集合を以下で定める。

$$\{\frac{1}{1+e^{-ax+b}} \mid a \in \mathbb{R}^n, b \in \mathbb{R}\}$$

この関数は $ax \to \infty$ の時 1,$ax \to -\infty$ の時 0 となる.

1.1.2 損失関数

推論では 1 である確率 $P(y = 1 \mid x \in \mathbb{R}^N)$ を $\frac{1}{1+e^{wx_i}}$ と考えるので、この推論結果が担保されるように損失関数を設定する.

基本的には全データを使う.(実際は計算速度が問題になる場合はデータをランダムに選択しているだろう) その場合、各事象が独立だとすると今回の結果になる確率は $\prod_i P(y_i = 1 \mid x_i)^{y_i}(P(y_i = 0 \mid x_i))^{1-y_i}$ であり, これが最大になることで損失関数を表現したい。

とはいえ、積は実際計算や改善アルゴリズムがあまり精度高くできないので、積を和の形にする.

典型的には log を使うこで積を和に変換できるので、以下のようにロス関数を定める

$$L(D) = \log \prod_i P(y_i = 1 \mid x_i)^{y_i}(P(y_i = 0 \mid x_i))^{1-y_i} = \sum (y_i \log \frac{1}{1+e^{-ax+b}} + (1-y_i)\log(1-\frac{1}{1+e^{-ax+b}}))$$

1.1.3 parameter 変更アルゴリズム

汎用的に考えると勾配降下法やニュートン法等で問題を解く。ただし, データ数が多すぎる場合は確率的勾配降下法を使う. ニュートン法自体の解説は別の章を参照せよ.

Remark. 多値の場合は *1,0* とわけるのではなく第 n 成分が *1* となるように分ける. その場合は *sigmoid* 関数を使う. つまり,

$$P(y_i = 1 \mid X = x) = \frac{e^{a_i x+b}}{\sum_j e^{a_j x+b}}$$

となるように推論時の関数を設定すればよいこれから損失関数も求まる