やさしく学ぶ機械学習を理解するための数学のきほん

2018 年 6 月 27 日

1　イントロダクション

1.1　機械学習が得意とすること

- 回帰 (Regression)
- 分類 (Classification)
- クラスタリング (Clustering)

1.1.1　回帰

連続するデータ (例えば時系列データなど) を扱うときに使われる

株価のような過去のデータが入手できるもので、「明日の株価はどうなりそうか」などを予測するのに回帰を使うことがある。

1.1.2　分類

例えばスパムメールの判定などに使われる。

スパムかどうかなどの分類先が二つしかないものは二値分類と呼ばれ、３つ以上の場合は多値分類と呼ばれる。数字の識別は多値分類に相当する。

1.1.3　クラスタリング

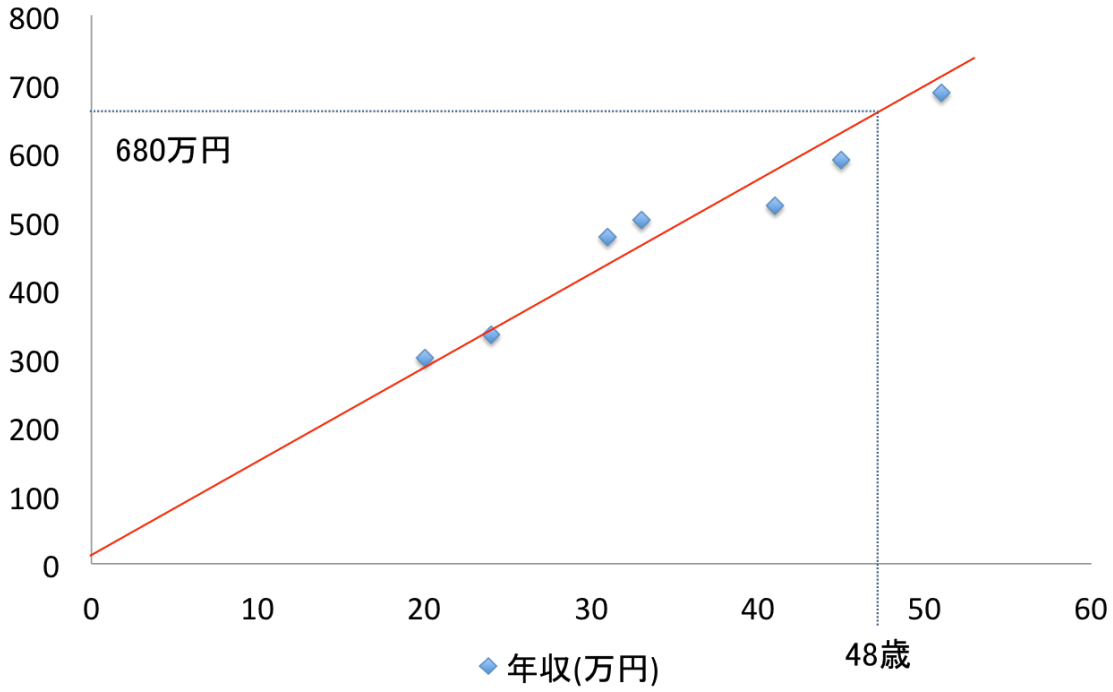
例えば生徒が 100 人いる学校の学力テストがあったとする。そのテストの点数によって生徒 100 人を幾つかのグループに分けるような問題のこと。

分類に似ているが、違うのはデータにラベルが付いていないということ。スパムメール判定で、メールの内容と一緒に、そのメールがスパムかどうかというラベルが付与されていた。でもテストの点数のデータに分類に関するラベルは付いていない。

ラベルが付いているデータを学習することを教師あり学習、ラベルが付いていないデータを使って学習することを教師なし学習という。回帰と分類は教師あり学習、クラスタリングは教師なし学習と言える。

2　回帰

以下の図のように年齢と年収のデータがあったとする (ドット)。このデータに近似するように一次関数の線を引くことができる (赤線)。



ここで赤線の関数を以下の式と定義する。

$$f(\theta) = \theta_0 + \theta_1 x$$

θ は未知数 (パラメータ) の事。統計学の世界では未知数や推定値を θ で表すことが多い。

2.1　最小二乗法

θ の値を実データとの誤差が採用になるように求める必要がある。誤差が 0 になるのが理想だが、すべてのデータに対して誤差を 0 にするのは不可能なので、すべてのデータの誤差の合計がなるべく小さくなるようにする。

実データが n 個あるとして、実データごとの誤差の総和は以下のように表すことができる。

$$E(\theta) = \frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - f_\theta(x^{(i)}))^2$$

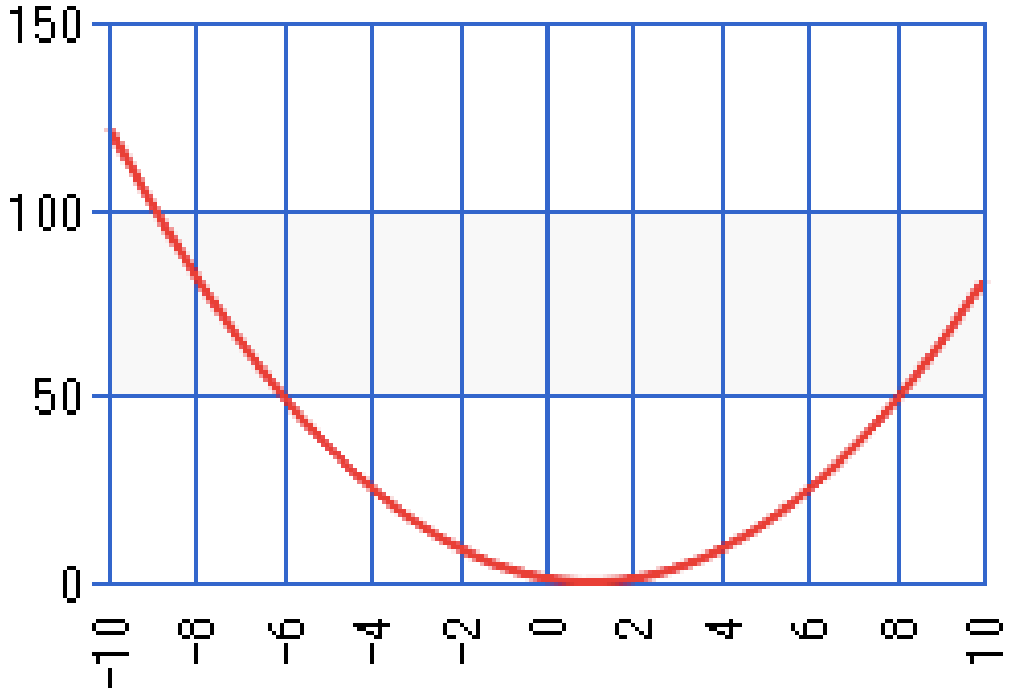
$y^{(i)}$ などは i 乗という意味ではなく、i 番目の実データという意味。

各データごとの誤差を 2 乗してそれを全部足し、$\frac{1}{2}$ してあげることで、E(θ) の値が一番小さくなるような θ を見つけることが目的。こういうものを**最適化問題という。**

絶対値でなく２乗することも、$\frac{1}{2}$ をかけるのも、あとあと微分する時に計算を楽にするため。
こういうアプローチのことを最小二乗法という

2.2　最急降下法

θ の値を適当に変えながら $\mathbf{E}(\theta)$ を計算し行くのは面倒。微分を使って求めていく。
微分について簡単に、$g(x) = (x-1)^2$ を例に見ていく。



展開すると $(x-1)^2 = x^2 - 2x + 1$ であり、微分すると $\frac{d}{dx}g(x) = 2x - 2$。どう関数の値を小さくなるように $\mathbf{x}$ の値を更新する。これを最急降下法や勾配降下法と呼ぶ。以下を更新式とする。

$$x := \eta \frac{d}{dx} g(x)$$

$A := B$ は「$\mathbf{A}$ を $\mathbf{B}$ によって定義する」という意味。η は「学習率」と呼ばれる正の定数。学習率の大小によって、最初うちにたどり着くまでの更新回数が変わってくる。収束の速さが変わるともいう。

目的関数 $\mathbf{E}(\theta)$ に話を戻すと、$\mathbf{E}(\theta)$ は $\mathbf{g(x)}$ と同様に下に凸の形をしているから同じ議論を当てはめることができる。ただ、この目的関数は θ_0 と θ_1 の二つの変数を含んでいるから、普通の微分ではなくて偏微分になっている。更新式は以下のようになる

$$\theta_0 := \theta_0 - \eta \frac{\partial E}{\partial \theta_0} \qquad \theta_1 := \theta_1 - \eta \frac{\partial E}{\partial \theta_1}$$

$\mathbf{E}(\theta)$ を偏微分してみる。$\mathbf{E}(\theta)$ の中に $f_\theta(x)$ が出てきて $f_\theta(x)$ の中に θ_0 が出てくるので、それぞれ以下のように考える。

$$u = E(\theta) \qquad v = f_\theta(x)$$

すると段階的に微分することができる。

$$\frac{\partial u}{\partial \theta_0} = \frac{\partial u}{\partial v} \cdot \frac{\partial v}{\partial \theta_0}$$

まずは **u** を **v** で微分する。

$$\begin{aligned}
\frac{\partial u}{\partial v} &= \frac{\partial}{\partial v}(\frac{1}{2}\sum_{i=1}^{n}(y^{(i)} - v)^2) \\
&= \frac{1}{2}\sum_{i=1}^{n}(\frac{\partial}{\partial v}(y^i - v)^2) \\
&= \frac{1}{2}\sum_{i=1}^{n}(y^{(i)^2} - 2y^{(i)}v + v^2)) \\
&= \frac{1}{2}\sum_{i=1}^{n}(-2y^{(i)} + 2v) \\
&= \sum_{i=1}^{n}(v - y^{(i)})
\end{aligned}$$

次に **v** を θ_0 で微分する。

$$\frac{\partial v}{\partial \theta_0} = \frac{\partial}{\partial \theta_0}(\theta_0 + \theta_1 x) = 1$$

以上より、$\frac{\partial u}{\partial \theta_0} = \frac{\partial u}{\partial v} \cdot \frac{\partial v}{\partial \theta_0}$ が求まる。

$$\begin{aligned}
\frac{\partial u}{\partial \theta_0} &= \frac{\partial u}{\partial v} \cdot \frac{\partial v}{\partial \theta_0} \\
&= \sum_{i=1}^{n}(v - y^{(i)}) \cdot 1 \\
&= \sum_{i=1}^{n}(f_\theta(x^{(i)}) - y^{(i)})
\end{aligned}$$

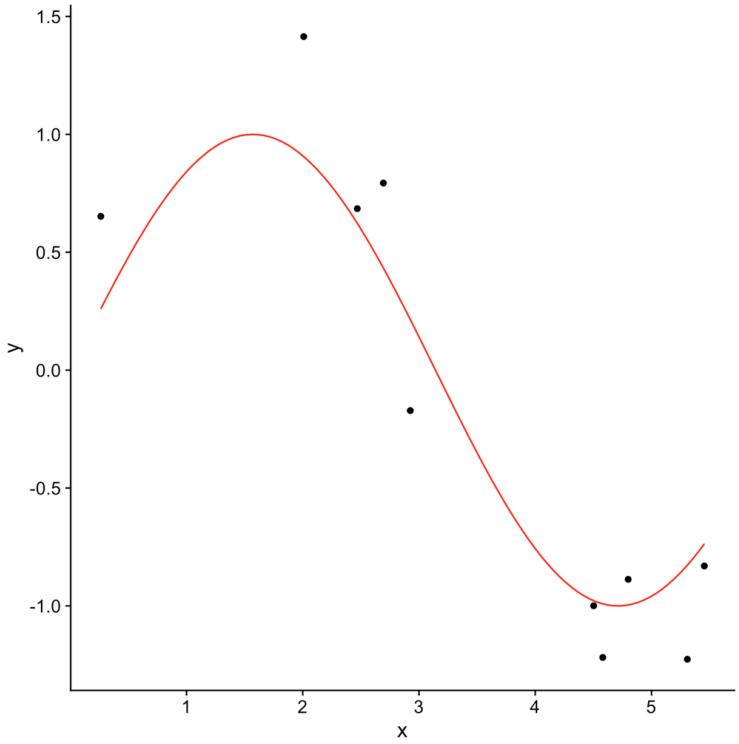
同様の処理を θ_1 にも行うと、それぞれの更新式は以下のようになる。

$$\begin{aligned}
\theta_0 * &= \theta_0 - \eta\sum_{i=1}^{n}(f_\theta(x^{(i)}) - y^{(i)}) \\
\theta_1 * &= \theta_1 - \eta\sum_{i=1}^{n}(f_\theta(x^{(i)}) - y^{(i)})x^{(i)}
\end{aligned}$$

この式にしたがって θ_0 と θ_1 を更新していけば正しい形の 1 次関数 $f_\theta(x)$ が見つかる。

3　多項式回帰

上記で挙げた一次関数の形よりも曲線のほうがフィットする場合がある。



これは関数 $f_\theta(x)$ を二次関数として定義することで実現できる。

$$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2$$

もしくはもっと大きな字数を持った式にしても適応できる。

$$f_\theta(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_n x^n$$

次数を増やせば増やすほどフィッットするようにはなるが、「過学習」と呼ばれる避けては通れない別の問題が発生する。

二次関数の場合の更新式は以下のようになる。

$$\theta_0 = \theta_0 - \eta \sum_{i=1}^{n} (f_\theta(x^{(i)}) - y^{(i)})$$
$$\theta_1 = \theta_1 - \eta \sum_{i=1}^{n} (f_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$
$$\theta_2 = \theta_1 - \eta \sum_{i=1}^{n} (f_\theta(x^{(i)}) - y^{(i)}) {x^{(i)}}^2$$

このように、パラメータが θ_3, θ_4 と増えていっても同じことをして更新式を求めることができる。多項式の次数を増やした関数を使うものは、「多項式回帰」と呼ばれる。

4　重回帰

解きたい問題の多くは、変数が２つ以上の複雑な問題のことが多い。

例えば、「広告費用の金額によってクリック数が決まる」と設定すると、単純に広告費用に比例したクリック数となるが、現実問題、クリック数は様々な要因で上下する。例えば、広告の位置であったりサイズであったり。複数の要素がクリック数に影響を及ぼす。

具体的に「広告費 $= x_1$」「広告の横幅 $= x_2$」「広告の縦幅 $= x_3$」の **3** つのパラメータで考えると、以下のように表すことができる。

$$f_\theta(x_1, x_2, x_3) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

この時のパラメータ $\theta_0, \cdots, \theta_3$ は、それぞれ偏微分してパラメータを更新すれば求めることができる。

上記式を一般化して

$$f_\theta(x_1, \cdots, x_n) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

と表すことができる。上記のように毎回 $\mathbf{n}$ 個の $\mathbf{x}$ を書いていくのは大変なので、パラメータ θ と変数 $\mathbf{x}$ をベクトルとみなす。

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}$$

ベクトル $\mathbf{x}$ に $\mathbf{1}$ を追加しているのは、この方が自然だから。θ の添字が $\mathbf{0}$ から始まっているので、それと合わせるために $x_0 = 1$ として、$\mathbf{x}$ の最初の要素に x_0 をおく方が綺麗。

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \quad \mathbf{x} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} (\mathbf{x_0 = 1})$$

θ を転置したものと $\mathbf{x}$ をかけたものを見てみると

$$\theta^{\mathbf{T}}\mathbf{x} = \theta_{\mathbf{0}}\mathbf{x_0} + \theta_{\mathbf{1}}\mathbf{x_1} + \theta_{\mathbf{2}}\mathbf{x_2} + \cdots + \theta_{\mathbf{n}}\mathbf{x_n}$$

と、ベクトル表記する前の式となる。ベクトル式だと、

$$f_\theta(\mathbf{x}) = \theta^{\mathbf{T}}\mathbf{x}$$

次に、この $f_\theta(\mathbf{x})$ を使ってパラメータの更新式を求める。$u = E(\theta), v = f_\theta(x)$ とおくのは同じ。一般化して考えるために **j** 番目の要素の θ_j で偏微分すると、

$$\frac{\partial u}{\partial \theta_j} = \frac{\partial u}{\partial v} \cdot \frac{\partial v}{\partial \theta_j}$$

$\mathbf{v}$ を θ_j で微分すると、

$$\begin{aligned}
\frac{\partial v}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j}(\theta^{\mathbf{T}}\mathbf{x}) \\
&= \frac{\partial}{\partial \theta_j}(\theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n) \\
&= x_j
\end{aligned}$$

最終的に、**j** 番目のパラメータの更新式は以下のようになる。

$$\theta_j := \theta_j - \eta \sum_{i=1}^{n} (f_\theta(\mathbf{x^{(i)}}) - \mathbf{y^{(i)}})\mathbf{x_j^{(i)}}$$

このように複数の変数を使ったものを「重回帰」という。

5　確率的勾配降下法

最急降下法には、計算に時間がかかる以外にも、局所解に捕まってしまうという欠点がある。

最急降下法で関数の最小値を見つけるにしても、まず最初にどの $\mathbf{x}$ からスタートするかを決める必要がある。多くは乱数を使ってスタート位置を決めるが、初期位置が不適当な場合、最小値でない部分に落ち着いてしまうことがある。これを「局所解に捕まる」と表現する。

最急降下法のパラメータ更新式、

$$\theta_j := \theta_j - \eta \sum_{i=1}^{n} (f_\theta(\mathbf{x^{(i)}}) - \mathbf{y^{(i)}})\mathbf{x_j^{(i)}}$$

は、すべての学習データの誤差を使っているが、確率的勾配降下法ではランダムに学習データを 1 つ選んで、それをパラメータの更新に使う。この式の $\mathbf{k}$ はランダムに選ばれたインデックスのことである。

$$\theta_j := \theta_j - \eta(f_\theta(\mathbf{x^{(k)}}) - \mathbf{y^{(k)}})\mathbf{x_j^{(k)}}$$

ランダムで選ぶデータの数は 1 つ以上でも構わない。$\mathbf{m}$ 個選んだ場合は、インデックスの集合を $\mathbf{K}$ 遠くと以下のようにパラメータを更新する。

$$\theta_j := \theta_j - \eta \sum_{k \in K}^{n} (f_\theta(\mathbf{x^{(k)}}) - \mathbf{y^{(k)}})\mathbf{x_j^{(k)}}$$

この方法を「ミニバッチ法」という。