サポートベクトルマシン入門（前編）

前編では、

- サポートベクトル分類の仕組み
- カーネル法とはなにか？

の2点を詳しく紹介できたら良いなと思っています。

1. 線形サポートベクトル分類の問題設定

1.1 問題設定

2クラス分類（binary classification）とよばれる機械学習のタスクがあります。これは、説明変数 x の値から、2種類のラベル ± 1 のどちらかが振られている目的変数 y を予測する式 $f(x)$ を得ようというタスクです。

サポートベクトルマシンには、この2クラス分類のための手法として **線形サポートベクトル分類**（linear support vector classifier）があります。線形サポートベクトル分類では、目的変数 y を予測する式として

$$f(x;w,b) = w^T x + b$$

を考え、$f(x;w,b)$ の値が正のときには $y=+1$、負のときには $y=-1$ と予測すると約束します。この関数 $f(x;w,b)$ のことを**決定関数**（decision function）といいます。また、目的変数 y の予測が $f(x;w,b)=0$ で切り替わることから、この式を**決定境界**（decison boundary）といいます。

ところで、この式を用いて目的変数を予測するには、決定関数 $f(x;w,b)$ に現れる係数 w と切片 b が決まらないといけません。これら二つの記号をまとめて**パラメータ**といいます。パラメータを求めるためには、事前に説明変数と目的変数を複数観測したデータ $(x_i, y_i), i=1,\cdots,n$ を準備し、以下の関数が最も小さくなるような w, b の値を求めます。

$$L(w,b) = C\max\{1-y_i f(x_i;w,b), 0\} + \frac{1}{2}\|w\|^2$$

C には正の実数をあらかじめ決めておきます。パラメータを求めるために準備したデータのことを**訓練データ**（trainning data）といい、最小化する関数 $L(w,b)$ を**損失関数**（loss function）といいます。

1.2 損失関数の意味を理解しよう

損失関数

$$L(w,b) = C\max\{1-y_i f(x_i;w,b), 0\} + \frac{1}{2}\|w\|^2$$

が小さくなるのがなぜ嬉しいのかを知るために、損失関数が何を意味しているのかを考察しましょう。損失関数は、2種類の関数 $\max\{1-yf(x), 0\}$ と $\|w\|^2$ を足し合わせた形になっていますが、以下では別々に考えていきます。

$\max\{1-yf(x),0\}$ の意味

この式の意味を理解するために、**マージン**（margin）という言葉を導入します。2クラス分類では、x を説明変数、$y\in\{\pm1\}$ を目的変数、$f(x)$ を判別関数としたとき、$yf(x)$ のことをマージンといいます。

問題 : 以下の問いを通して、マージンの符号と絶対値の大きさの意味を考察してください。

(1) マージンの値が正のときは目的変数とその予測が等しいこと、負のときは異なっていることを証明してみてください。

(2) 以下のデータ点について、決定関数 $f(x)=x$ の マージンを計算することで、マージンの絶対値 $|yf(x)|$ がデータ点と決定境界との離れ具合を表していることを確認してください。

i	x	y
1	-2	-1
2	-1	$+1$
3	1	-1
4	2	$+1$

つまりマージン $yf(x)$ は負の値に大きくなるほど目的変数とその予測が食い違っていると捉えることができます。

さて、$\max\{1-yf(x),0\}$ の意味を解釈する準備ができました。これはデータ点 (x,y) に対して決定関数 $f(x)$ のマージン $yf(x)$ が 1 を下回ったときは、$1-yf(x)$ だけ大きな予測の誤りを犯したと評価すると解釈できます。また、マージンが 1 を上回ったときは予測の誤りを 0 と評価します。この関数を**ヒンジ損失**（hinge loss）といいます。ヒンジ損失がなるべく小さくなるような w,b の値を求めることで、目的変数とその予測の間の食い違いを小さくできることが期待できるわけです。

$\|w\|^2$ の意味

分類問題の目的は、新たにデータ点を得たとき、その説明変数 x の値によって目的変数の値を正しく予測できるようになることです。しかし、マージン損失のみを訓練データで最小化するように w,b の値を求めてしまうと、得られた決定関数 $f(x;w,b)$ による予測は訓練データに対してはうまく予測しているにも関わらず、新しいデータに対しては予測が高くならないという現象が発生しうることが知られています。これを**過剰適合**（overfitting）と言います。

線形サポートベクトルマシンで過剰適合が発生する原因の一つには、本来は予測に関係ないはずの変数が予測に影響を及ぼすような決定関数が得られてしまうことがあるからです。

予測に関係ない説明変数の影響を弱めるには、その変数についている係数の大きさを 0 に近くできると良いでしょう。そこで線形サポートベクトルマシンでは、損失関数に $\|w\|^2$ を加えることで係数 w がとれる値の大きさに制限を加える工夫をしています。実際、損失関数 $L(w,b)$ が小さくなるためには、係数 w の大きさ $\|w\|^2$ がある程度小さくなくてはいけません。このような工夫は他の機械学習の手法でもよく現れるもので、**l^2-正則化**（l^2-regularization）とよばれています。