7.1.1 重なりのあるクラス分布

7.1 節では非線形特徴空間上で線形分離可能な条件のもとでのSVMを考えてきた。しかし現実には線形分離できない場合がある。また、クラスの条件付き確率分布に重なりがある場合、完全に分離することは過学習になりうる。

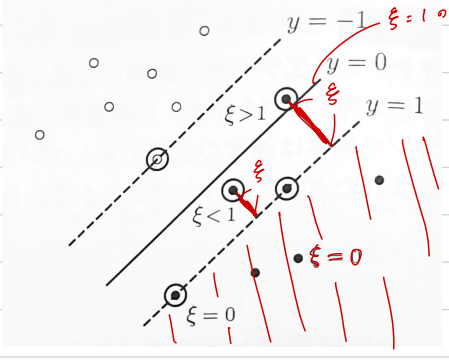
本節では、ある程度誤分類を許すことで重なりのあるクラス分布に対して汎化性能の上げることを目的とする。そのために前回導入したラグランジュの未定乗数法に改良を加える。また、ラグランジュ関数を最小化する最適化問題の解き方についても扱う。

誤分類を許すSVMのモデル化

7.1 節では、 $t_n y(x_n) \geq 1$ という条件により完全に分離される条件を表現していた。今回は、スラック変数 $\xi_n \geq 0$ を導入して以下のような条件とする。

$$t_n y(x_n) + \xi_n \geq 1$$

これにより、$t_n y(x_n)$ は1以上である必要がなくなる。ξ_n は、正しい分離面からの距離に相当し、直観的な理解としては分類の間違い度合いに相当する（下図）。



誤分類を許すだけであれば、どんな適当なモデルでも誤差 $\frac{1}{2}\|w\|^2$ を最小にできてしまう。そこで、誤差に対して ξ_n をペナルティとして与えて、次の式を誤差関数とする。

$$C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|w\|^2$$

（感想）

ξ_n をペナルティとするモチベーションは分かるが、1次式の形になる理由はイマイチ分からなかった。例えば ξ_n の2次の正則化項などではダメなのか…？

ここで $C > 0$ はペナルティとマージンの大きさを制御するパラメータであり、$C = \infty$ の場合にハードマージンSVMと等価になる。最適化問題をまとめると次のようになる。

$$
\begin{aligned}
&\min. && f(w,b,\xi) = C\sum_{n=1}^{N}\xi_n + \frac{1}{2}\|w\|^2 \\
&s.t. && \forall n,\ g(w,b,\xi_n) = t_n y(x_n) - 1 + \xi_n \geq 0 \\
& && \forall n,\ h(\xi_n) = \xi_n \geq 0
\end{aligned}
$$

これに対するラグランジュ関数は、ラグランジュ乗数 $\{a_n \geq 0\}$, $\{\mu_n \geq 0\}$ を用いて次の式になる。

$$
L(w,b,\xi,a,\mu) = \underbrace{\frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}\xi_n}_{f(w,b,\xi)} - \underbrace{\sum_{n=1}^{N}a_n\{t_n y(x_n) - 1 + \xi_n\}}_{a\ g(w,b,\xi_n)} - \underbrace{\sum_{n=1}^{N}\mu_n\xi_n}_{\mu\ h(\xi_n)}
$$

KKT 条件は次のようになる。

	対応	
$a_n \geq 0$	$a \geq 0$	(7.23)
$t_n y(x_n) - 1 + \xi_n \geq 0$	$g(w,b,\xi_n) \geq 0$	(7.24)
$a_n\{t_n y(x_n) - 1 + \xi_n\} = 0$	$a\,g(w,b,\xi_n) = 0$	(7.25)
$\mu_n \geq 0$	$\mu \geq 0$	(7.26)
$\xi_n \geq 0$	$h(\xi_n) \geq 0$	(7.27)
$\mu_n\xi_n = 0$	$\mu\,h(\xi_n) = 0$	(7.28)

あとはこれまで通り、ラグランジュ関数における w , b , ξ_n の停留点を求め、パラメータ a に関する目的関数を取得する。

$$\frac{\partial L}{\partial w} = 0 \Leftrightarrow w - \frac{\partial}{\partial w}\left\{ \sum_{n=1}^{N} a_n \left(t_n w^T \phi(x_n) + t_n b - 1 + \xi_n \right) \right\} = 0$$

$$\Leftrightarrow w = \sum_{n=1}^{N} a_n t_n \phi(x_n) \quad (7.29)$$

$$\frac{\partial L}{\partial b} = 0 \Leftrightarrow \frac{\partial}{\partial b}\left\{ \sum_{n=1}^{N} a_n \left(t_n w \phi(x_n) + t_n b - 1 + \xi_n \right) \right\} = 0$$

$$\Leftrightarrow \sum_{n=1}^{N} a_n t_n = 0 \quad (7.30)$$

$$\forall n , \frac{\partial L}{\partial \xi_n} = 0 \Leftrightarrow \forall n, \ C - a_n - \mu_n = 0$$

$$\Leftrightarrow \forall n, \ a_n = C - \mu_n \quad (7.31)$$

これらをラグランジュ関数に代入すると、双対形のラグランジュ関数が得られる。

途中式で (7.7) - (7.10) 式の結果を適用できる点に注意する。

$$L(w, b, \xi, a, \mu) = \frac{1}{2}\|w\|^2 - \sum_{n=1}^{N} a_n \{ t_n y(x_n) - 1 \} - \sum_{n=1}^{N} a_n \xi_n + C \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \mu_n \xi_n$$

(7.7) ~ (7.10) の結果を適用　　(7.31) を代入

$$\tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m) - C \sum_{n=1}^{N} a_n \xi_n + \sum_{n=1}^{N} \mu_n \xi_n + C \sum_{n=1}^{N} a_n \xi_n - \sum_{n=1}^{N} \mu_n \xi_n$$

$$= \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m) \quad (7.32)$$

ハードマージンの時と同じ式

これをある条件のもとで**最小化**すれば良い。ここまでで、KKT条件の式 (7.23-7.28) と停留点の式 (7.29-7.31) が条件として得られているが、ラグランジュ関数の中に現れる変数は a のみなので、 a についての条件だけ考慮すれば良い。

$$a_n \geq 0 \quad (7.23)$$

$$\sum_{n=1}^{N} a_n t_n = 0 \quad (7.29)$$

$$a_n = C - \mu_n \quad (7.31)$$

$$\mu_n \geq 0 \quad (7.26) \quad \leftarrow (7.31) \text{ で } \mu_n \text{ が出てきてしまったので追加}$$

これらは次の2式にまとめられる。

$$0 \leq a_n \leq C \qquad (7.33)$$

$$\sum_{n=1}^{N} a_n t_n = 0 \qquad (7.34)$$

この条件で7.32を最小化する a を求めると、ソフトマージンSVMの学習が完了したこととなる。最小化の方法は後で述べる。

推論時のロジック

推論の際は、$w = \sum_{n=1}^{N} a_n t_n \phi(x_n)$ を $y(x) = w^T \phi(x) + b$ 式に代入した次の式を使う。

$$y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b \qquad (7.13)$$

既知　　未知

パラメータ b はハードマージンSVMと同様に、一部のデータのみを使って算出する。「一部のデータ」とは具体的に $t_n y(x_n) = 1$ が成り立つデータのことである。この場合等号による方程式で簡単に b が求まる。

結果から述べると、このようなデータは $0 < a_n < C$ を満たすデータである。

なぜなら、

$a_n < C$ $\overset{(7.31)}{\Rightarrow}$ $\mu_n > 0$

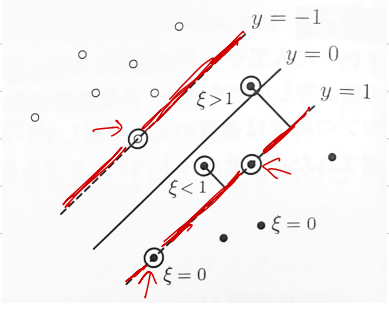
$\overset{(7.28)}{\Rightarrow}$ $\xi_n = 0$ ―― ①

$0 < a_n$ $\overset{(7.25)}{\Rightarrow}$ $t_n y(x_n) - 1 + \xi_n = 0$

$\overset{①}{\Rightarrow}$ $t_n y(x_n) = 1$

成り立って欲しい条件が成り立つ

視覚的には、境界上に存在するデータのことである。



ここでも数値計算上の誤差を小さくするために、$0 < a_n < C$ を満たすすべてのデータで平均を取る。

$$b = \frac{1}{N_M} \sum_{n \in M} \left(t_n - \sum_{m \in S} a_m t_m k(x_n, x_m) \right)$$

> （疑問）
> 教科書に合わせて $\mathcal{M}$ と S を分けて書いたが、$\mathcal{M} = S$ なのではないかと思っている。

ν-SVM の詳細については割愛する。ハイパーパラメータC をデータから決定する方法の一つのように思われる。

ラグランジュ関数 $\tilde{L}(a)$ の最小化問題を解く

最後に、ラグランジュ関数の最小化問題について、課題と解決法の概要解説する。

$$\min. \quad \tilde{L}(a) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(x_n, x_m)$$

$$\text{s.t.} \quad 0 \leq a_n \leq C$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

目的関数は2次式であるので局所解は大局解に一致する。伝統的な二次計画法（ニュートン法とか？）は、計算コストおよびメモリ使用量が莫大になる問題がある（らしい）。

> （感想）
> ニュートン法はヘッセ行列を用いる関係で、データ数 × データ数の行列が必要となり、メモリ使用量が膨大になることは確かに想像つく。

この問題に対して、次の手法が存在する。

■ **チャンキング**　(1982)

■ **分解法** (1996)

■ **逐次最小問題最適化法** (SMO; sequential minimal optimization) (1999)

現在は SMO が最も実用的らしいので、SMOについてのみ紹介する。ロジックの詳細は以下を参照

Wikipedia：https://ja.m.wikipedia.org/wiki/%E9%80%90%E6%AC%A1%E6%9C%80%E5%B0%8F%E5%95%8F%E9%A1%8C%E6%9C%80%E9%81%A9%E5%8C%96%E6%B3%95

ブログ：https://fussy.web.fc2.com/algo/pattern6_svm.htm

通常の二次計画法のアルゴリズムでは、すべての $\{a_n\}$ について同時に最適化することを考えるが、SMOでは二つの乗数 a_i , a_j についてのみ最適化し、2つの乗数の組み合わせを変えながら繰り返し最適化を実施する。このとき、対象とする2つの乗数以外は定数として扱う。（詳しくはブログ参照）

次元の呪いについての話

カーネル法は高次元の特徴空間での内積に相当すると述べてきた。したがって、特徴空間を陽に扱わずカーネル関数のみで記述することは、一見すると次元の呪いから逃れているように思われる。しかし、実際にはそのようなメリットはなく、特徴空間で扱う場合もカーネル法で扱う場合も実質的な次元数は変わらない。

（感想）

ここでの内容は補足的な内容であり、あまり重要ではなさそう。陽に多次元特徴空間で扱うよりもSVMの方が次元の呪いを抑えられる、という誤解が多かったのだろうか。

例えば、2次の多項式カーネルについて、カーネル関数で扱う場合は二次元特徴空間で扱うことができる。

$$k(x, z) = (1 + x^T z)^2$$

これを基底関数の特徴空間で扱うとすると、一見6次元空間として扱っているように思われる。

$$k(x, z) = \phi(x)^T \phi(z)$$
$$= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)(1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1z_2, z_2^2)^T$$

しかし、基底関数の特徴空間は、6次元特徴空間中に存在する2次元多様体上に写像されるため、次元の呪いの影響を受けることはない。
というのも、ここでいう次元の呪いとは、次元が増えた場合に空間を充足するための訓練データが必要になるという問題である（上巻34頁下段）。特徴空間が6次元の場合でも、実際のデータはより低次元の多様体に写されるので、空間を充足するほどの多くのデータは必要にならない、という理解をした。