PRML 7~7.1.1 + 付録E

5501 酒井一徳
5/2/18

今日の内容

7　疎な解を持つカーネルマシン

7.1 最大マージン分類器 + 付録E

7.1.1 重なりのあるクラス分布

疎な解を持つカーネルマシン

疎な解

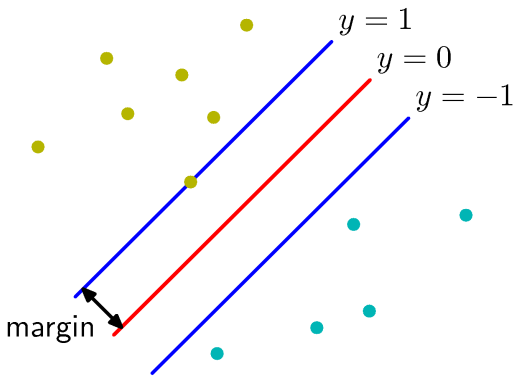
疎な解(sparse solution):　訓練データの一部だけに対して計算し，求まる解

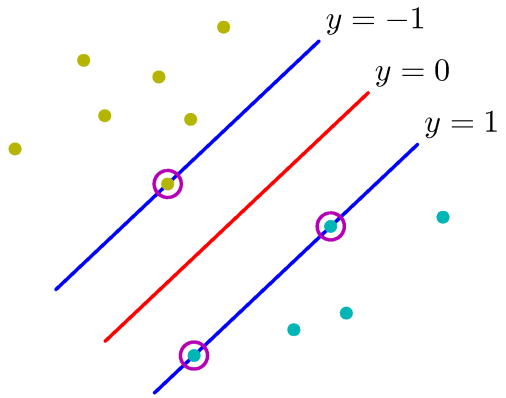
前章では…

例えばガウス過程などは訓練データ全ての対についてカーネル関数の計算が必要だった.
学習時，もしくは**予測時に**非常に時間がかかる可能性がある

SVMの特徴

- 訓練データを特徴空間において分類する
- 正例と負例の境界にあるもの(サポートベクトル)だけを予測に使用する
- サポートベクトルとの距離(Margin)が最大となる分類境界を求める
- モデルパラメータが凸最適化問題の解として求まる
- 確率的出力は一切ない





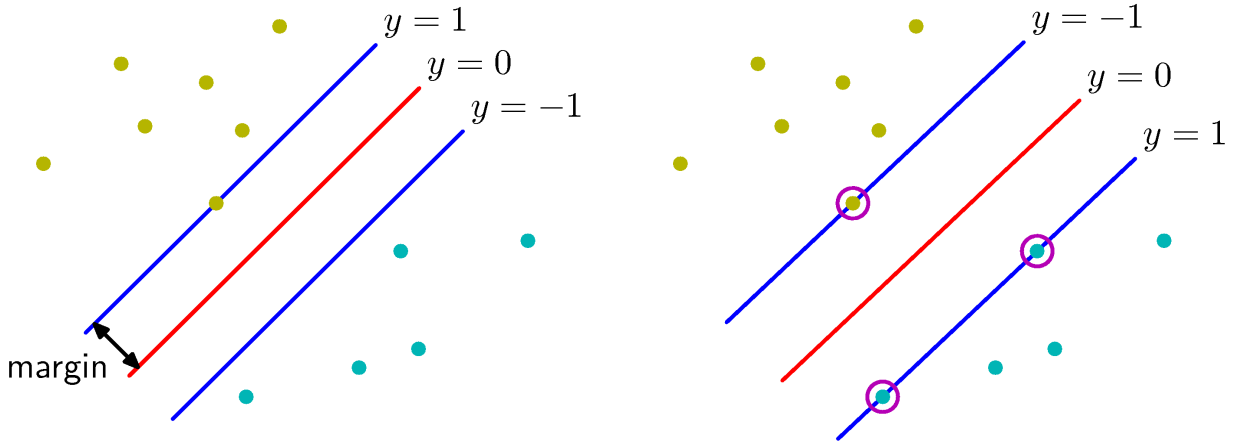
最大マージン分類器

解の非一意性と汎化誤差最小

一般に線形分離可能なデータに対し，その解は多数存在し得る．

実際に予測する際の誤差(汎化誤差)を最小にする解が望ましい

マージン(margin)の
概念を導入



マージン最大化を行う動機

➢ なぜサポートベクトルに対してのみマージンを最大化すれば良いのだろう？

共通のパラメータ σ^2を持つガウスカーネルを用いたParzen推定法で各クラスごとの入力ベクトル $\mathbf{x}$ の分布を推定する．今クラスラベル t について，

$$p(\mathbf{x}|t) = \frac{1}{N_t} \sum_{n=1}^{N} \frac{1}{Z_k} \exp\left\{ -\frac{\|\mathbf{x} - \mathbf{x}_n\|^2}{2\sigma^2} \right\} \delta(t, t_n)$$

s.t.

$$\delta(t, t_n) = \begin{cases} 1 & \text{if } t = t_n \\ 0 & \text{otherwise} \end{cases}, \qquad N = \sum_t N_t, \qquad Z_k = (2\pi\sigma^2)^{D/2}$$

であり，Nはデータの数である．ここでクラスの事前分布が求まるのであれば，

ベイズの定理 $p(t|\mathbf{x}) \propto p(\mathbf{x}|t)p(t)$ から決定境界は求まる．

マージン最大化を行う動機

クラス事前分布を無情報であるときに，誤分類が少ない事後分布の選択はモデルの選択と等価である．
その分類境界は2クラス分類の時，

$$p(\mathbf{x}|t=-1) = p(\mathbf{x}|t=1)$$

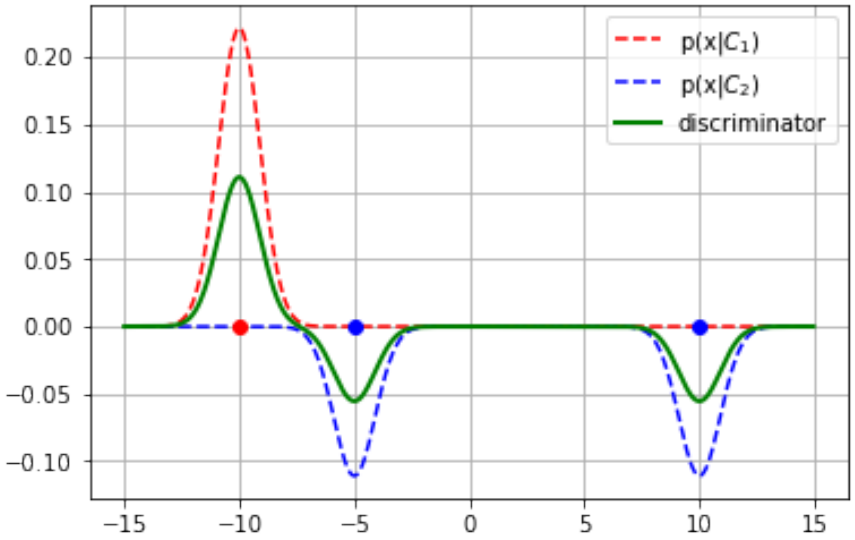
で与えられる．よって，

$$\frac{1}{N_{-1}} \sum_{n:t_n=-1} \frac{1}{Z_k} \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|}{2\sigma^2}\right\} = \frac{1}{N_{+1}} \sum_{n:t_n=+1} \frac{1}{Z_k} \exp\left\{-\frac{\|\mathbf{x}-\mathbf{x}_n\|}{2\sigma^2}\right\}.$$

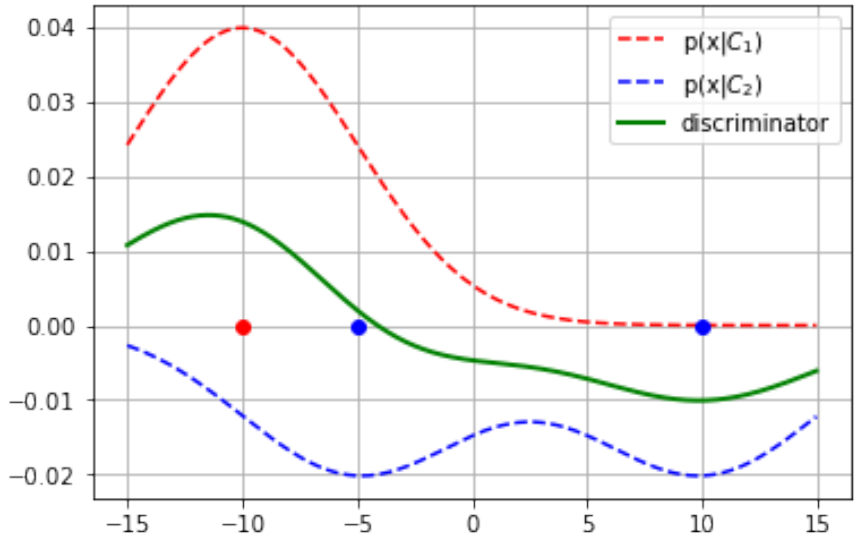
$\sigma^2 \to 0$　の極限を考えると，

マージン最大化を行う動機

分散小



分散大



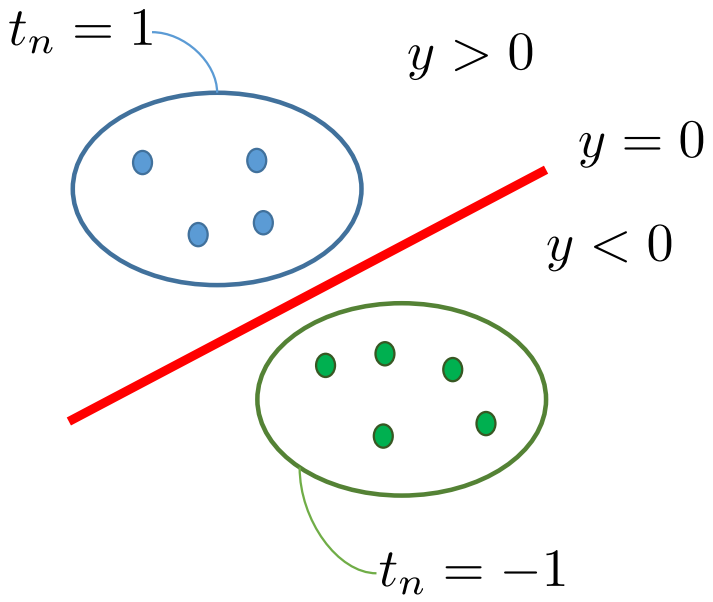
サポートベクトル以外のデータ点の影響が少なくなる.

マージン最大化の定式化

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b.$$

上記の線形モデルと用いて 2 値分類を解くことを考える.
訓練データは,

入力データ	$\mathbf{x}_1, \ldots, \mathbf{x}_N$
対応する目的値	$t_1, \ldots, t_N (t_n \in \{-1, 1\})$



今，訓練データは特徴空間で線形分離可能とするため,

正しく線形分離できるデータについて以下が成立するとする.

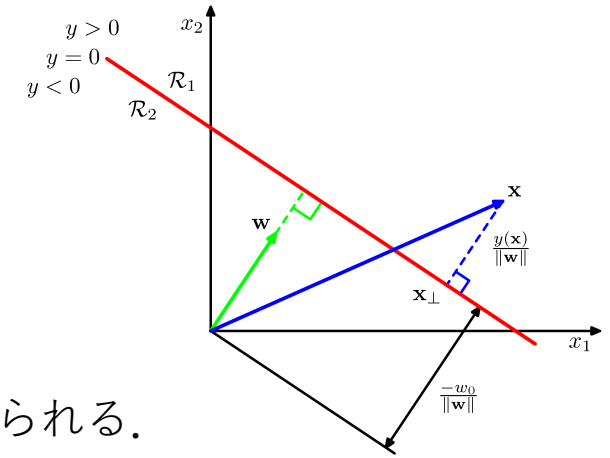
$$t_n y(\mathbf{x}_n) > 0.$$

マージン最大化の定式化

超平面 $y(\mathbf{x}) = 0$ から点 $\mathbf{x}$ までの距離は $|y(\mathbf{x})| / \|\mathbf{w}\|$ で与えられる.

$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b$, $t_n y(\mathbf{x}_n) > 0$ から分類境界から点 $\mathbf{x}$ までの距離は次のようになる.

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}.$$



マージンとは訓練データと(正しく分類する)分類境界との
最短距離である.
そのマージンを最大にするパラメータは以下の最適化問題によって得られる.

$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \right] \right\}.$$

マージン最大化の定式化

パラメータ $\mathbf{w}, b$ を同じ値だけ定数倍しても前述の目的関数の値に影響がない,
よって適当な定数をかけることで分類境界に最も近い点(サポートベクトル)について,

$$\min_n \left[t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \right] = 1$$

とできる．この時マージン最適化の問題は,

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t.$$

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geqslant 1, \qquad n = 1, \ldots, N.$$

制約式の等号が成立する場合，この制約は**有効**であるという．
この問題は**二次計画法**の一例である．

ラグランジュ乗数(上巻付録E)

ラグランジュ乗数と問題例

ラグランジュ乗数(Lagrange multiplier) (未定乗数とも)

複数の変数に1つ以上の制約条件が課されたときに，(ラグランジュ)関数の停留点を求めるため用いられる.
例えば以下の問題において,

$$\max_{x_1,x_2} f(x_1,x_2), \quad s.t. \quad g(x_1,x_2)=0$$

制約条件から $x_2 = h(x_1)$ のような表現を見つけ，最大化問題を1変数に変えるアプローチが考え得るだろう．しかし,

- いつも解析的に陽に表現できるとは限らない.
- 元の問題の対称性を活かせておらず，エレガントじゃない.

より手軽で，エレガントな手法を

ラグランジュ乗数法の幾何学的解釈

D 次元の変数 $\mathbf{x} = (x_1, \ldots, x_D)$ を考えると，$g(\mathbf{x}) = 0$ とは $(D-1)$ 次元の曲面となる．

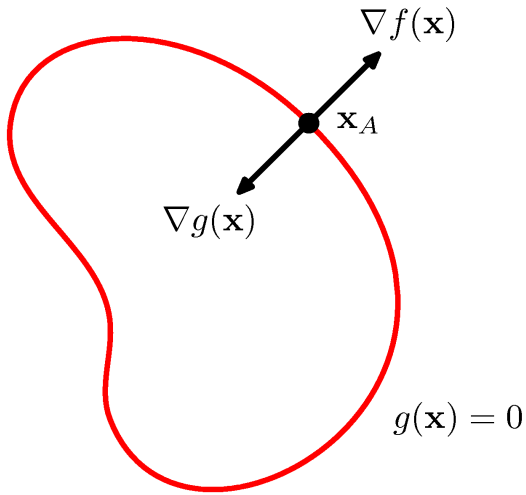
制約面 $g(\mathbf{x}) = 0$ は勾配 $\nabla g(\mathbf{x})$ と常に直交し (後述)，また制約面上で $f(\mathbf{x})$ を最大化する $\mathbf{x}^\star$においては $\nabla f(\mathbf{x})$ とも直交する必要がある．

つまりあるパラメータ$\lambda \neq 0$ が存在し，

$$\nabla f + \lambda \nabla g = 0$$

が成立する必要がある．

λはラグランジュ乗数 (Lagrange multiplier) と呼ばれる．



ラグランジュ関数

ここで以下で定義される**ラグランジュ関数(Lagrangian)**

$$L(\mathbf{x}, \lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

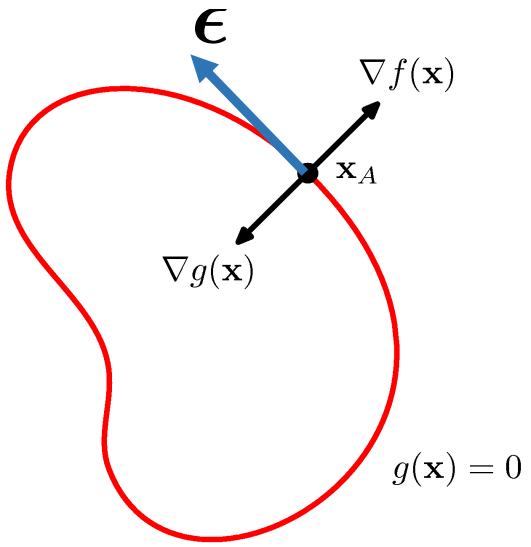
を導入することで，停留条件と制約式は以下で表現できる.

$$\frac{\partial L}{\partial \mathbf{x}} = \frac{\partial L}{\partial \lambda} = 0$$

勾配が直交する理由

制約面 $g(\mathbf{x}) = 0$ 上の 2 点 $\mathbf{x}, \mathbf{x}+\boldsymbol{\epsilon}$ を考える．
$\mathbf{x}$ の周りでの $g(\mathbf{x})$ のテイラー展開より，以下が得られる．

$$g(\mathbf{x}+\boldsymbol{\epsilon}) \simeq g(\mathbf{x}) + \boldsymbol{\epsilon}^{\mathrm{T}} \nabla g(\mathbf{x}).$$



2 点 $\mathbf{x}, \mathbf{x}+\boldsymbol{\epsilon}$ は制約面上にあるので $g(\mathbf{x}) = g(\mathbf{x}+\boldsymbol{\epsilon})$，よって

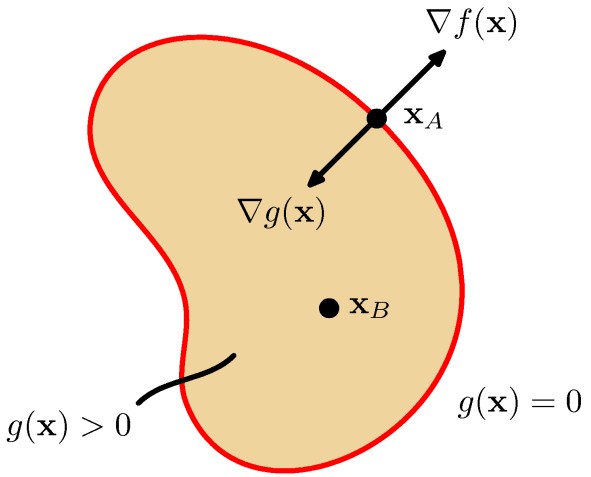
$$\boldsymbol{\epsilon}^{\mathrm{T}} \nabla g(\mathbf{x}) \simeq 0.$$

$\|\boldsymbol{\epsilon}\| \to 0$ の極限においては，　$\boldsymbol{\epsilon}^{\mathrm{T}} \nabla g(\mathbf{x}) = 0$ が成立し，
かつ $\boldsymbol{\epsilon}$ は制約面の接線であるから $\nabla g(\mathbf{x})$ は制約面に対し垂直である．

不等式制約の解のパターン

制約が $g(\mathbf{x}) \geqslant 0$ という不等式制約で与えられた際の $f(\mathbf{x})$ の最大化問題を考える.

このとき解は次の二通りに分類できる.



① $g(\mathbf{x}) > 0$ の領域にある場合…**無効制約**

制約領域にあるならば単に $\nabla f(\mathbf{x}) = 0$ を満たせば良いだけ.

② $g(\mathbf{x}) = 0$ の上にある場合…**有効制約**

$\nabla f(\mathbf{x})$ は制約領域の外を向く必要がある. 何故なら制約領域の方向を向いているならば, 制約領域内で停留点を取りうるためである. つまりある $\lambda > 0$ が存在し,

$$\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$$

が成立する必要がある.

Karush-Kuhn-Tucker条件

$$\max_{\mathbf{x}} f(\mathbf{x}) \quad s.t. \quad g(\mathbf{x}) \geqslant 0.$$

前述から，上記の問題は以下の条件の元でのラグランジュ関数の停留点を求める問題になる.

$$L(\mathbf{x}, \lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

KKT条件

$$\begin{cases} g(x) \geqslant 0 \\ \lambda \geqslant 0 \\ \lambda g(\mathbf{x}) = 0 \end{cases}$$

$\lambda = 0$
$g(\mathbf{x}) = 0$
のどちらかが成り立つという意味.
解が制約内部か制約面上かどちらかであるという意味でもある.

最大化ではなく最小化であればラグランジュ関数を以下に変更し最小化すれば良い.

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x})$$

複数条件への拡張

例として以下の複数の制約の下での最大化問題を考える.

$$\max_{\mathbf{x}} f(\mathbf{x}) \qquad s.t. \qquad g_j(\mathbf{x}) = 0 \ (j = 1, \ldots, J),$$
$$h_k(\mathbf{x}) \geqslant 0 \ (k = 1, \ldots, K).$$

この場合，複数のラグランジュ乗数 $\{\lambda_j\}$ および $\{\mu_k\}$ を導入し
以下のラグランジュ関数を最適化すれば良い.

$$L\left(\mathbf{x}, \{\lambda_j\}, \{\mu_k\}\right) = f(\mathbf{x}) + \sum_{j=1}^{J} \lambda_j g_j(\mathbf{x}) + \sum_{k=1}^{K} \mu_k h_k(\mathbf{x}).$$

このときのKKT条件は以下である.

$$\mu_k \geqslant 0,$$
$$h_k(\mathbf{x}) \geqslant 0,$$
$$\mu_k h_k(\mathbf{x}) = 0. \qquad (k = 1, \ldots, K)$$

再訪：最大マージン分類器

ラグランジュ乗数の導入

ラグランジュ乗数 $\mathbf{a} = (a_1, \ldots, a_N)^{\mathrm{T}} (a_n \geqslant 0)$ を導入することで，
次のラグランジュ関数を得る．

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n \left(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) + b \right) - 1 \right\}.$$

$$\begin{aligned} & \arg\min_{\mathbf{w}, b} \frac{1}{2}\|\mathbf{w}\|^2 \\ s.t. \quad & \\ & t_n(\mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) + b) \geqslant 1, \\ & n = 1, \ldots, N \end{aligned}$$

$\mathbf{w}, b$ について最小化，$\mathbf{a}$ について最大化を行う．

$\mathbf{w}, b$ についての微分を 0 とおくと，

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n) \qquad (\mathbf{w} \text{ を } \mathbf{a} \text{ の線形結合で表現している})$$

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

が得られる．これらの式をラグランジュ関数に代入することで，
ラグランジュ関数の**双対表現**が得られる．

マージン最大化の双対問題

マージン最大化の最適化問題は以下の目的関数の $\mathbf{a}$ についての最大化問題となる.

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

$$s.t.$$

$$a_n \geqslant 0, \qquad n = 1, \ldots, N, \qquad \sum_{n=1}^{N} a_n t_n = 0.$$

$$where$$

$$k(\mathbf{x}, \mathbf{x}') := \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}')$$

この問題は再び**二次計画法**になっていることに注意する.

解く方法は後述(7.1.1節)

意味があったのか

主問題

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

$$s.t.$$

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b) \geqslant 1, \quad n=1,\ldots,N$$

変数 (基底関数) の数 M

双対問題

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

$$s.t.$$

$$a_n \geqslant 0, \qquad n=1,\ldots,N, \qquad \sum_{n=1}^{N} a_n t_n = 0.$$

変数 (データ点) の数 N

M 個の変数を持つ二次計画法を解くには $O(M^3)$

基底関数の数 M をデータ点の数 N が上回るならば，カーネルで取り扱える利点もあり効率的に扱える．
カーネルが正定値ならばラグランジュ関数$\tilde{L}(\mathbf{a})$ は上に有界の凸最適化問題となる．

※上に有界: bounded above
pdfでは　“bounded below”となっていた

サポートベクトル

KKT条件から以下が言える.

$$
\begin{aligned}
a_n &\geqslant 0 \\
t_n y(\mathbf{x}_n) - 1 &\geqslant 0 \\
a_n \left\{ t_n y(\mathbf{x}_n) - 1 \right\} &= 0.
\end{aligned}
$$

$$
\begin{aligned}
&\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\
s.t. \quad & \\
& t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geqslant 1, \\
& n = 1, \ldots, N
\end{aligned}
$$

ここで，全てのデータ点について $a_n = 0$ あるいは $t_n y(\mathbf{x}_n) = 1$ が成立する.

$a_n = 0$ となるデータ点は右記から予測に影響を及ぼさない.

$a_n \neq 0$ となるデータ点は**サポートベクトル**と呼ばれ，マージンの縁に存在する.

サポートベクトルは $t_n y(\mathbf{x}_n) = 1$ が成立するデータ点でもある.

$$
\begin{aligned}
y(\mathbf{x}) &= \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b. \\
\mathbf{w} &= \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)
\end{aligned}
$$

分類規則

学習したモデルを用いてデータを分類するには，**求まった**パラメータ $\{a_n\}$ とカーネル関数で表される以下の関数の符号で判断できる．

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}) + b.$$

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)$$

$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b.$$

$$b = \frac{1}{N_{\mathcal{S}}} \sum_{n \in \mathcal{S}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right).$$ ($\mathcal{S}$ はサポートベクトルの添字の集合)

$N_{\mathcal{S}}$ はサポートベクトルの総数である．

バイアスパラメータ b の導出については任意のサポートベクトル $\mathbf{x}_n$ について $t_n y(\mathbf{x}_n) = 1$ が成立することから右記の式を用いた．

$$t_n \left(\sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1$$

ハードマージンSVM

SVMの最大マージン学習は次の誤差関数の最小化に等しい.

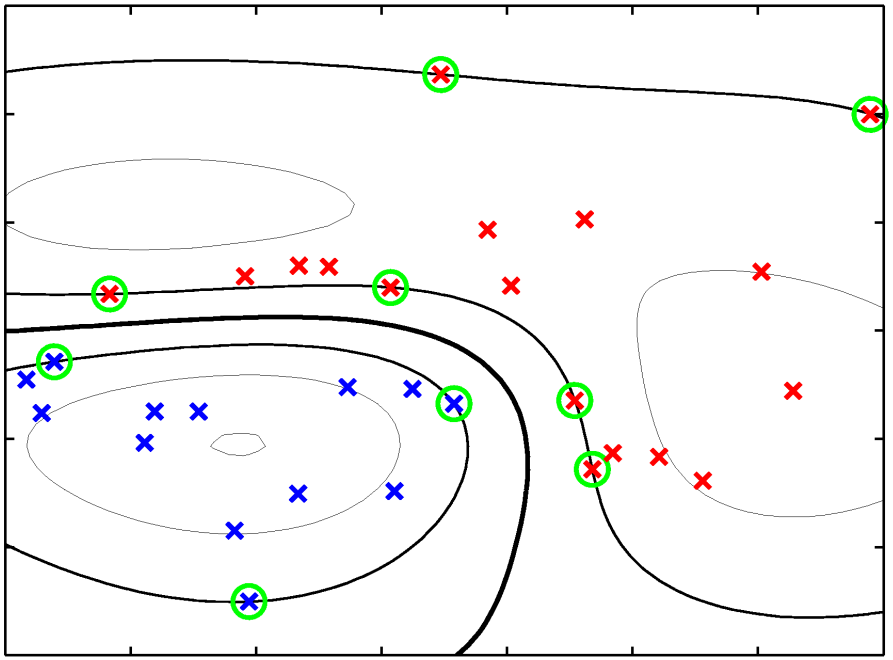
$$\sum_{n=1}^{N} E_\infty(y(\mathbf{x}_n)t_n - 1) + \lambda\|\mathbf{w}\|^2.$$

$$E_\infty(z) \equiv \begin{cases} 0 & (z \geqslant 0) \\ \infty & (otherwise) \end{cases}$$

正則化パラメータが $\lambda > 0$ である限り，解は一意に収束する.

誤分類に対して無限のペナルティを与えている.
のちのソフトマージンと比較してハードマージンと呼ばれる.

非線形SVMの適用例



ガウスカーネルを用いたSVMによる
2クラス分類.

特徴空間においては線形分離可能なデータ
を用いた.

分類境界(濃い太線)の位置はサポート
ベクトルの位置のみに依存する.

重なりのあるクラス分布

完全分類と汎化性能

クラスの条件付き確率分布が重なっている場合も考えうる.

訓練データに対して完全分離する解が必ず汎化性能に長けるとは限らない.

一部の訓練データに対して誤分類も許すSVMを構築したい.

ハードマージンSVMは誤分類するデータに無限大のペナルティを与えているも同義なので，距離に応じたペナルティに変更しよう

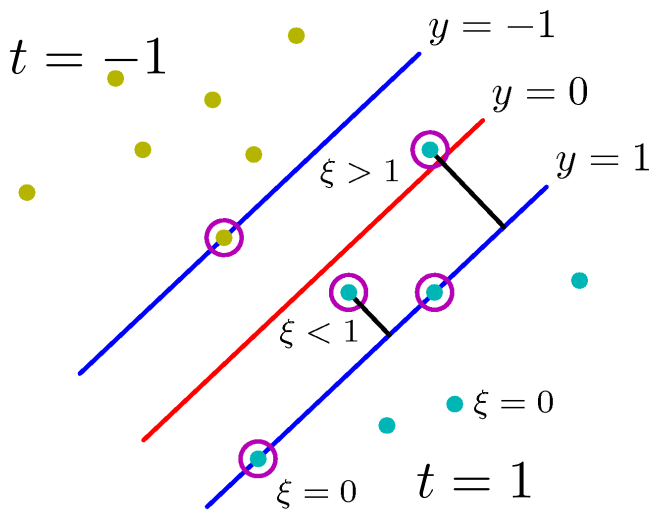
スラック変数

不等式における両辺の差，余裕を表す変数．スラック変数を用いて不等式制約 $g(x) \leq 0$ を等式制約 $g(x) + \xi = 0$ と非負条件 $\xi \geq 0$ で表現できる．

ただし，今回はスラック変数をペナルティとして導入するため以下のように定義される．

$$\xi_n \equiv \begin{cases} 0 & (\text{正しく分類され，マージン境界の上もしくは外側に位置する}) \\ |t_n - y(\mathbf{x}_n)| & (otherwise) \end{cases}$$

$$(n = 1, \ldots, N)$$



制約条件の修正(ソフトマージンへの緩和)

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b)\geqslant 1,\quad n=1,\ldots,N.$$

誤分類を許さない
ハードマージンの制約式

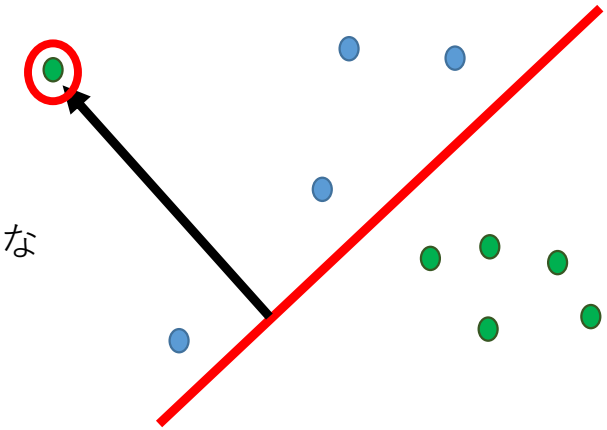
緩和

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)+b)\geqslant 1-\xi_n,\quad n=1,\ldots,N.$$

誤分類をある程度許す
ソフトマージンの制約式

外れ値に頑健になった訳ではないため注意.

$|t_n - y(\mathbf{x}_n)|$
に比例する大きな
ペナルティ



マージンの最大化の定式(ソフトマージンver.)

よってソフトマージンSVMのマージン最大化の目的関数は以下の通りである.

$$\arg\min_{\mathbf{w},b} \left\{ C\sum_{n=1}^{N} \xi_n + \frac{1}{2}\|\mathbf{w}\|^2 \right\}.$$

$$s.t.$$

$$t_n(\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n) + b) \geqslant 1 - \xi_n,$$

$$\xi_n \geqslant 0. \qquad (n = 1, \ldots, N)$$

ただし $C > 0$ はスラック変数を通して表されるペナルティとマージンの大きさの間のトレードオフを制御するパラメータである.

$\sum \xi_n$ は**誤分類されたデータ数の上界**.

C 大 ⇒ エラーを減らそうとする
(モデルが複雑化)

C 小 ⇒ エラーが増えてもいいから
マージンを大きくする

双対問題の導出

ハードマージンと同様にラグランジュ乗数 $\{a_n\}, \{\mu_n\}$ を導入することで,
目的関数のラグランジュ関数は以下となる.

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

対応するKKT条件は以下である.

$$\begin{aligned}
a_n &\geqslant 0 && \\
t_n y(\mathbf{x}_n) - 1 + \xi_n &\geqslant 0 && \text{(サポートベクトルのみ等号)} \\
a_n(t_n y(\mathbf{x}_n) - 1 + \xi_n) &= 0 && \text{(上の二式のどちらかは等号である)} \\
\mu_n &\geqslant 0 && \\
\xi_n &\geqslant 0 && \text{(マージン境界外の点やサポートベクトルは等号)} \\
\mu_n\xi_n &= 0 && \text{(上の二式のどちらかは等号である)}
\end{aligned}$$

ただし, $n = 1, \ldots, N$

双対問題の導出

$\mathbf{w}, b, \{\xi_n\}$ についての停留条件から以下が導かれる.

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{n=1}^{N} a_n t_n \boldsymbol{\phi}(\mathbf{x}_n)$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{n=1}^{N} a_n t_n = 0$$

$$\frac{\partial L}{\partial \xi_n} = 0 \quad \Rightarrow \quad a_n = C - \mu_n$$

双対問題の導出

ソフトマージンSVMの最適化問題は以下の目的関数の最大化問題となる.

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

$$s.t.$$

$$0 \leqslant a_n \leqslant C \quad \text{(矩形制約)}$$

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

解についての解釈

$a_n = 0$ となる点はハードマージンと同様，識別関数に影響を及ぼさない．
それ以外の点，つまりサポートベクトルは以下を満たす．

$$a_n > 0, \quad t_n y(\mathbf{x}_n) = 1 - \xi_n$$

$a_n < C$ のとき，

$a_n = C - \mu_n$ から $\mu_n > 0$

$\mu_n > 0$ と $\mu_n \xi_n = 0$ から $\xi_n = 0$ が言える．

つまり，マージン境界上の点である．

$a_n = C$ のとき，

同様の議論からマージン境界の内側の点であることがわかり，
特に $\xi_n > 1$ のとき誤分類している．

識別関数の決定(パラメータb)

$0 < a_n < C$ が成立するデータ点では $\xi_n = 0$ が成立することから,

$$t_n \left(\sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) + b \right) = 1$$

から理論上は計算ができるが，数値計算の誤差も加味して以下のように平均をとる.

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(t_n - \sum_{m \in \mathcal{S}} a_m t_m k(\mathbf{x}_n, \mathbf{x}_m) \right).$$

$$\mathcal{M} = \{n \in \mathbb{N} | 0 < a_n < C\}$$

$$\mathcal{S} = \{n \in \mathbb{N} | a_n > 0\}$$

外れ値の影響はここでも確認できる.

異なる定式化(ν-SVM)

同値だが異なる定式化の方法として ν-SVM が提案されている.

$$\tilde{L}(\mathbf{a}) = -\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

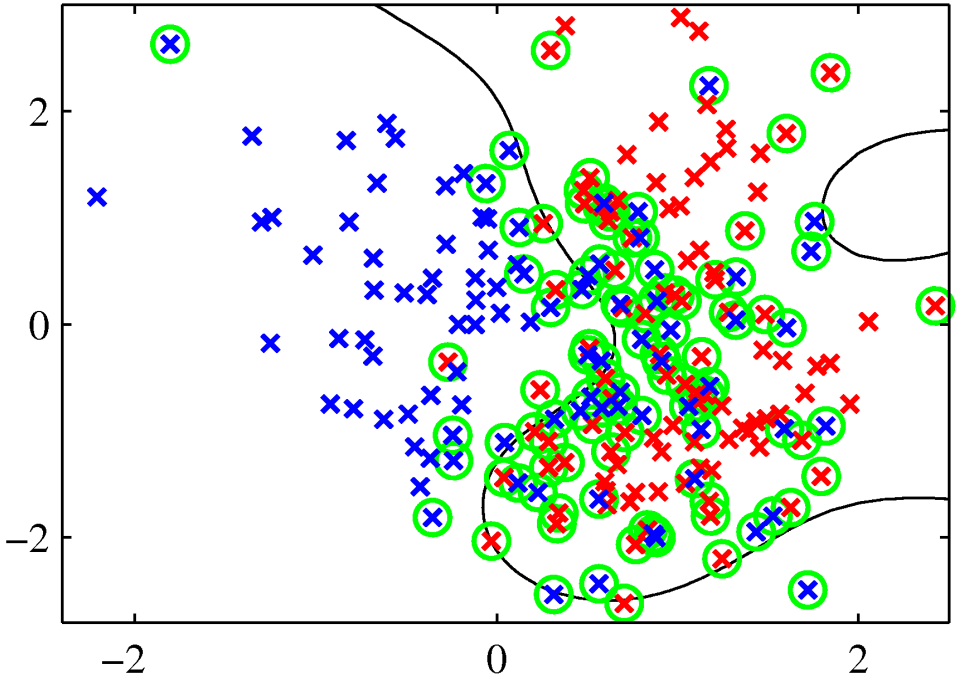
$$s.t.$$

$$0 \geqslant a_n \geqslant 1/N, \quad \sum_{n=1}^{N} a_n t_n = 0, \quad \sum_{n=1}^{N} a_n \geqslant \nu.$$

C の代わりに導入されたパラメータν が訓練データ全体に占める**マージン誤差の割合の上界**として解釈できる.

マージン誤差とは $\xi_n > 0$ となる点であり，マージンの誤った側に存在する点である.

ν −SVMの適用例



暗に定義される特徴空間においても
線形分離不可能なデータに対して
ガウスカーネルを用いたν-SVMを適用した例

ガウスカーネルは以下の形.

$$\exp\left(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2\right),\ \gamma=0.45$$

丸で囲まれた点はサポートベクトル.
誤分類されているものは全てサポートベクトルに
なっていることがわかる.

訓練は結局どうするの

サポートベクトルしか必要ないのは予測時であり，訓練時には全ての訓練データが必要である.
一般にはN個の変数を持つ二次計画問題は$O(N^3)$の時間がかかる.

効率よく解く手法が必要である.

- **チャンキング**(chunking)(Vapnik, 1982)
  - 最終的に0にならないラグランジュ乗数のみ残す.
  - カーネル行列の大きさを非ゼロのラグランジュ乗数の数の 2 乗にまで削減できる.
  - 射影共役勾配法を用いて実装できる.

- **分解法**(decomposition method)(Osuna, 1996)
  - サイズの小さな二次計画問題を繰り返し解くことで，解を得る.
  - 小分けしても，結局は二次計画問題を解くのに数値計算が必要
  - 発展版のSMO(Sequential minimal optimization)が存在する.

SMO(Sequential minimal optimization)

逐次最小最適化アルゴリズム

- 全ての $\{a_n\}$ ではなく２点 a_i, a_j で逐次更新を行う．解析的に解が求まるため速度が出る．
- なぜ二つなのかというと，$\sum_{n=1}^{N} a_n t_n = 0$ の制約から一つ更新したなら少なくともあともう一つは更新せざるを得ないため．
- ２点の選び方はいくつかのヒューリスティックが存在する(詳しくは参考文献(2)を参照のこと)
- 行列演算もないためメモリにも優しい．

SMOアルゴリズム概略

いま選択した２点をa_1, a_2とする．右記の制約から以下を満たす必要がある．

$$a_1^{new} t_1 + a_2^{new} t_2 = a_1^{old} t_1 + a_2^{old} t_2$$

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

$$0 \leqslant a_n \leqslant C$$

さらに右記の矩形制約から新しく以下の制約が導かれる．

$$U \leq a_2^{new} \leq V$$

$t_1 = t_2$ の場合，

$$U = \max(0,\ a_1^{old} + a_2^{old} - C)$$
$$V = \min(C,\ a_1^{old} + a_2^{old})$$

$t_1 \neq t_2$ の場合，

$$U = \max(0,\ a_2^{old} - a_1^{old})$$
$$V = \min(C,\ C - a_1^{old} + a_2^{old})$$

SMOアルゴリズム概略

目的関数(7.32)はa_1, a_2の関数としてみると，以下のように整理できる．

$$\tilde{L}(a_1, a_2) = a_1 + a_2 - \frac{1}{2}k_{11}a_1^2 - \frac{1}{2}k_{22}a_2^2 - t_1t_2k_{12}a_1a_2 - t_1a_1v_1 - t_2a_2v_2 + const$$

ただし，

$$k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$$

$$v_i = \sum_{j=3}^{N} t_j a_j k(\mathbf{x}_i, \mathbf{x}_j)$$

この目的関数を a_2 について微分し，結果を0とおくことで，

SMOアルゴリズム概略

$$a_2^{new} = a_2^{old} + \frac{t_2\left((f(\mathbf{x}_1) - t_1) - (f(\mathbf{x}_2) - t_2)\right)}{k_{11} + k_{22} - 2k_{12}}$$

ただし前述の制約を満たす必要がある. a_1^{new}については以下の式からわかる.

$$a_1^{new}t_1 + a_2^{new}t_2 = a_1^{old}t_1 + a_2^{old}t_2$$

次元の呪い

特徴空間を陽に扱ってないから一見してSVMは次元の呪いを克服しているように思われる.

しかし，特徴空間での実質的な次元数は見かけの上の次元よりも小さくなる．例えば,

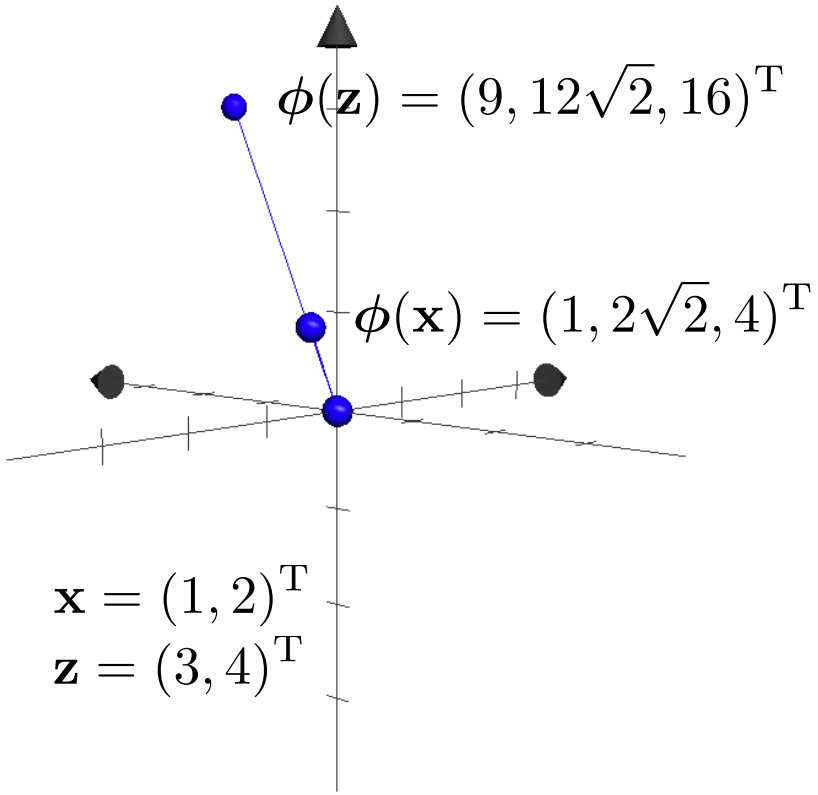
$$
\begin{aligned}
k(\mathbf{x}, \mathbf{z}) &= (1 + \mathbf{x}^{\mathrm{T}}\mathbf{z})^2 = (1 + x_1 z_1 + x_2 z_2)^2 \\
&= 1 + 2x_1 z_1 + 2x_2 z_2 + x_1^2 z_1^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\
&= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1 x_2, x_2^2)(1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, \sqrt{2}z_1 z_2, z_2^2)^{\mathrm{T}} \\
&= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{z})
\end{aligned}
$$

上記のカーネル関数は 2 次元ベクトルを 6 次元の特徴ベクトルに写像した後，内積をとったもの.
入力ベクトルは 6 次元特徴空間中に存在する 2 次元非線形多様体に写像される.

6 次元とかビジュアライズできないので 3 次元で試してみると…

次元の呪い

$$
\begin{aligned}
k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}^{\mathrm{T}}\mathbf{z})^2 = (x_1 z_1 + x_2 z_2)^2 \\
&= x_1^2 z_1^2 + 2 x_1 z_1 x_2 z_2 + x_2^2 z_2^2 \\
&= (x_1^2, \sqrt{2} x_1 x_2, x_2^2)(z_1^2, \sqrt{2} z_1 z_2, z_2^2)^{\mathrm{T}} \\
&= \boldsymbol{\phi}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{z})
\end{aligned}
$$



２次元の入力ベクトルを３次元特徴空間に写像した後に内積をとったと言えるが,
結局３次元特徴空間中の２次元空間に写像されている.

確率的予測

SVMは確率的な出力がない.

より大きな確率的な予測システムの一部とするならば，分類される確率は必要である，

➢ ロジスティックシグモイド関数をSVMの出力に適用する方法が提案された．(Platt, 2000)

$$p(t=1|\mathbf{x}) = \sigma\left(Ay(\mathbf{x}) + B\right)$$

- 2クラス分類問題の際の，求めたい条件付き確率を上式とした.
- パラメータA,Bはある訓練データ上で予測と正解ラベルのクロスエントロピー誤差が最小になるように定める.
- パラメータ決定のためのデータはSVMの学習のためのデータとは独立である必要がある.
- 識別関数が対数オッズに相当すると仮定することと等しい.
- 得られる確率は良い近似とならない可能性がある(Tipping, 2001).

終わり

参考

1） 高村大也(2010)『言語処理のための機械学習入門』奥村学監修，コロナ社.
2） Nello Cristianini(2005)『サポートベクターマシン入門』共立出版