機械学習エンジニアコース
Sprint

－ SVM －

DIVE INTO CODE

今回のモチベーション

目的はなにか

1. 統計モデルを知る
   スクラッチを通してSVMを理解する

2. 線形モデルと異なる手法に触れる

このスライドは?

ここでは、SVMの基本的な知識を学びましょう

SVMとはなにか

分類器としてのSVMは、入力に対し、直接クラスラベルを返すことから**識別関数**と呼ばれる。これに対し、ロジスティック回帰やニューラルネットワーク（分類器）の場合は、入力に対しクラスの条件付き確率（ $P(C=y_i|\mathbf{x}_i;\theta)$ ）を返す。クラスの条件付き確率（事後確率）をモデル化することから、こちらは**分類モデル**と呼ばれる。分類モデルは、しきい値の設け方によってどのクラスに属するかを調整することができる。

この課題の対象者

①scikit-learnの分類モデルを用いて、学習、推定するコードが書ける方

②分類モデルの計算過程(仮定関数、目的関数、最急降下法)を知っている方(1)

(1) sprint ロジスティック回帰スクラッチを解いた方

この後の流れ

SVMの問題設定を知る

① 予測値を導く式（仮定関数）をたてる

② 最小化する問題を設定し、式（目的関数）をたてる

③ 主問題の目的関数を双対問題の目的関数に置き換える

④ 目的関数の最適解を探索的に求める(勾配法)

⑤ 得られた疎な解(サポートベクトル)を用いて推定する

この後の流れ

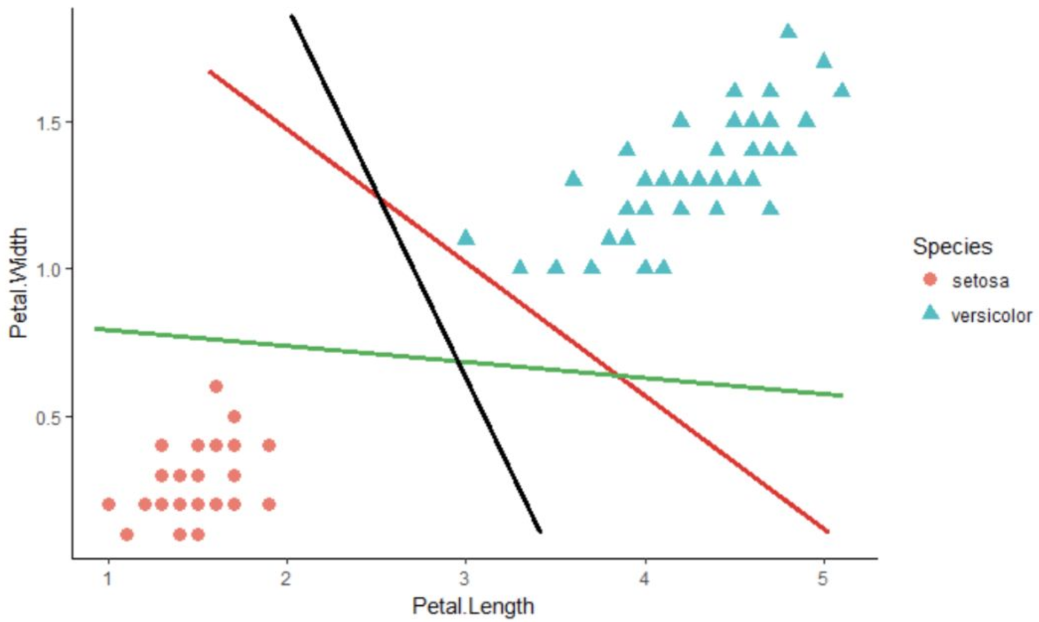
Iris data

いまここにIrisデータセットがあるとしよう。

データ点はあらかじめクラスごとに色分けされている。

ある特徴量X_1 (petallength) と特徴量X_2（petalwidth）を選び、

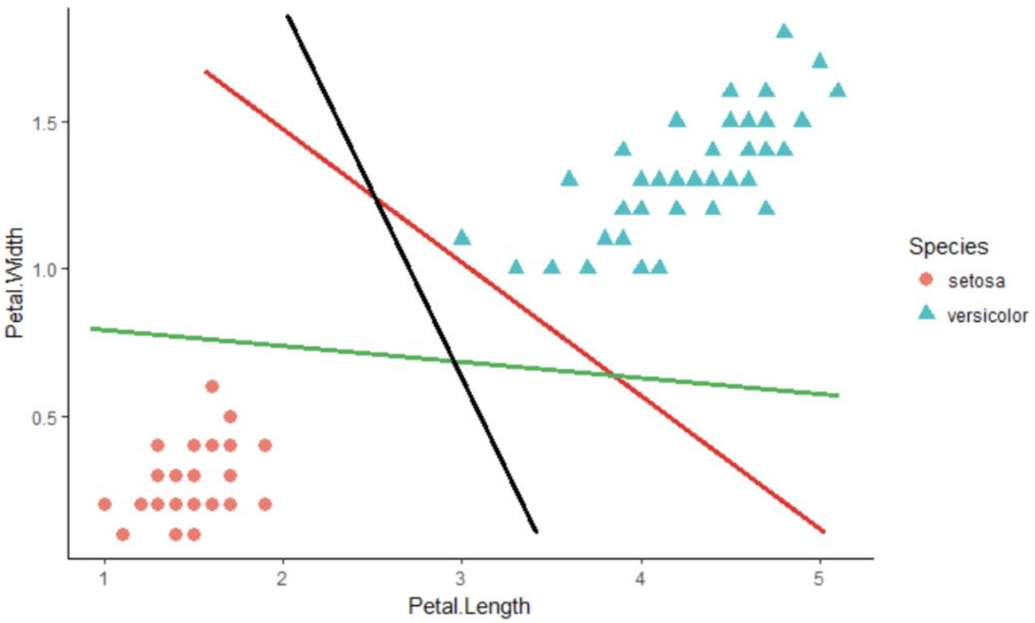
二変数間の関係 をプロットしてみよう。



この後の流れ

今回もIris-setosaとIris-versicolorのクラスを分類するための識別境界（境界線）が引けると嬉しいが、SVMはロジスティック回帰のように**すべてのデータ点に対してクラスに帰属する確率を出力することはない。**

では、どのような仕組みによって識別境界を引くのだろうか。



この後の流れ

SVMの問題設定を知る

① 予測値を導く式（仮定関数）をたてる

② 最小化する問題を設定し、式（目的関数）をたてる

③ 主問題の目的関数を双対問題の目的関数に置き換える

④ 目的関数の最適解を探索的に求める(勾配法)

⑤ 得られた疎な解(サポートベクトル)を用いて推定する

この後の流れ

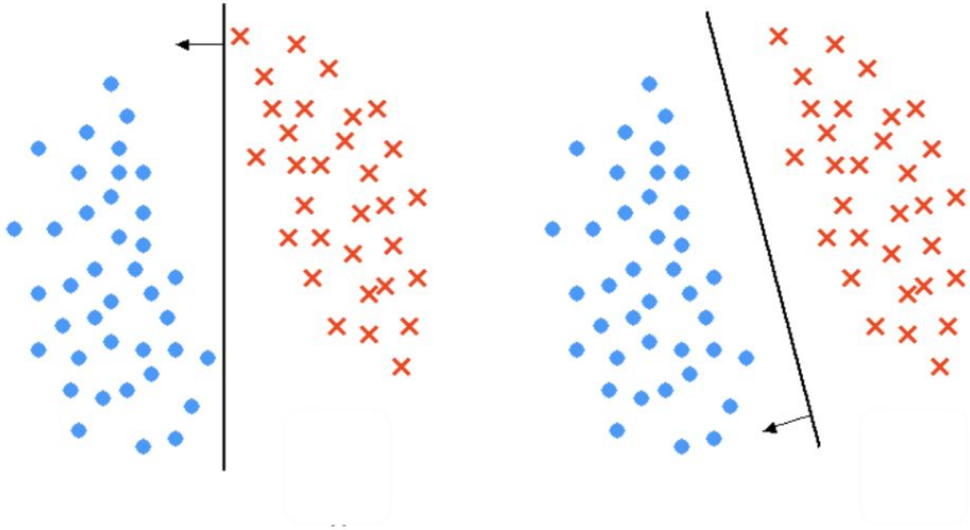
今回の仮定関数

今回も識別境界を引くために次のような仮定関数を想定する。

符号関数（sign）に線形結合を入力して得られる出力：

線形結合： $w^T x$

$$y = \mathrm{sign}(w^T x)$$



$$y \in \{+1(\bullet), -1(\times)\}$$

この後の流れ

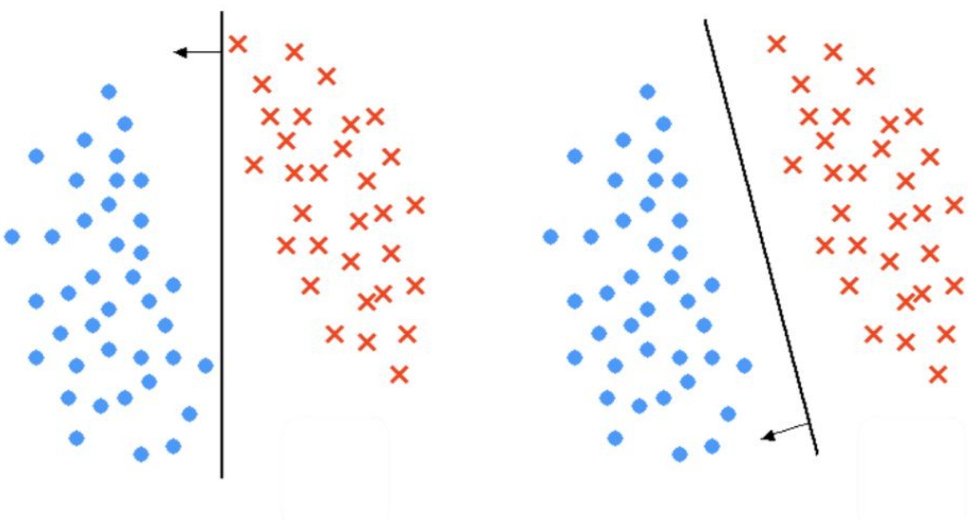
sign関数は次のような出力を返す。

$$\mathrm{sign}(\mathrm{w}^T x) = \begin{cases} +1 & \text{if } (\mathrm{w}^T x) \geqq 0 \\ -1 & \text{otherwise} \end{cases}$$

y ∈ {-1, 1} であるから、条件をまとめて $\boxed{y\,(\mathrm{w}^T x) \geqq 0}$ と書くことができる。

※ のちの双対問題を解き、疎な解（サポートベクトル）を得てはじめてこの仮定関数にて推定が可能になる。

y = 1のとき(w x)は正なので成立する
y= -1 のとき(w x)は負なので成立する



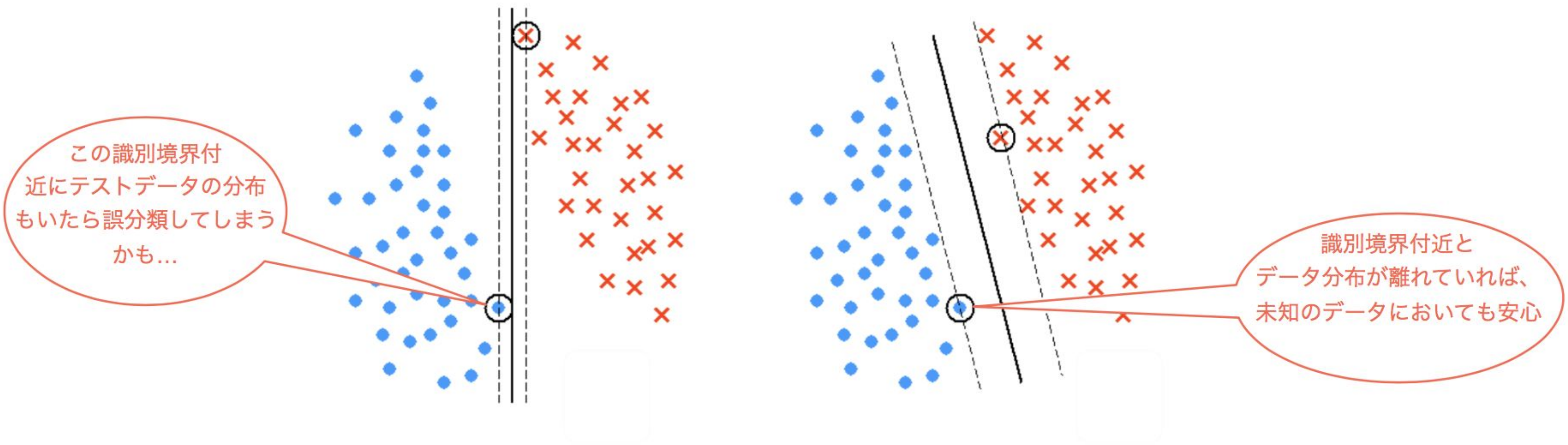
$y \in \{+1(\bullet), -1(\mathbf{x})\}$

この後の流れ

識別境界からの距離

先ほどの仮定関数を用い、1と-1のクラス領域に分けるという前提から出発しよう。それでは、どのようにパラメータを制御し、識別境界の位置を決めるのが良いだろうか。

この問いへの答えは、**もっとも近いサンプルとの距離**（マージン）**が最大**となるような識別境界を求めることである。



$y \in \{+1(\bullet), -1(\times)\}$

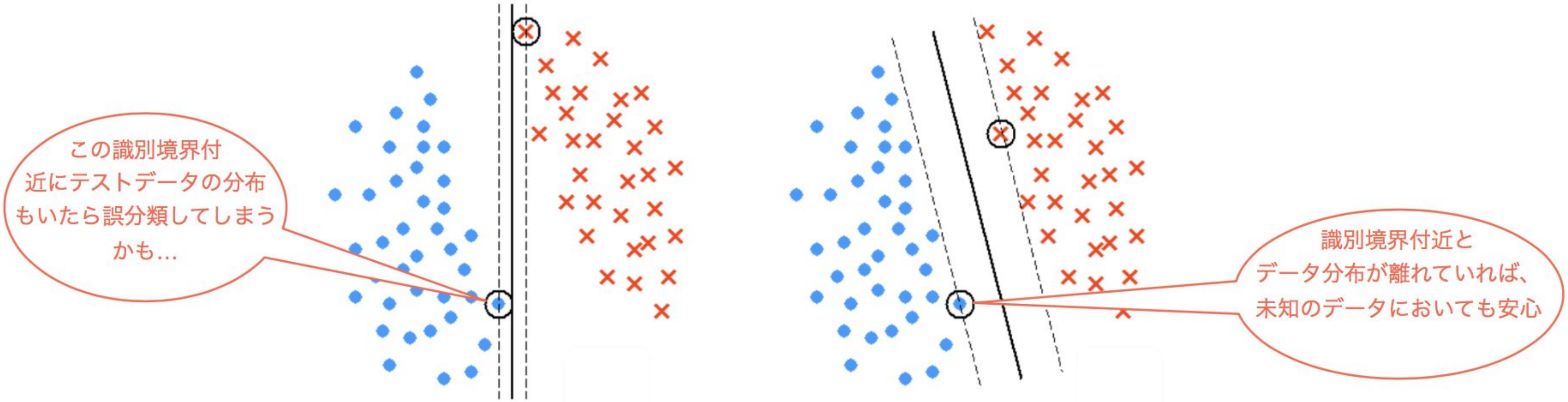
この後の流れ

したがって、識別境界からの距離を計算する必要がある。

識別境界を $y(\mathrm{w}^T x) = 0$ とすると、識別境界から各サンプル(x_i) までの距離は、$\dfrac{y_i\left(\mathrm{w}^T x_i\right)}{\|\mathbf{w}\|}$ で与えられる(1)。

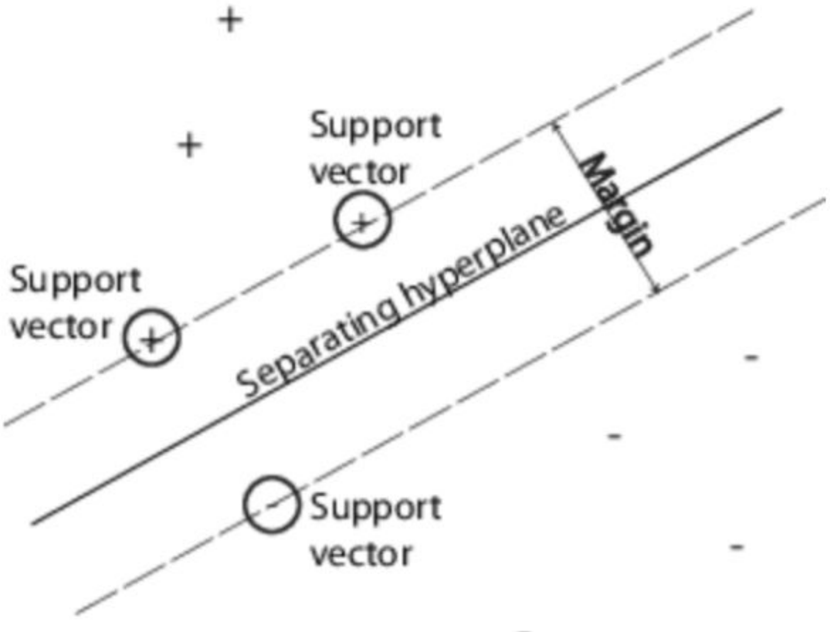
また、すべての i (i ∈ N)について $y_i(\mathrm{w}^T x_i) \geqq 0$ が成り立つ。

(1) 「点と線の距離の公式」より。



$y \in \{+1(\bullet), -1(\times)\}$

この後の流れ



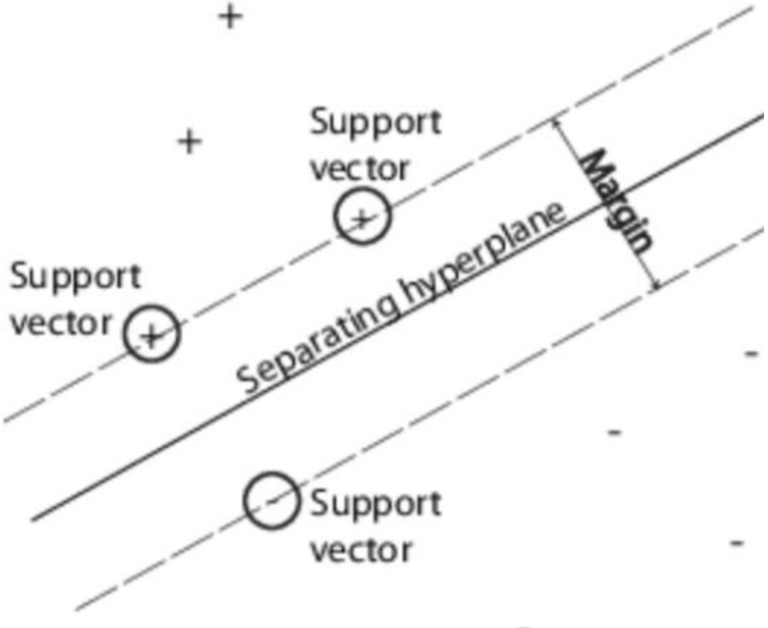
マージン最大化を目指す

マージンを最大化する前に、まず、**識別境界からの距離が最小である** x_i を求めたい。式で表すと以下のようになる。

$$\min_i \frac{y_i\left(\mathbf{w}^T x_i\right)}{\|\mathbf{w}\|}$$

このような x_i を**サポートベクトル**と呼ぶ。

この後の流れ



wは i には依存しないため、$\frac{1}{\|\mathrm{w}\|}$ を $\min_i$ の前に出す。

$$\frac{1}{\|\mathrm{w}\|}\min_i\left(\mathrm{y}_i\left(\mathrm{w}^T\mathrm{x}_i\right)\right)$$

さらに、そのような x_i について、距離（式全体）が最大になる
ようなwを見つけるには、以下の式を最適化すればよい。

$$\arg\max_{\mathrm{w}}\left\{\frac{1}{\|\mathrm{w}\|}\boxed{\min_i\left(\mathrm{y}_i\left(\mathrm{w}^T\mathrm{x}_i\right)\right)}\right\}$$

ここが最小になる i (つまり
x_i：サポートベクトル)を求める

そのx_iから識別境界がもっとも遠ざかる
（**マージンが最大になる**）ようなwを見つける

ここで、すべてのパラメータ(ベクトル w で表したすべての値)を、kwというように定数倍した w' でも最適解になる（分母子ともにwは存在するゆえ、wがk倍されても相殺される）ため、識別境界からの最小の(x_iまでの)距離が1となるように定数倍してもよい。

$$\mathrm{y}_i\left(\mathrm{w}'^T\mathrm{x}_i\right) = 1$$

$$\arg\max_{\mathrm{w}'}\left\{\frac{1}{\|\mathrm{w}'\|}\min_i\left(\mathrm{y}_i\left(\mathrm{w}'^T\mathrm{x}_i\right)\right)\right\}$$

この後の流れ

SVMの問題設定を知る

① 予測値を導く式（仮定関数）をたてる

② 最小化する問題を設定し、式（目的関数）をたてる

③ 主問題の目的関数を双対問題の目的関数に置き換える

④ 目的関数の最適解を探索的に求める(勾配法)

⑤ 得られた疎な解(サポートベクトル)を用いて推定する

この後の流れ

目的関数を立てる

$y_i\left(w'^T x_i\right)=1$ とスケーリングすると $\min_i\left(y_i\left(w'^T x_i\right)\right)$ が消え、さらに変数名 $w'=w$とおきかえると、以下を最大化する問題に帰着する。

$$\arg\max_{w} \frac{1}{\|w\|}$$

これは $\|\mathbf{w}\|^{-1}$ を最大化する問題であるが、$\|\mathbf{w}\|^2$ を最小化する問題として再定式化する。

$$\arg\min_{w} \|w\|^2$$

また、後で行う微分を容易にするため係数 1/2 をつけておく。

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

上で、$y_i\left(w'^T x_i\right)=1$ とスケーリングしたことから、すべての

i (i ∊ N) について以下の制約条件が成立する。

$$y_i\left(w^T x_i\right) \geqq 1$$

この後の流れ

ここまでで成立した目的関数と制約条件を記す。

$$\underset{\mathrm{w}}{\arg\min}\frac{1}{2}\|\mathrm{w}\|^2$$

subject to $\mathrm{y}_i\left(\mathrm{w}^T\mathrm{x}_i\right) \geqq 1$

このような線形不等式あるいは線形等式による制約条件において二次関数を最小化するような問題を二次計画問題と呼ぶ。
二次計画問題は、局所解が最適解になることが保証されており、同じく二次計画問題である**ラグランジュ未定乗数法（KKT条件つき）**を解くことによって、最適解が得られる。これは双対定理(1)に基づく問題の置き換えである。

上の目的関数と、不等式制約条件を組み合わせた *L* **(ラグランジュ関数)** を以下のように定義する。

$$\mathrm{L}(\mathrm{w},\lambda) = \frac{1}{2}\|\mathrm{w}\|^2 - \sum_{i=1}^{\mathrm{N}}\lambda_i\left\{\mathrm{y}_i\left(\mathrm{w}^T\mathrm{x}_i\right) - 1\right\}$$

式中の λ (ラムダ) は**ラグランジュ乗数**と呼ばれる。
この式中のwをλを用いた式で置き換えると、最適化すべき (主問題に対する) 双対表現が得られる（スライドp 24を参照）。主問題が目的関数の最小化を目的とする場合、双対問題ではラグランジュ関数を最大化することを目的とする。

(1) 主問題と双対問題のいずれか一方が最適解を持つなら、もう一方も最適解を持ち、主問題の最小値と双対問題の最大値は一致する。

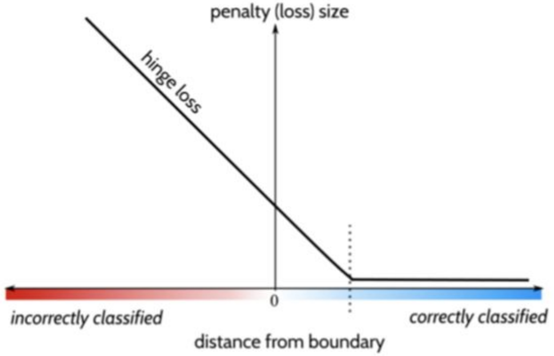
https://ja.wikipedia.org/wiki/%E5%8F%8C%E5%AF%BE%E5%95%8F%E9%A1%8C

この後の流れ

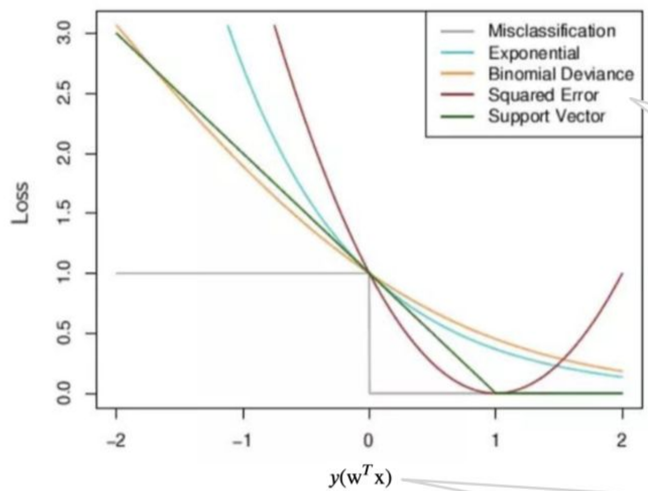
制約条件について

前出したSVMの**制約条件** $y_i\left(w^T x_i\right) \geqq 1$ は、損失関数として定式化できる。これをヒンジ損失関数という。SVMのヒンジ損失関数は、 $y_i\left(w^T x_i\right)$ が1になるまで線形に減少し、1以上では出力が0で一定になる関数。後ろに正則化項がつくこともある。

ヒンジ損失関数



損失関数の比較



グレー ：1(y*wx<0)
ブルー ：exp(-y*wx)
イエロー：log2(1+exp(-y*wx))
エンジ ：(y-wx)^2
グリーン ：max(0, 1-y*wx)

SVMのヒンジ損失関数のパラメータ

https://rohanvarma.me/Loss-Functions/

この後の流れ

$y_i\,(\mathrm{w}^T\mathrm{x}_i)$が1より大きければ0を返す

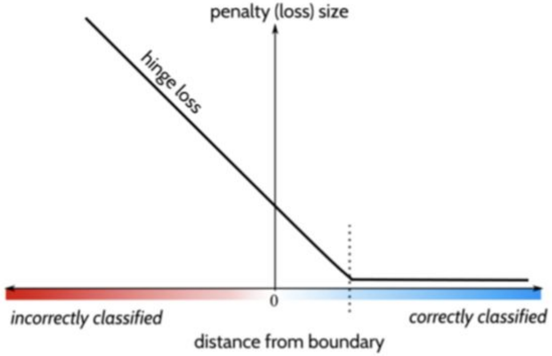
ヒンジ損失関数

$$hinge((\mathrm{w}^T x), y) = \max\{0, 1 - y(\mathrm{w}^T \mathrm{x})\}$$

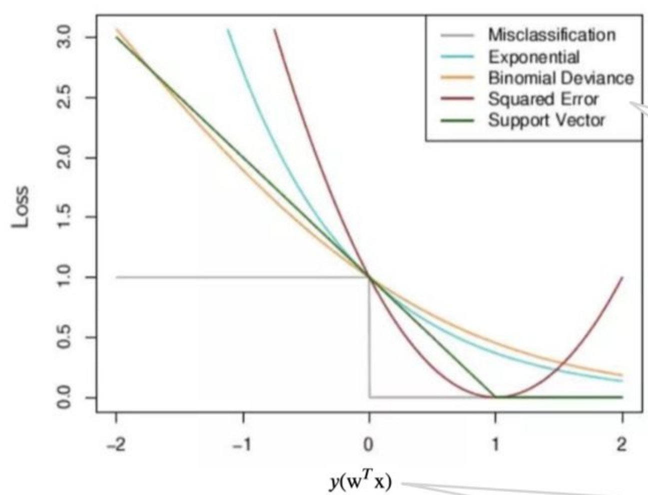
$y(\mathrm{w}^T\mathrm{x})$ が1以上の（識別境界から遠ざかる）とき、損失のペナルティを与えない。例えば、平均二乗誤差を用いると、十分に識別されているサンプルにもペナルティを与えるのに対し、ヒンジ損失関数の場合、サポートベクトルより遠く（ $y(\mathrm{w}^T\mathrm{x}) \geqq 1$）にいるサンプルには影響を与えない。一方、サポートベクトルより識別境界に近いほど大きなペナルティを課される。

https://axa.biopapyrus.jp/machine-learning/model-evaluation/loss-function.html

ヒンジ損失関数



損失関数の比較



グレー　：1(y*wx<0)
ブルー　：exp(-y*wx)
イエロー：log2(1+exp(-y*wx))
エンジ　：$(y\text{-}wx)^2$
グリーン：max(0, 1-y*wx)

SVMのヒンジ損失関数のパラメータ

https://rohanvarma.me/Loss-Functions/

この後の流れ

SVMの問題設定を知る

① 予測値を導く式（仮定関数）をたてる

② 最小化する問題を設定し、式（目的関数）をたてる

③ 主問題の目的関数を双対問題の目的関数に置き換える

④ 目的関数の最適解を探索的に求める(勾配法)

⑤ 得られた疎な解(サポートベクトル)を用いて推定する

この後の流れ

ラグランジュ未定乗数法

これは、制約条件の下で関数の極値を求める**等号制約付き最適化手法**である。具体的には、`g(x) = 0` という制約条件の下において、**`f(x)`**という目的関数を最小（あるいは最大）にするxを求める手法である。以下の式は最適化問題が満たすべき条件の一般形式である。

f(x) は最小化したいが
L(x,λ) 全体は最大化する

$$\frac{\partial \mathrm{L}}{\partial x} = 0$$
$$\frac{\partial \mathrm{L}}{\partial \lambda} = 0$$
$$\mathrm{L}(x, \lambda) = \boxed{f(x)} - \boxed{\lambda^{\mathrm{T}} g(x)}$$
$$\lambda = (\lambda_1, \ldots, \lambda_m)^{\mathrm{T}}, g(x) = (g_1(x), g_2(x), \ldots, g_m(x))^{\mathrm{T}}$$

今回は *f(x)* を最小にする問題だが、
最大化する問題の場合は
$$L(x, \lambda) = f(x) + \lambda^{\mathrm{T}} g(x)$$

ラグランジュ関数の一般形式

主問題の目的関数と制約条件を
このラグランジュ関数の形式に
落としこむ

1/2 ||w||^2 は最小化したいが
L(x,λ) 全体は最大化する

$$\mathrm{L}(\mathrm{w}, \lambda) = \underbrace{\boxed{\frac{1}{2}\|\mathrm{w}\|^2}}_{f(x)} \underbrace{\boxed{- \sum_{i=1}^{N} \lambda_i \left(y_i \left(\mathrm{w}^T x_i\right) - 1\right)}}_{-\lambda g(x)}$$

今回の**ラグランジュ関数** (p15参照)

この後の流れ

KKT条件 (カルーシュ・クーン・タッカー条件)

最適化問題を解く際に極値が満たすべき必要条件。ラグランジュ未定乗数法にKKT条件を適用し解法を求めることの利点は、ラグランジュ未定乗数法が一般化され、等号のみならず**不等号制約**も扱える点である。

SVMの目的関数は不等号制約 $y_i\left(w^T x_i\right) \geqq 1$ (p13,14参照) を持つため、**ラグランジュ未定乗数法（KKT条件つき）**にて定式化し双対問題を解く。

形式的には、**g(x) ≧ 0** という不等号制約条件にて、**f(x)**という目的関数を最小化するxを求める。

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

② $\frac{\partial L}{\partial \lambda} \leqq 0$ ← $g(x) \geqq 0$ から帰着

① $\lambda \geqq 0$

③ $\lambda^{\mathrm{T}} g(x) = 0$

$$L(x, \lambda) = f(x) - \lambda^{\mathrm{T}} g(x)$$

$$\lambda = (\lambda_1, \ldots, \lambda_m)^{\mathrm{T}}, g(x) = (g_1(x), g_2(x), \ldots, g_m(x))^{\mathrm{T}}$$

ラグランジュ関数（KKT条件つき）
の一般形式

適用 →

$f(x)$ は最小化したいが
L(x,λ) 全体は最大化する

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

KKT条件

① $\lambda_i \geqq 0$

② $(y_i\left(w^T x_i\right) - 1) \geqq 0$ ※制約条件（$g(x) \geqq 0$）

③ $\lambda_i(y_i\left(w^T x_i\right) - 1) = 0$ ※相補性条件（①②から導かれる）

$$L(w, \lambda) = \underbrace{\frac{1}{2}\|w\|^2}_{f(x)} \underbrace{- \sum_{i=1}^{N} \lambda_i \left(y_i \left(w^T x_i\right) - 1\right)}_{-\lambda g(x)}$$

今回の**ラグランジュ関数（KKT条件つき）**

この後の流れ

双対表現と勾配法の準備

前述のラグランジュ関数を式展開し、**双対表現**を得る。
さらに、双対表現をラグランジュ乗数にて微分し、後の勾配法のための**勾配**を準備する。

式展開の途中で、SVMのパラメータであるwについて式を微分して得られる $\mathrm{w} = \sum_{i=1}^{N} \lambda_i y_i x_i$ を代入し、ラグランジュ関数のパラメータをラグランジュ乗数のみとしている。

（w以外にバイアス表現(b)が含まれる場合も、同様に消去できる）

$$
\begin{aligned}
L(\mathrm{w}, \lambda) &= \underbrace{\frac{1}{2}\|\mathrm{w}\|^2}_{f(x)} \underbrace{- \sum_{i=1}^{N} \lambda_i \left(y_i \left(\mathrm{w}^T x_i\right) - 1\right)}_{-\lambda g(x)} \\
&= \frac{\mathrm{w}^T\mathrm{w}}{2} - \sum_{i=1}^{N} \lambda_i \left(y_i \left(\mathrm{w}^T x_i\right) - 1\right) \\
&= \frac{1}{2}\left(\sum_{i=1}^{N} \lambda_i y_i x_i\right)^T \left(\sum_{i=1}^{N} \lambda_i y_i x_i\right) - \sum_{i=1}^{N} \lambda_i \left\{ y_i \left| \left(\sum_{i=1}^{N} \lambda_i y_i x_i\right)^T x_i \right| - 1 \right\} \\
&= \frac{1}{2} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} \lambda_i - \sum_{i=1}^{N} \lambda_i y_i \left(\sum_{i=1}^{N} \lambda_i y_i x_i\right)^T x_i \\
&= \frac{1}{2} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} \lambda_i - \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j x_i^T x_j \\
&= \boxed{\sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j x_i^T x_j}
\end{aligned}
$$

※ラグランジュ関数をwで微分して得られる $\mathrm{w} = \sum_{i=1}^{N} \lambda_i y_i x_i$ を代入する

$$
\boxed{\frac{\partial L(\lambda)}{\partial \lambda_i} = 1 - y_i \sum_{j=1}^{N} \lambda_j y_j x_i^T x_j}
$$

上式の目的関数を λj で微分すると、こちらの**勾配**を得る。

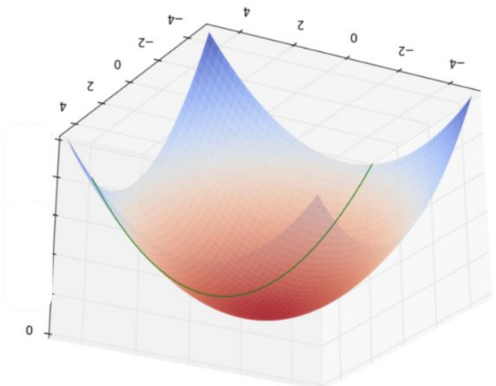
これを主問題（SVM）に対する**双対表現**とする。（後式のΣをまとめているがDiverの式と同じ目的関数）

この後の流れ

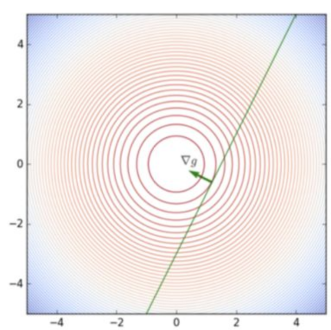
ラグランジュ関数が示していること

幾何学的関係を見て、直感的に理解してみよう。
f(x)を下図のように下に凸の3次元曲面で表したとき、**曲面上の曲線（等高線）**は f(x) = z を表現している。
さらに、g(x) = 0もまた**曲線（緑の線）**で描くことができる。

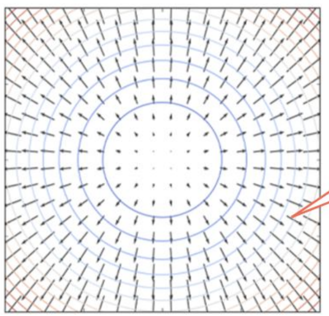
今回の最適化問題の目標は、g(x) ≧ 0 において、f(x)が最小となる点を見つけることである。そこでまず、等高線とg(x) = 0が接している点を探す。g(x) = 0（等式条件）上にそのような点があれば λ ≠ 0 となり、それがサポートベクトルである（このとき、$y(w^T x)$ =1）。一方、g(x) > 0（不等式）領域にある点は λ = 0 で、サポートベクトルではない。これはKKT条件のうち相補性条件（p17）を満たす。



f(x)とg(x)=0



g(x) = 0の法線ベクトル



等高線の法線ベクトル

等高線とg(x) = 0が接している点では、KKT条件より勾配ベクトル∇fと∇gは平行になる（今回はベクトルの向きも同じ）。
曲面の勾配ベクトル∇fとは等高線の法線ベクトルのことであり、この法線ベクトルは原点から放射線状に広がっている。

この後の流れ

SVMの問題設定を知る

① 予測値を導く式(仮定関数)をたてる

② 最小化する問題を設定し、式(目的関数)をたてる

③ 主問題の目的関数を双対問題の目的関数に置き換える

④ 目的関数の最適解を探索的に求める(勾配法)

⑤ 得られた疎な解(サポートベクトル)を用いて推定する

この後の流れ

ラグランジュ関数の最大化

双対表現（p 24を参照） をλ (ラグランジュ乗数)で微分して得た勾配で、λを更新する。

この双対表現の関数は、λに対して最大化を目指すため、λに勾配を**加算**して更新する。このように関数の最大値を探す場合は、勾配上昇法 (gradient ascent mothod) と呼ぶ。

勾配

$$\frac{\partial L(\lambda)}{\partial \lambda_i} = \boxed{1 - y_i \sum_{j=1}^{N} \lambda_j y_j x_i^T x_j}$$

$$\lambda_i^{new} = \lambda_i + \boxed{\alpha}(1 - y_i \sum_{j=1}^{N} \lambda_j y_j x_i^T x_j)$$

学習率

この後の流れ

SVMの問題設定を知る

① 予測値を導く式(過程関数)をたてる

② 最小化する問題を設定し、式(目的関数)をたてる

③ 主問題の目的関数を双対問題の目的関数に置き換える

④ 目的関数の最適解を探索的に求める(勾配法)

⑤ 得られた疎な解(サポートベクトル)を用いて推定する

この後の流れ

仮定関数

最適解を求めた結果得られた疎な解（サポートベクトル：s_n）と、仮定関数（ p10参照) $f(x) = \mathrm{sign}(\mathrm{w}^T x)$ を用いて推定を行う。

仮定関数に現れる w は $\mathrm{w} = \sum_{i=1}^{N} \lambda i y_i x_i$ によって置き換えられる。

sign関数は、$\mathrm{sign}(\mathrm{w}^T x) = \begin{cases} +1 & \text{if } (\mathrm{w}^T x) \geqq 0 \\ -1 & \text{otherwise} \end{cases}$ より、1か-1の値を返す。

$$f(x) = \mathrm{sign}(\mathrm{w}^T x)$$

$$= \mathrm{y}(\mathrm{w}^T x)$$

この等式はラグランジュ関数をwで微分すると得られる。（p18参照）

ここで w を $\mathrm{w} = \sum_{i=1}^{N} \lambda i y_i x_i$ にて置き換える

未知のデータ

サポートベクトル

推定用の仮定関数：$f(x) = \sum_{n=1}^{N} \lambda_n y_n x^T s_n$

サポートベクトルの数

この後の流れ

SVMの問題設定を知った

① 予測値を導く式（仮定関数）をたてる

② 最小化する問題を設定し、式（目的関数）をたてる

③ 主問題の目的関数を双対問題の目的関数に置き換える

④ 目的関数の最適解を探索的に求める(勾配法)

⑤ 得られた疎な解(サポートベクトル)を用いて推定する

SVM
完