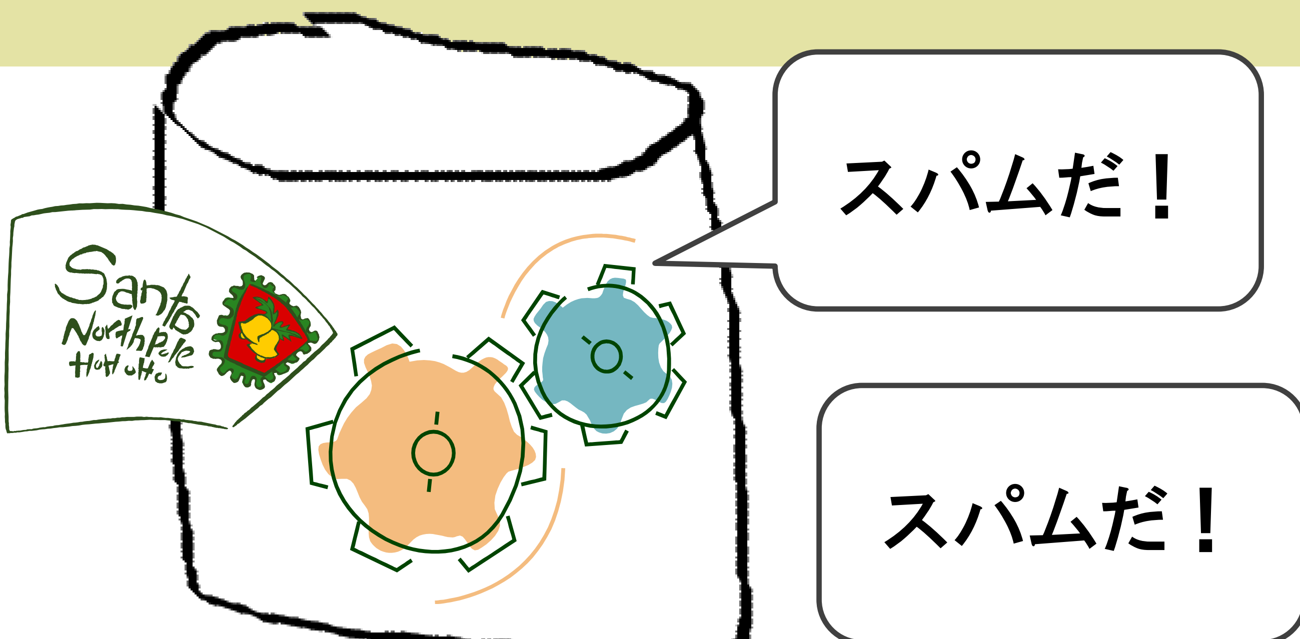
これならわかるSVM 再

挫折しない分類器入門

@D-Lec



ここにはある学生の顔写真が入っていましたが，
肖像権の関係で削除しました。

本日の見どころ

- タスクを２値分類問題として定式化する

- 分離平面
- ヒンジロス
- マージン最大化

× 学習（最適化） ☜来週，TBKくんがやってくれるはず

SVMとは…

- 「予め設定した次元数(d)」の実数値ベクトルを入力すると「正」か「負」を返す関数

- 例：

$$\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^d$$

（正例）

$$svm(\boldsymbol{x}_1) = \text{正}$$

2値分類

$$svm(\boldsymbol{x}_2) = \text{負}$$

（負例）

もっと詳しくいうと…

- 実際は，入力されたベクトルに対して，－∞~＋∞の実数値をスコアとして返す関数があって，SVMはそのスコアの符号を返す

$$svm(\boldsymbol{x}) = \begin{cases} \text{正} & score(\boldsymbol{x}) \geq 0 \\ \text{負} & score(\boldsymbol{x}) < 0 \end{cases}$$

例：

$score(\boldsymbol{x}_1) = 13.21333$ ⇨ $svm(\boldsymbol{x}_1) =$ 正

$score(\boldsymbol{x}_2) = -0.0026$ ⇨ $svm(\boldsymbol{x}_2) =$ 負

じゃあ，スコアって？

- スコア関数は以下の式で表される

重みベクトル

$$score(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$$

ベクトルの内積

$$\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^{d}$$

ここに−b（バイアス項）とか付くこともあるけど簡単のため省略.

まとめると

$$svm(\boldsymbol{x}) = \begin{cases} 正 & score(\boldsymbol{x}) \geq 0 \\ 負 & score(\boldsymbol{x}) < 0 \end{cases}$$

$$score(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x}$$

$$\boldsymbol{w}, \boldsymbol{x} \in \mathbb{R}^d$$

コラム１　ベクトルの内積(1/2)

- ２つの次元数が同じベクトルx_1, x_2が与えられたとき，x_1とx_2の内積は以下のように計算できる

$$\boldsymbol{x}_1 = \begin{bmatrix} x_{11} \\ \vdots \\ x_{1d} \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} x_{21} \\ \vdots \\ x_{2d} \end{bmatrix}$$

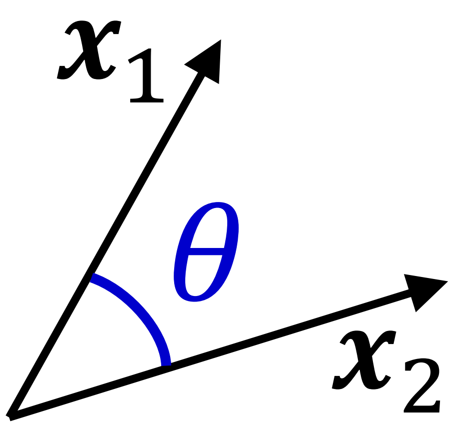
次元数: d

$$\boldsymbol{x}_1 \cdot \boldsymbol{x}_2 = (x_{11} \times x_{21}) + (x_{12} \times x_{22}) + \cdots + (x_{1d} \times x_{2d})$$

左右入れ替えても結果は同じ値

コラム１　ベクトルの内積(2/2)

- 内積は以下のようにも計算可能
  - 後で出てきます！



θ：２つのベクトルがなす角度

L2ノルム（ベクトルの長さ）

$$|\boldsymbol{x}| = \sqrt{{x_1}^2 + {x_2}^2 + \cdots + {x_d}^2} \geq \mathbf{0}$$

$$\boldsymbol{x}_1 \cdot \boldsymbol{x}_2 = |\boldsymbol{x}_1||\boldsymbol{x}_2| cos\theta$$

左右入れ替えても結果は同じ値

じゃあ，NLPでどう使うの？

事例

① <u>分類したいもの</u>(e.g., 文書，文，単語，文字列 etc…)をベクトルで表現

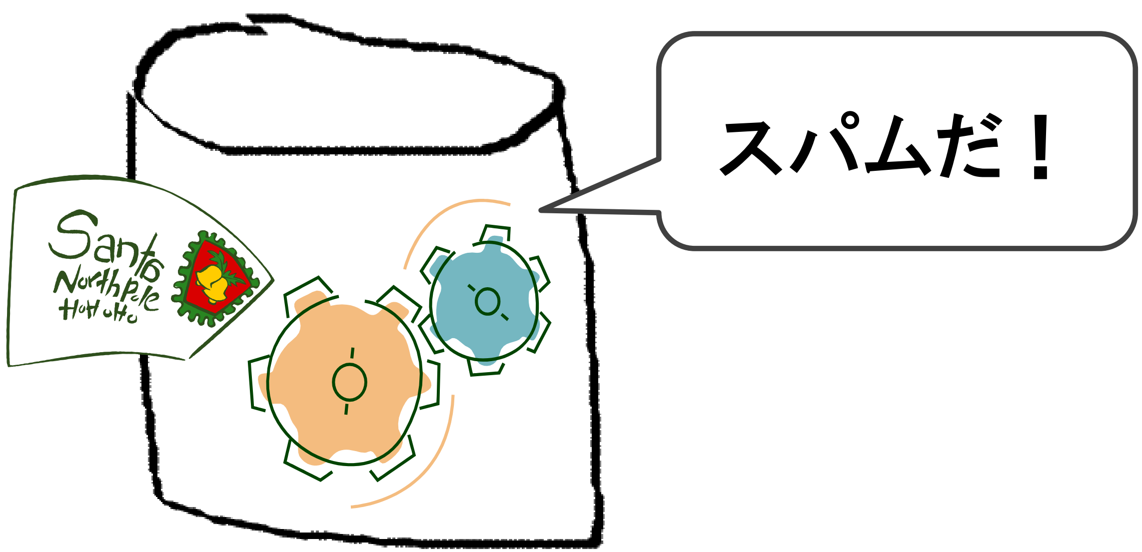
② ＳＶＭで２値分類

③ ２値分類した結果(正or負)を<u>分類先のラベル</u>に書き換え

クラスラベル

例：スパム判定

- 事例：　メール（１通）
- クラスラベル：
  - スパムだ！
  - スパムじゃない！



素性ベクトルと素性関数

重要

- 事例をベクトルで表現
  - 方法：メールをベクトルに変換する関数を作成

素性ベクトル

$$\boldsymbol{\phi}(\text{メール}) = \begin{bmatrix} \phi_1(\text{メール}) \\ \phi_2(\text{メール}) \end{bmatrix}$$

素性関数

$$\phi_1(\text{メール}) = \begin{cases} 1 & \text{メール文内に単語「儲かる」が入っている} \\ 0 & \text{入っていない} \end{cases}$$

$$\phi_2(\text{メール}) = \begin{cases} 1 & \text{メール文内に単語「}NLP\text{」が入っている} \\ 0 & \text{入っていない} \end{cases}$$

正負の値とクラスラベル

- **SVMの出力（正or負）をクラスラベルに対応付ける**
  - 正：　スパムだ！
  - 負：　スパムじゃない！

ここまでの流れを指して，
「スパム判定を2値分類問題として定式化した」
という

実際に分類してみよう

メール1

スパム

楽して儲かる
方法教えます！

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

メール2

スパムじゃない

NLP勉強会を
開催します。

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

メール3

スパム

NLPで儲かる
２０の秘訣！

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

重みベクトルwは適当に設定

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

まず、メール1

メール1

楽して儲かる
方法教えます！

スパム

$$\begin{bmatrix}1\\0\end{bmatrix}$$

$$\mathrm{score}\left(\begin{bmatrix}1\\0\end{bmatrix}\right)=\begin{bmatrix}1\\-2\end{bmatrix}\cdot\begin{bmatrix}1\\0\end{bmatrix}$$
$$=(1\times 1)+(-2\times 0)$$
$$=1\geq 0$$

よって，SVMの出力は正
つまり，スパムがちゃんと
「スパムである」と判定された！

次に、メール２

メール2

NLP勉強会を
開催します。

スパム
じゃない

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\begin{aligned} \mathrm{score}\left(\begin{bmatrix} 0 \\ 1 \end{bmatrix}\right) &= \begin{bmatrix} 1 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &= (1 \times 0) + (-2 \times 1) \\ &= -2 < 0 \end{aligned}$$

よって，SVMの出力は負
つまり，スパムでないメールがちゃんと
「スパムではない」と判定された！

最後に、メール3

メール3

NLPで儲かる
２０の秘訣！

スパム

$$\begin{bmatrix}1\\1\end{bmatrix}$$

$$\mathrm{score}\left(\begin{bmatrix}1\\1\end{bmatrix}\right)=\begin{bmatrix}1\\-2\end{bmatrix}\cdot\begin{bmatrix}1\\1\end{bmatrix}$$
$$=(1\times 1)+(-2\times 1)$$
$$=-1<0$$

判定失敗！

よって，SVMの出力は負
つまり，スパムが
「スパムではない」と判定されてしまった！

重みを変えてみる

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad \Rightarrow \quad \boldsymbol{w}' = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

$$\begin{aligned} \mathrm{score}\left(\begin{bmatrix} 1 \\ 1 \end{bmatrix}\right) &= \begin{bmatrix} 3 \\ -2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= (3 \times 1) + (-2 \times 1) \\ &= 1 \geq 0 \end{aligned}$$

判定成功！

よって，SVMの出力は正
つまり，スパムがちゃんと
「スパムである」と判定された！

これがつまり，学習

$$\boldsymbol{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \Rightarrow \boldsymbol{w}' = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$$

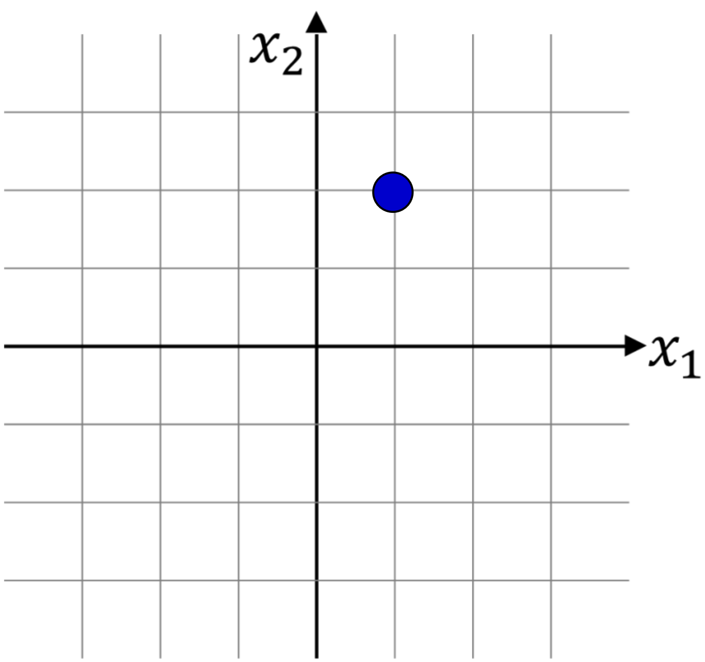
- **SVMの学習：**
  - 人手でラベル付された事例(訓練事例)が複数与えられたときに，それらを正しく分類できるようなパラメータwを求める手続き

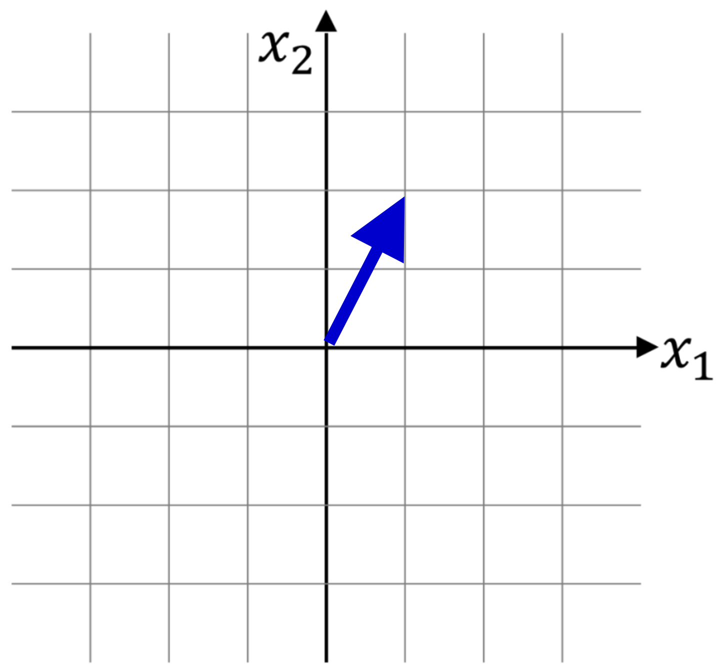
☞ 教師あり学習

- ちなみに，上のw’はメール３だけでなく，１と２も正しく分類可能

コラム２　ベクトルの表し方

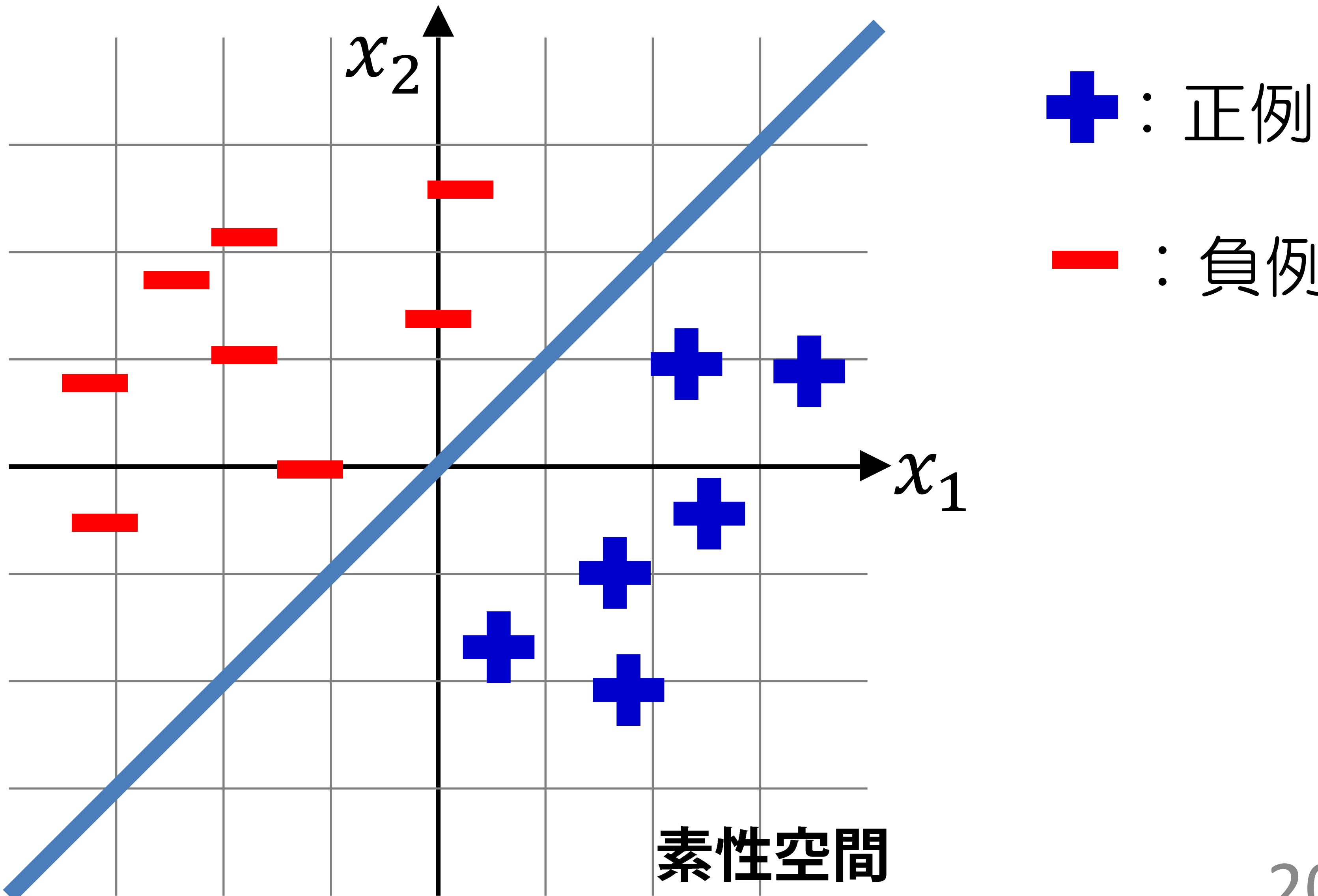
例：　$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$





分類器の話をするときは１つの図内で上記二つの表記が混在することもあるので注意

ところで，分離平面ってよく聞くよね？



分離平面の正体

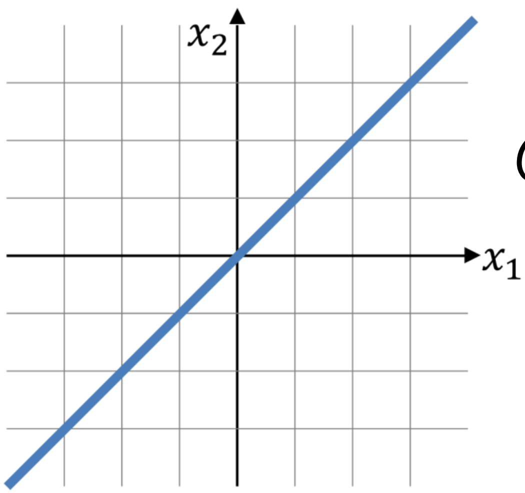
- **分離平面：**
  - 重みベクトルwがある値に定まったとき,

$$score(\boldsymbol{x}^*) = \boldsymbol{w} \cdot \boldsymbol{x}^* = 0$$

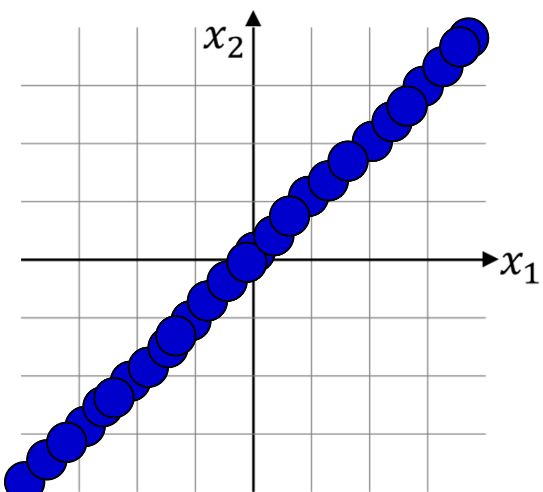
が成り立つようなベクトルx*の集合

$$\{\boldsymbol{x}^* | \boldsymbol{w} \cdot \boldsymbol{x}^* = 0,\ \boldsymbol{x}^*, \boldsymbol{w} \in \mathbb{R}^d\}$$

つまり,



の正体は



分離平面は重みベクトルと直交する

- **証明**

$\boldsymbol{w} \neq \boldsymbol{0}$ とする

固定

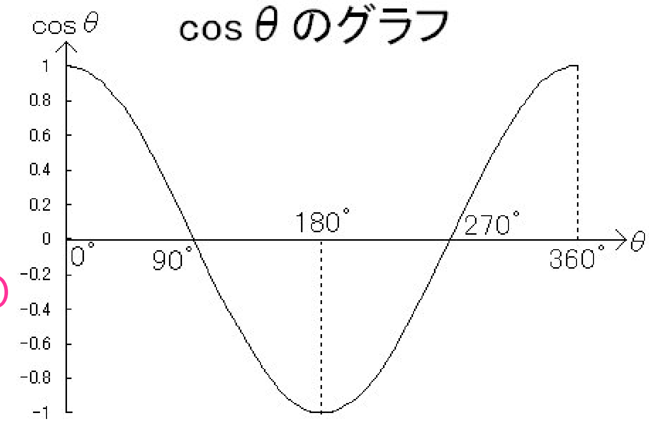
$$\boldsymbol{w} \cdot \boldsymbol{x}^* = 0$$

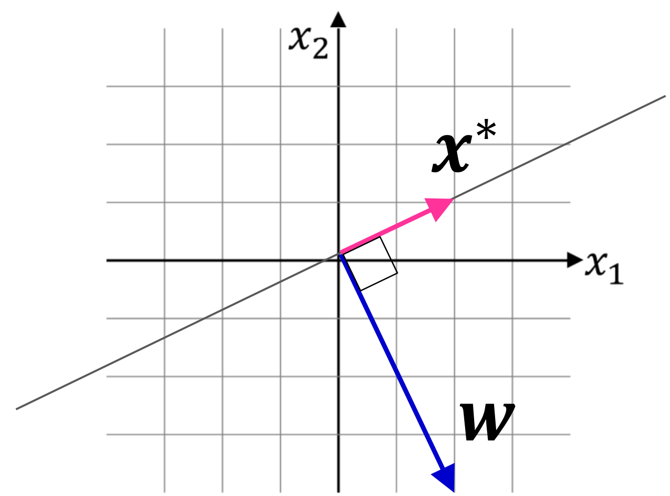
0以外の分離平面上の任意のベクトル

$$\underset{\vee \atop 0}{|\boldsymbol{w}|}\,\underset{\vee \atop 0}{|\boldsymbol{x}^*|}\,cos\theta = 0$$

$$cos\theta = 0$$

$$\theta = 90°$$





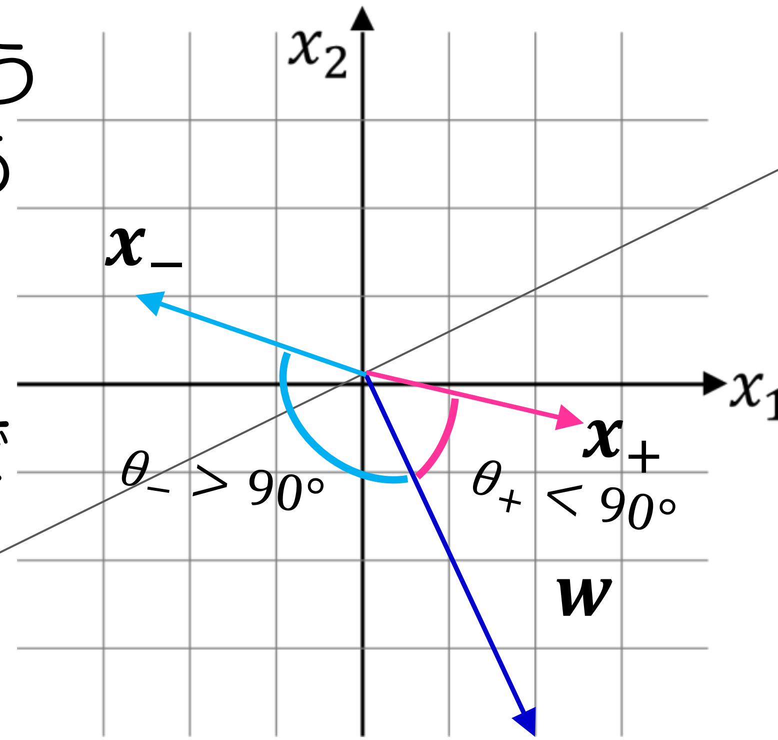
よって，wは分離平面上のすべてのベクトルに対して垂直なので，wは分離平面と直交する

実は分離平面の定義から一発

$$\left\{ \boldsymbol{x}^* \,\middle|\, \underbrace{\boldsymbol{w} \cdot \boldsymbol{x}^* = 0,}_{\text{２つのベクトルが直交する条件}} \ \boldsymbol{x}, \boldsymbol{w} \in \mathbb{R}^d \right\}$$

分離平面のw側はスコアが必ず正反対は負

- 証明：
  - wとx*がなす角が90°ということは，分離平面のw側にあるあらゆるベクトルx₊はwとなす角が90°よりも小さい
  - 反対にwと分離平面を挟んで反対にあるベクトルx₋がwとなす角は90°よりも大きい

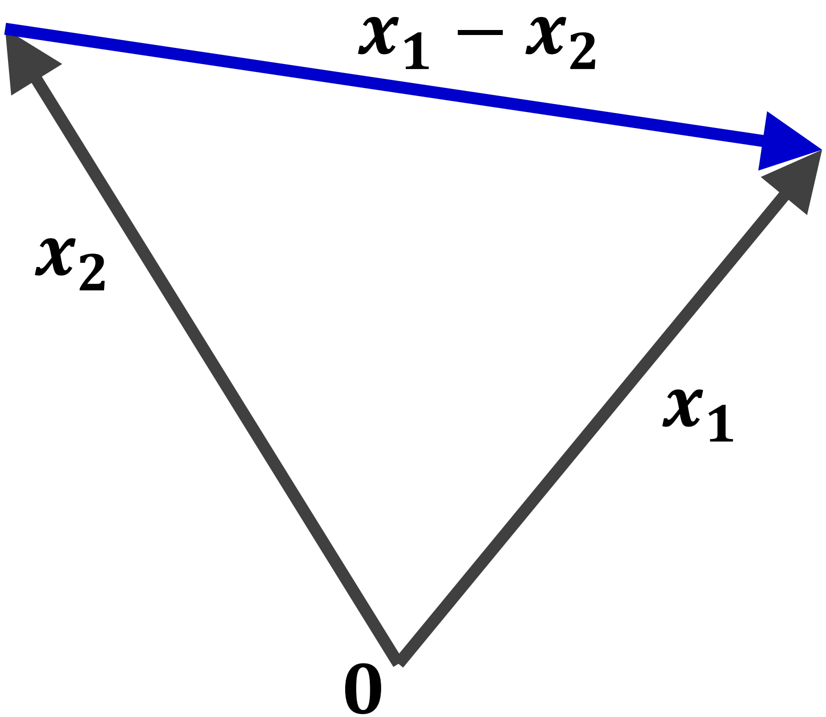


$$score(\boldsymbol{x}) = |\boldsymbol{w}||\boldsymbol{x}|cos\theta$$
$$\quad\quad\quad\quad\ \ \rotatebox{-60}{$\geq$}0 \quad \rotatebox{-60}{$\geq$}0$$

より，スコアの正負はθの値によって決まるので，表題が成り立つ

コラム３　ベクトルの差

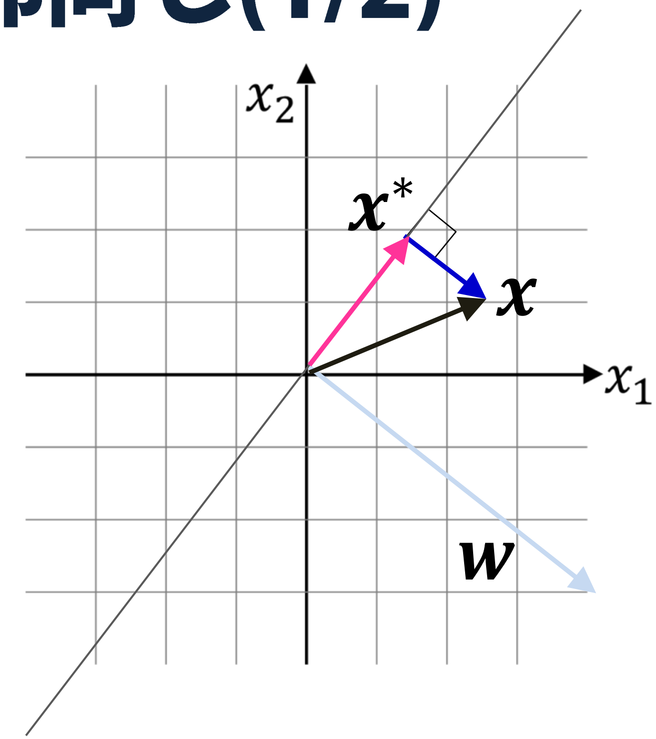
- 2つのベクトルの差x_1-x_2は以下のような図形的意味を持つ



分離平面から等距離にあるベクトルのスコアの絶対値は全部同じ(1/2)

- 証明

$$
\begin{aligned}
score(\boldsymbol{x}) &= \boldsymbol{w}\cdot\boldsymbol{x} \\
&= \boldsymbol{w}\cdot\boldsymbol{x} - \underbrace{\boldsymbol{w}\cdot\boldsymbol{x}^*}_{=0} \\
&= \boldsymbol{w}\cdot(\boldsymbol{x}-\boldsymbol{x}^*) \\
&= |\boldsymbol{w}||\boldsymbol{x}-\boldsymbol{x}^*|cos\theta
\end{aligned}
$$



ここでx^*を右のようなベクトルとする

すると，wとベクトル(x−x^*)がなす角は
0°(正)or180°(負)なので，cosθ=1 or −1

正負の区別を取るために絶対値を取ると

$$|score(\boldsymbol{x})| = |\boldsymbol{w}||\boldsymbol{x}-\boldsymbol{x}^*|$$

次ページへ→

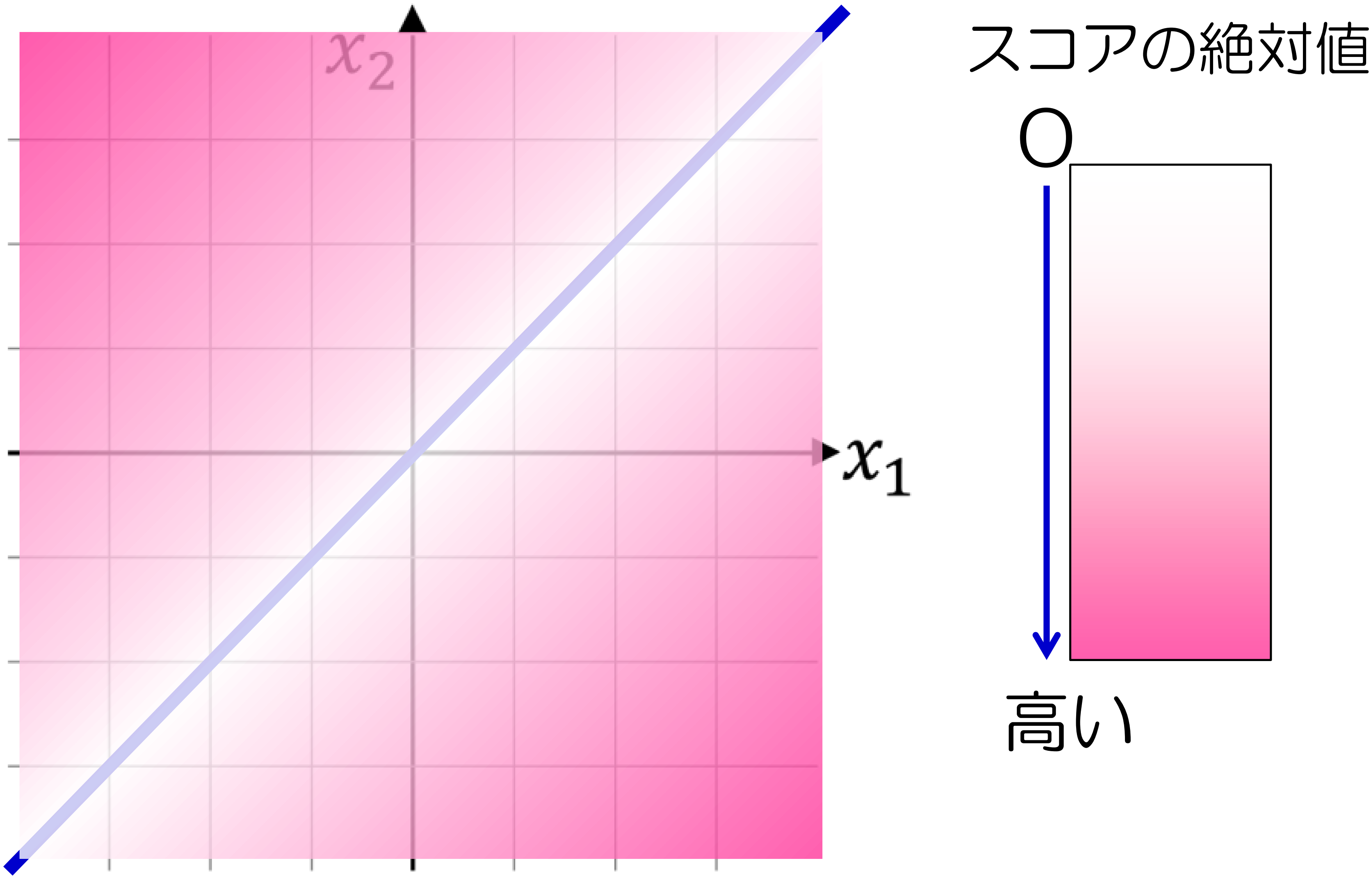
分離平面から等距離にあるベクトルのスコアの絶対値は全部同じ(2/2)

$$|score(\boldsymbol{x})| = |\boldsymbol{w}||\boldsymbol{x} - \boldsymbol{x}^*|$$

ここでwを一定とすると，scoreの大きさは$x-x^*$の長さ(=分離平面からの直線距離)によって決まる

⇒ 分離平面から遠ざかるほどスコアの絶対値は高くなる

つまり，分離平面からの直線距離が等しいベクトルのスコアの絶対値はすべて同じ


x_2
x_1
スコアの絶対値
0
高い

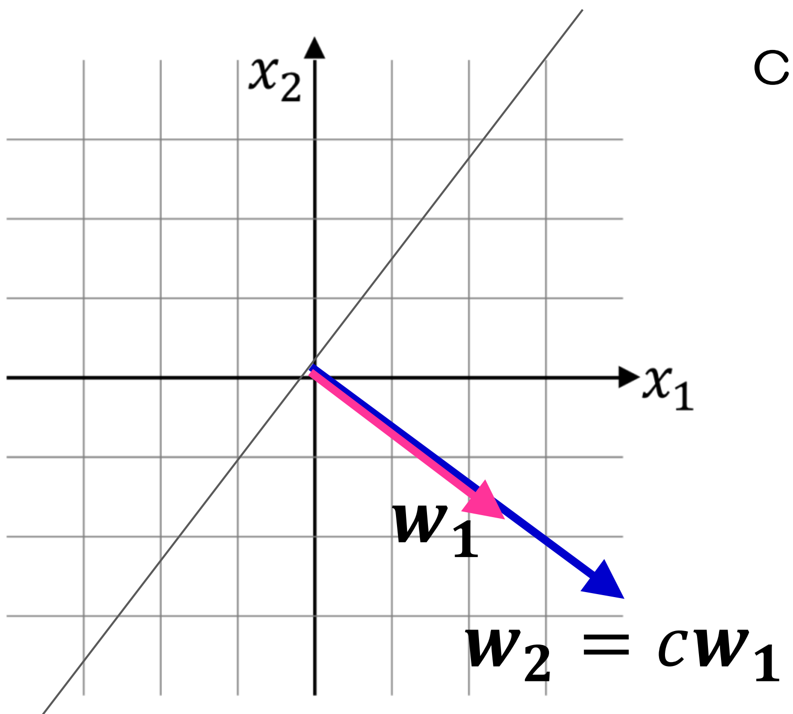
重みベクトルの大きさ|w|は分離平面の傾きとは無関係

- 証明

$\boldsymbol{w_1} \cdot \boldsymbol{x^*} = 0$ となるようなx^*を考える

$\boldsymbol{w_2} = c\boldsymbol{w_1}$ とおく（cは0よりも大きい定数）

$\boldsymbol{w_2} \cdot \boldsymbol{x^*} = c\boldsymbol{w}_1 \cdot \boldsymbol{x^*} = c(\boldsymbol{w}_1 \cdot \boldsymbol{x^*}) = 0$



c＞0より，

$\boldsymbol{w_1} \cdot \boldsymbol{x^*} = \boldsymbol{w_2} \cdot \boldsymbol{x^*} = 0$

つまり，w_1・x^*=0となるにx^*おいてはw_2・x^*=0となり，また逆も成り立つため，w_1の分離平面とw_2の分離平面は同一

じつは，ここまでの話は，
パーセプトロンやPAでも同じ話

なら，SVMは何が違うのか？

↓

最適なwを探す基準が違う！
（学習の基準）

SVMの損失関数

- SVMでは以下の値を最小にするような重みベクトルwを学習で求める

$$\max(0,\ \ \lambda - y\underbrace{\boldsymbol{w}\cdot\boldsymbol{x}}_{\text{スコア}})$$

y: 事例の正解ラベル

λ: 0以上の任意の定数
（1がよく使用される）

損失関数を読む

- **もし，事例xの分類に失敗していたら…**
  - yとw・xの符号は反対（-yw・x ≧0）

$$\Rightarrow \max(0,\ \lambda - y\boldsymbol{w}\cdot\boldsymbol{x}) = \lambda - y\boldsymbol{w}\cdot\boldsymbol{x}$$

- **もし，事例xの分類に成功していたら…**
  - yとw・xの符号は一致（-yw・x ≦ 0）
  - **もし，スコアの絶対値がλ未満なら…**
    - 0 < λ-yw・x ≦ λ

$$\Rightarrow \max(0,\ \lambda - y\boldsymbol{w}\cdot\boldsymbol{x}) = \lambda - y\boldsymbol{w}\cdot\boldsymbol{x}$$

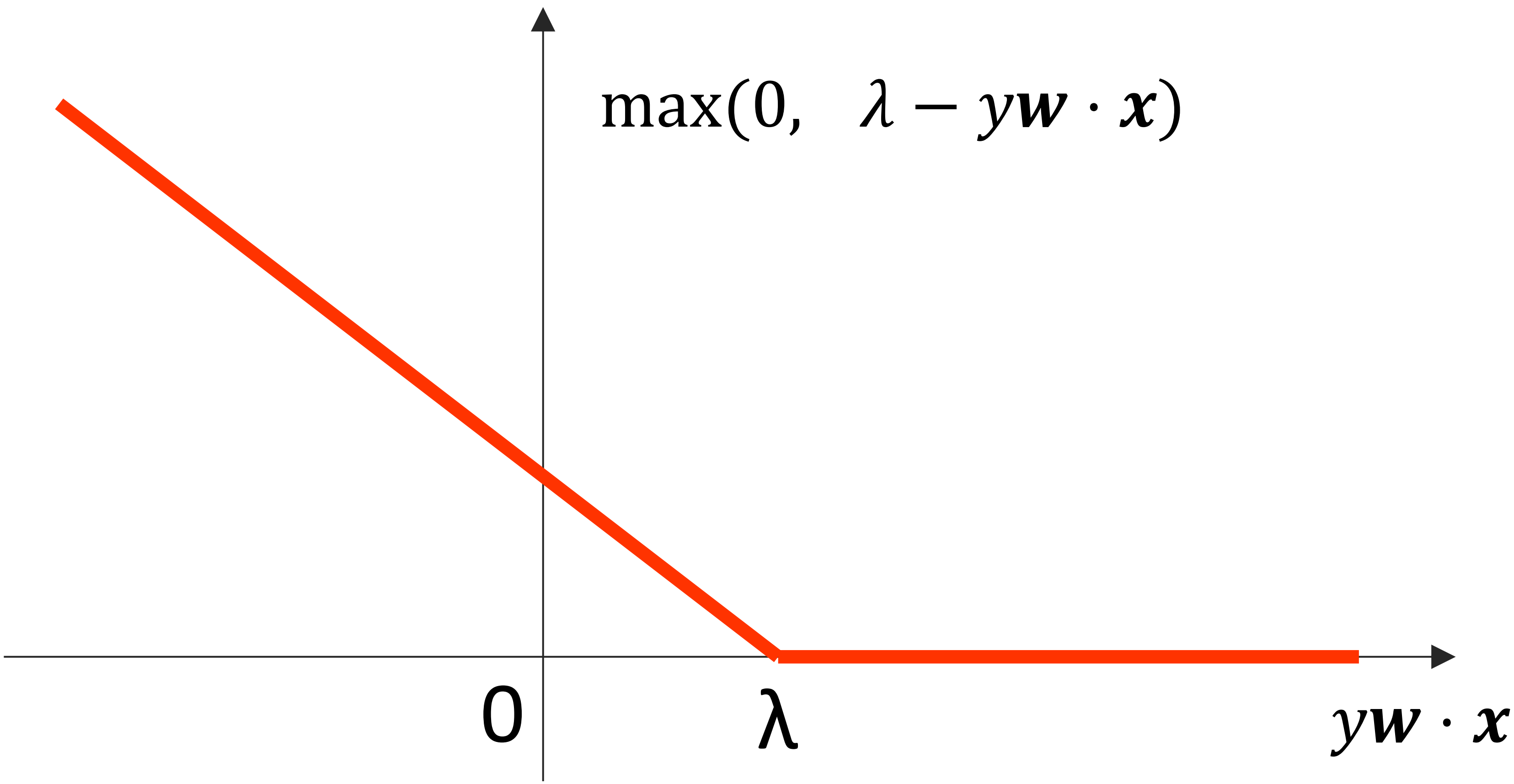
  - **もし，スコアの絶対値がλ以上なら…**
    - λ-yw・x ≦ 0

$$\Rightarrow \max(0,\ \lambda - y\boldsymbol{w}\cdot\boldsymbol{x}) = 0$$

損失関数の意味

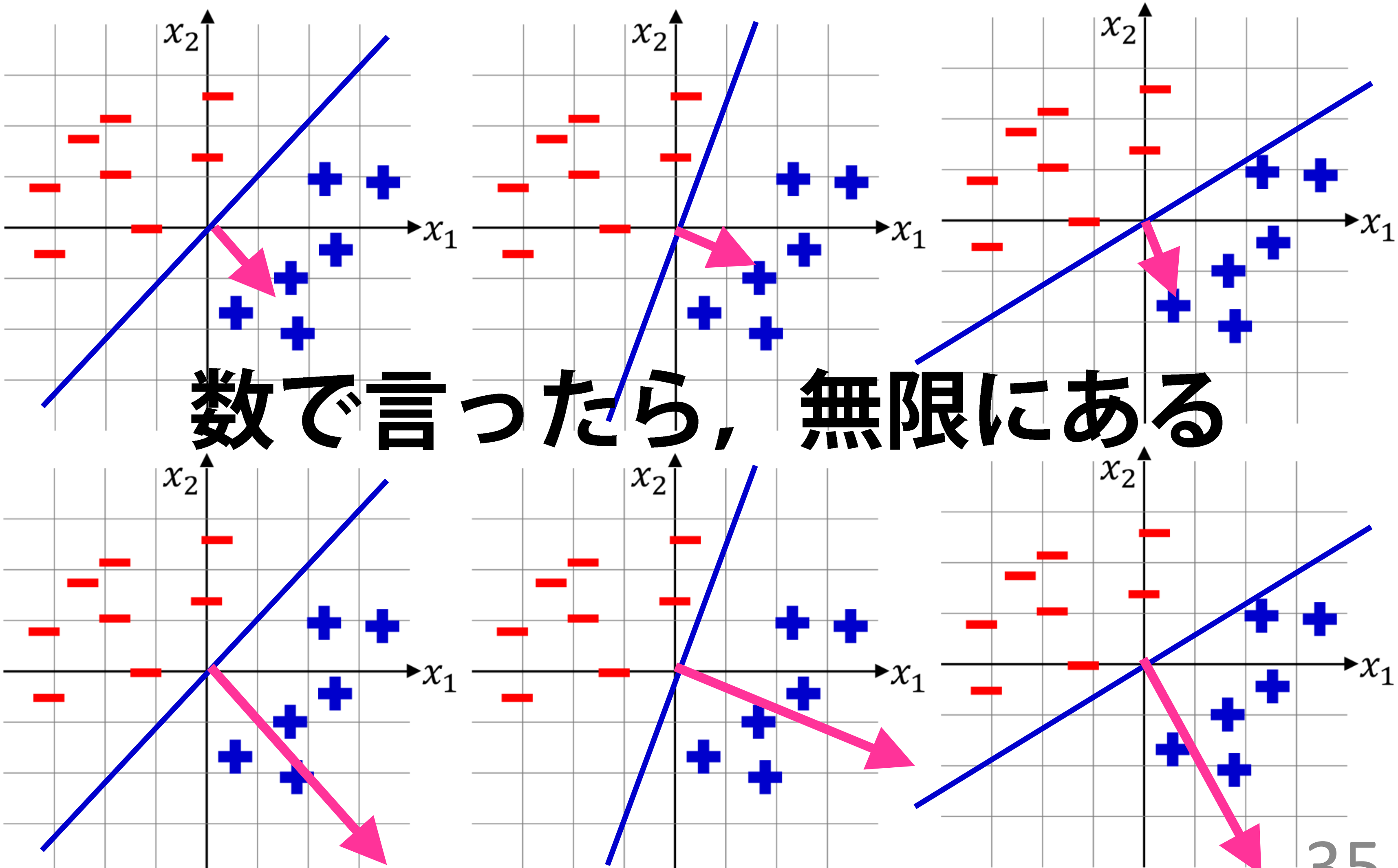
- スコアの絶対値がλ以上で分類に成功していれば損失は０だが，
- 分類に失敗していたり，
- 分類に成功していてもスコアの絶対値がλ未満ならば，
- ペナルティが付く．
- SVMはこのペナルティを最小にする重みベクトルを学習で探す．

ヒンジロス



$\max(0,\ \ \lambda - y\boldsymbol{w}\cdot\boldsymbol{x})$
0
λ
$y\boldsymbol{w}\cdot\boldsymbol{x}$

損失を最小にするwはたくさん



数で言ったら，無限にある

そういったwを集合で表すと

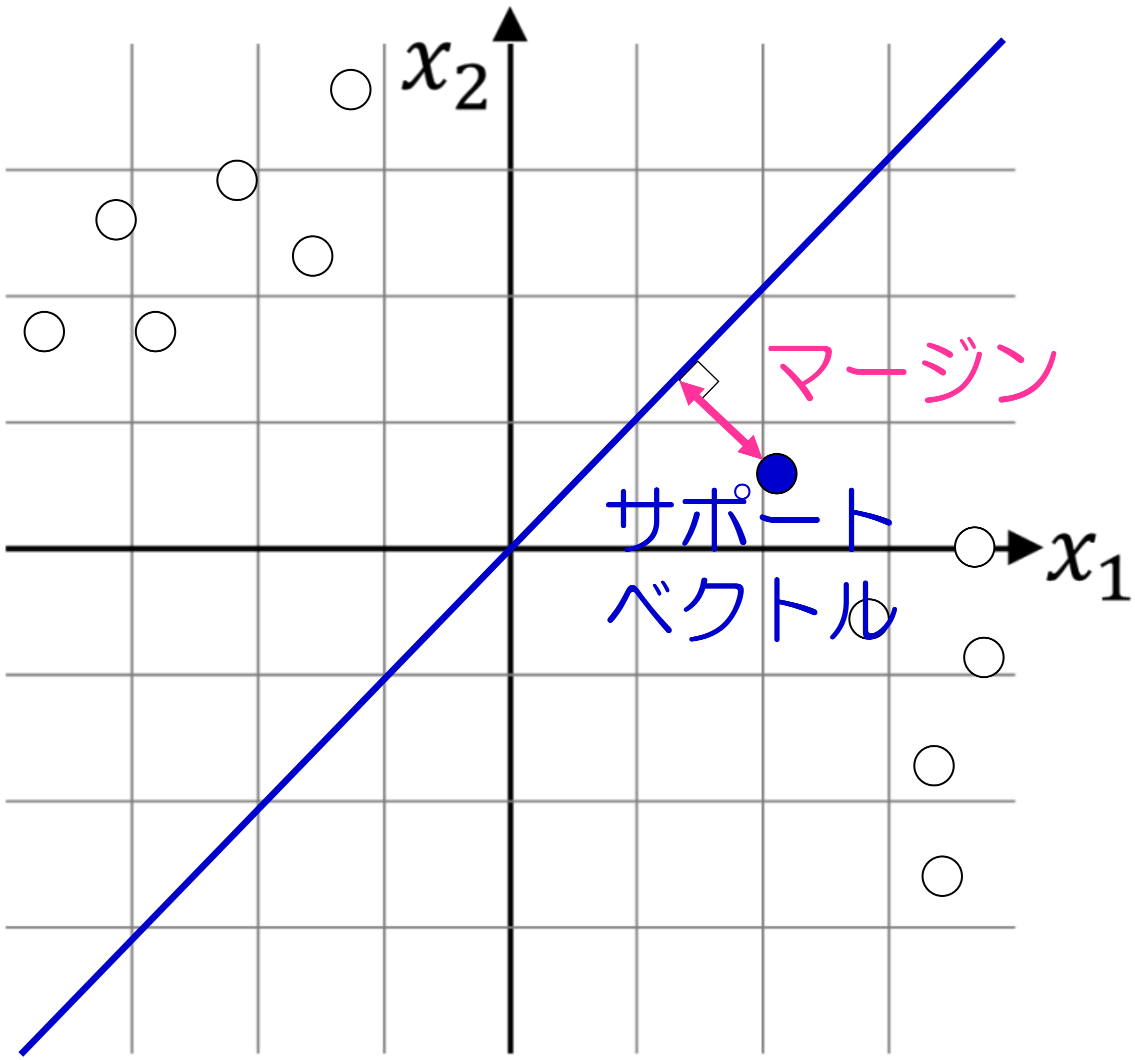
$$\{\boldsymbol{w} | y\boldsymbol{w} \cdot \boldsymbol{x}_{sv} \geq \lambda\}$$

$\boldsymbol{x}_{sv}$: $\boldsymbol{w}$が作る分離平面に最も近い事例（正負の区別なし）

サポートベクトル

この集合の中から，
一番いい重みベクトルを選びたい！

サポートベクトルとマージン



マージン

- マージン：
  - 分離平面から，分離平面にいちばん近い事例（サポートベクトル）までの距離

- **さっきのwの集合からマージンを一番大きくするような分離平面を選ぼう！**

⇩

マージン最大化

マージン最大化(1/2)

$$y\boldsymbol{w}\cdot\boldsymbol{x}_{sv} \geq \lambda$$

$$y\boldsymbol{w}\cdot\boldsymbol{x}_{sv} - \overset{=0}{y\boldsymbol{w}\cdot\boldsymbol{x}^{*}} \geq \lambda$$

$$y\boldsymbol{w}\cdot(\boldsymbol{x}_{sv} - \boldsymbol{x}^{*}) \geq \lambda$$

同符合

$$y|\boldsymbol{w}||\boldsymbol{x}_{sv} - \boldsymbol{x}^{*}|cos\theta \geq \lambda$$

$\theta = 0° or 180°$

$$|\boldsymbol{w}||\boldsymbol{x}_{sv} - \boldsymbol{x}^{*}| \geq \lambda$$

マージン最大化(2/2)

$$|\boldsymbol{w}||\boldsymbol{x}_{sv} - \boldsymbol{x}^*| \geq \lambda$$

$$|\boldsymbol{x}_{sv} - \boldsymbol{x}^*| \geq \frac{\lambda}{|\boldsymbol{w}|}$$

定数

マージン

マージン最大化の意味

$$|\boldsymbol{x}_{sv} - \boldsymbol{x}^*| \geq \frac{\lambda}{|\boldsymbol{w}|}$$

L2ノルムが小さい重みベクトルを選べばマージンが大きくなる

マージン最大化の意味

$$|\boldsymbol{x}_{sv} - \boldsymbol{x}^*| \geq \frac{\lambda}{|\boldsymbol{w}|}$$

L2ノルムが小さい重みベクトルを選べばマージンが大きくなる

ある学生

重みが小さくなるとマージンが広くなるっていうのがピンとこないです！

マージン最大化の厳密な意味

$$|\boldsymbol{x}_{sv} - \boldsymbol{x}^*| \geq \frac{\lambda}{|\boldsymbol{w}|}$$

$\{\boldsymbol{w}|y\boldsymbol{w} \cdot \boldsymbol{x}_{sv} \geq \lambda\}$ の中からL2ノルムが小さい重みベクトルを選べばマージンが大きくなる可能性がある

ベクトルwの長さが変わっても
傾きが変わらなければ,
分離平面は動かないので,
マージンも変わらない

そうしたらまたこの漢が…

それは，wの大きさがマージンの大きさと関係あることの説明になってないです！

証明しよう

まず，これ↓は自明

$W_{min} = \{\boldsymbol{w} | y\boldsymbol{w} \cdot \boldsymbol{x}_{sv} = \lambda\}$ としたとき，

$\{\boldsymbol{w} | y\boldsymbol{w} \cdot \boldsymbol{x}_{sv} \geq \lambda\} = \{c\boldsymbol{w} | \boldsymbol{w} \in W_{min}, c \geq 1\}$

自明な理由：

yw・x_{sv}≧λが成り立っているベクトルの長さをどんどん短くしていけば
必ず，yw・x_{sv}=λとなるwにたどり着く
このとき，分離平面の傾きは一切変わらない

つまり，|w|を小さくしてもマージンが変わらないことがある

でも，W_{min}を求めることで，wの候補数が絞り込める

$$W_{min} = \{\boldsymbol{w} | y\boldsymbol{w} \cdot \boldsymbol{x}_{sv} = \lambda\}$$

の中にあるwは全て異なる傾きを持っている
（傾きが同じなら，別のwの定数倍で表されるから）

そして，同じ傾きを持つwの中で
大きさが最小のものだけが入っている

この中から最小のものを探すのが
マージン最大化の本来の意図

$$y\boldsymbol{w}\cdot\boldsymbol{x}_{sv}=\lambda$$

$$|\boldsymbol{w}||\boldsymbol{x}_{sv}-\boldsymbol{x}^*|=\lambda$$

トレードオフ

より，λは定数であり，
x_{sv}およびx^*は
wの傾きで一意に決まっている

つまり，この状況でマージンを
大きくしようと思った場合，

$$W_{min}=\{\boldsymbol{w}|y\boldsymbol{w}\cdot\boldsymbol{x}_{sv}=\lambda\}$$

の中から，|w|が最小となるwを選ぶ必要がある！

ちなみに，|w|を最小化するのは，
L2正則化の効果もある．

途中で言った通り，|w|が無限に長くできるから，
できるだけ短いwを選ぶようにすることで，
候補を絞り込んでいる．

つまり，

$$y\boldsymbol{w} \cdot \boldsymbol{x}_{sv} \geq \lambda$$

の下での|w|の最小化 ⇒

- ベクトルwの長さについて，
  正則化の効果
- ベクトルwの傾きについて，
  マージン最大化の効果

を持っている

SVMの目的関数

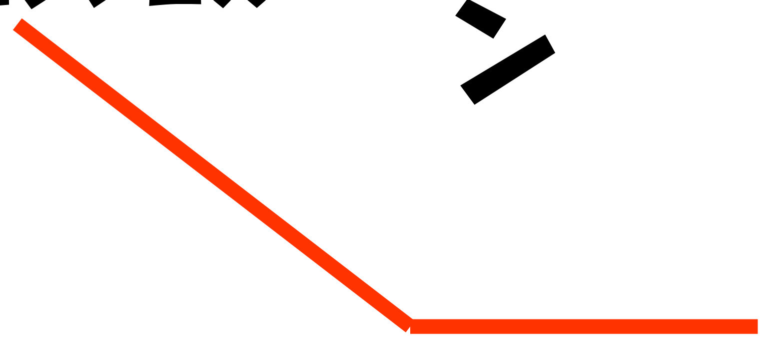
$$\max(0,\quad \lambda - y\boldsymbol{w}\cdot\boldsymbol{x}) + |\mathbf{w}|$$

を最小にするwを探すのがSVMの学習

SVMまとめ

実はほとんど
パーセプトロン

ヒンジロス



マージン最大化！
重みは最小に・・・

宿題

- スコア関数をw・xのようにすると，分離平面が必ず原点に拘束されることを証明しなさい

- スコア関数に以下のようにバイアス項を設けると，原点に拘束されず，素性空間の任意の位置に分離平面が引けるようになることを説明しなさい

$$score(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} - b$$

参考文献

学部時代，わかりやすい解説で
本当にお世話になりました.