

京都大学 情報学科 3回生向け講義「人工知能」

# 機械学習の基礎 1

## 単純ベイズ分類器とパーセプトロン



情報学研究科 教授 神田崇行  
kanda@i.kyoto-u.ac.jp

本講義資料の無断複製、無断配布を禁止します

[Original slides were created by Dan Klein and Pieter Abbeel for CS188 Intro to AI at UC Berkeley. All CS188 materials are available at <http://ai.berkeley.edu>.]

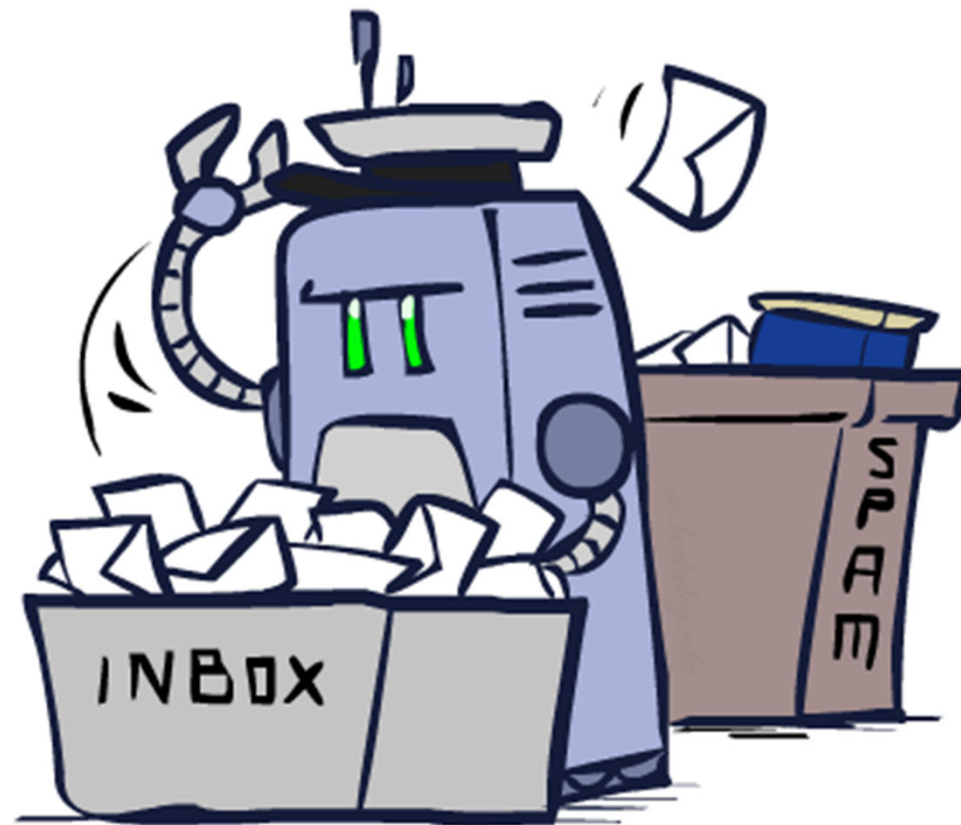
# 機械学習 (ML: Machine Learning)

---

- これまでの講義: モデルをつかって最適な意思決定を行うには？
- 機械学習: データや経験からモデルを獲得するには？
  - パラメータの学習 (e.g. 確率)
  - 構造の学習 (e.g. ベイジアンネットワークのグラフ)
  - 背後にあるコンセプトの学習  
(e.g. クラスタリング, ニューラルネットワーク)
- 今日の前半の講義: 単純ベイズ分類器によるモデルに基づく分類

# クラス分類 (Classification)

---



# 例: スパムフィルタ

- 入力: 電子メール
- 出力: スパムかどうか
- 準備:
  - 大量の電子メールの例、それぞれが“スパム”か“スパムでない”とラベル付けされたもの
  - 注意: 誰かが手作業ですべてのデータにラベル付けする必要!
  - 将来届く新しいメールのラベルが予測できるように学習したい
- 特徴量: スパムかどうかの決定に利用する属性
  - 単語: 「無料」!
  - テキストのパターン: ¥dd, CAPS
  - 本文外の情報: SenderInContacts, WidelyBroadcast
  - ...



すべての有料番組が無料で視聴できます  
WOWWOW・スカパー・スターチャンネル  
をはじめすべての有料番組が無料視聴できる  
ようになるカードの販売です  
本当なの?と半信半疑な方には「キャッ  
シュバック付きお試し版」もあります!



TO BE REMOVED FROM FUTURE  
MAILINGS, SIMPLY REPLY TO THIS  
MESSAGE AND PUT "REMOVE" IN THE  
SUBJECT.

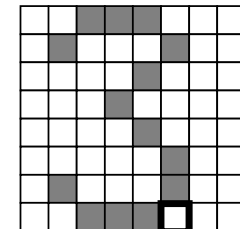
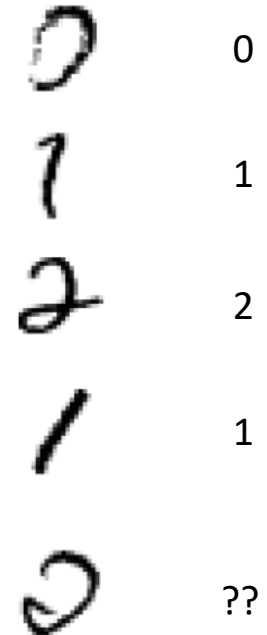
99 MILLION EMAIL ADDRESSES  
FOR ONLY \$99



水曜3限の講義「人工知能」を受講してま  
す。レポート課題の返却があったそうで  
すが、先週の講義を欠席してしまったため受  
け取れませんでした。レポートを研究室等  
まで受け取りに伺っても良いでしょうか?

# 例: 数字の認識

- 入力: 画像/ ピクセルのグリッド
- 出力: 0-9の数字
- 準備:
  - 大量の画像の例、それぞれがどの数字かラベル付けされたもの
  - 注意: 誰かが手作業ですべてのデータにラベル付けする必要!
  - 新しい画像の数字ラベルが予測できるように学習したい
- 特徴量: どの数字かの決定に利用する属性
  - ピクセル: (6,8)=ON
  - 形状のパターン: 要素数, アスペクト比, ループの数
  - ...
  - 特徴は手作業での作成よりも、データから推論されるようになってきている



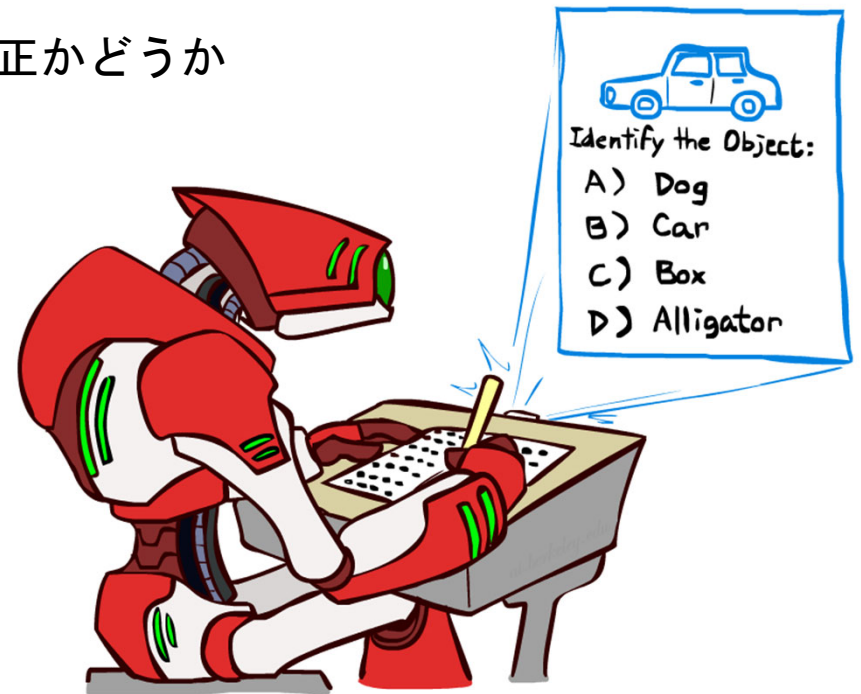
# その他の分類タスク

- 分類: 入力  $x$  が与えられたときに, ラベル(クラス)  $y$  を予測する

- 例:

- 医療診断      入力: 症状, クラス: 病名
- 不正検出      入力: アカウントの活動状況, 出力: 不正かどうか
- 小論文の自動採点    入力: 文章、出力: 成績
- カスタマーサービスに届くメールの配達
- レビューコメントの感情分析
- Language ID (言語の種類の識別)
- ... ほかにも多くのタスク

- 分類タスクは重要な商用技術！



# モデルに基づく分類 (Model-Based Classification)

---



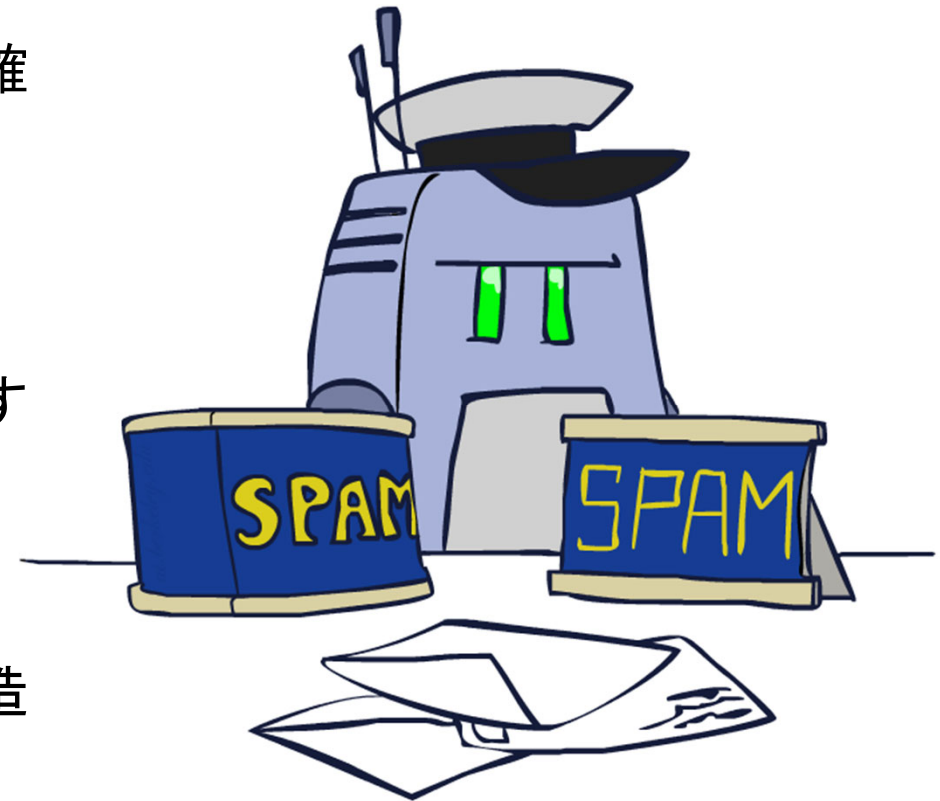
# モデルに基づく分類

## ■ モデルに基づくアプローチ

- 入力される特徴と出力ラベルがどちらも確率変数であるようなモデル（ベイジアンネットワークなど）を作成する
- 観測された特徴の変数を割り当てる
- これらの特徴の条件の下で、ラベルに関する確率分布を問い合わせる

## ■ チャレンジ

- ベイジアンネットワークはどのような構造であるべきか？
- このパラメータはどうやって学習するか？






# 数字認識の単純ベイズ分類器(Naïve Bayes)

- 単純ベイズ分類器(Naïve Bayes): 全ての特数量がラベルに対して独立に影響すると仮定

- 数字認識のシンプルなバージョン:

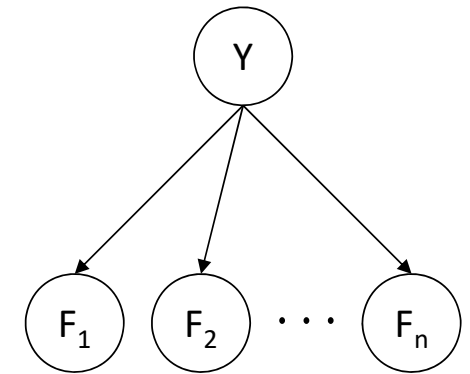
- 各グリッドの位置  $\langle i,j \rangle$  毎に1つの特徴 (変数)  $F_{ij}$
- 特数量は on / off . . . 画像中の画素の輝度が0.5以上かどうか
- 各入力画像は特徴ベクトルにマッピングされる。例えば、

  $\rightarrow \langle F_{0,0} = 0 \ F_{0,1} = 0 \ F_{0,2} = 1 \ F_{0,3} = 1 \ F_{0,4} = 0 \ \dots F_{15,15} = 0 \rangle$

- 多くの特数量、それぞれが二値変数

- 単純ベイズモデル:  $P(Y|F_{0,0} \dots F_{15,15}) \propto P(Y) \prod_{i,j} P(F_{i,j}|Y)$

- 何を学習する必要があるか？



# 一般的な単純ベイズ分類器(Naïve Bayes)

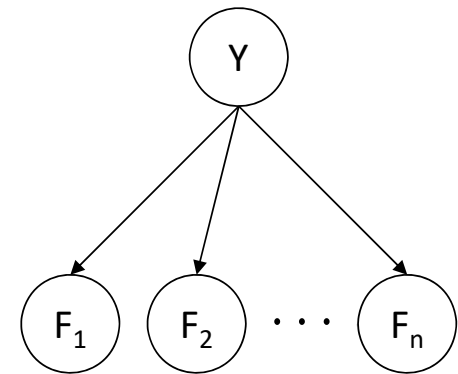
- 一般的な単純ベイズモデルは:

$|Y|$  parameters

$$P(Y, F_1 \dots F_n) = P(Y) \prod_i P(F_i|Y)$$

$|Y| \times |F|^n$  values

$n \times |F| \times |Y|$   
parameters



- 各特徴量がどのようにクラスに依存するか、のみを記述
- パラメータの数は  $n$  に **比例する**
- モデルは非常にシンプルだが、たいていはこれで上手く行く

# 単純ベイズ分類器における推論

- 目標: ラベル変数  $Y$  の事後確率を計算

- ステップ1: 各ラベルに対して、ラベルと証拠の同時確率を計算

$$P(Y, f_1 \dots f_n) = \begin{bmatrix} P(y_1, f_1 \dots f_n) \\ P(y_2, f_1 \dots f_n) \\ \vdots \\ P(y_k, f_1 \dots f_n) \end{bmatrix} \Rightarrow \begin{bmatrix} P(y_1) \prod_i P(f_i|y_1) \\ P(y_2) \prod_i P(f_i|y_2) \\ \vdots \\ P(y_k) \prod_i P(f_i|y_k) \end{bmatrix}$$

$\xrightarrow{+}$

- ステップ2: 合計して証拠に関する確率を得る

$$\begin{array}{c} \downarrow \\ P(f_1 \dots f_n) \end{array}$$
$$P(Y|f_1 \dots f_n)$$

- ステップ3: ステップ1 を ステップ2 で割って正規化する

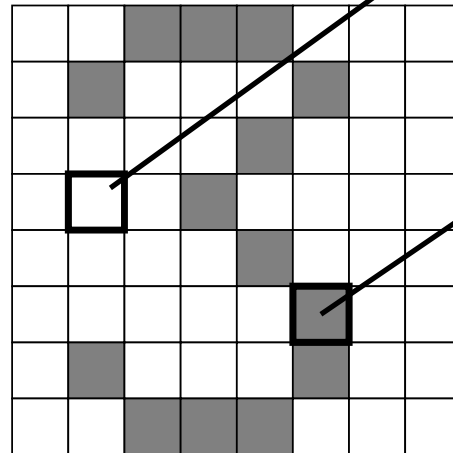
# 一般的な単純ベイズ分類器

- 単純ベイズ分類器を使うために何が必要か？
  - 推論方法 (前のスライド)
    - 大きな確率表から開始:  $P(Y)$  と  $P(F_i|Y)$  の表
    - 標準的な推論方法で  $P(Y|F_1 \dots F_n)$  を計算
    - それ以外の新しい方法は不要
  - 局所的な条件付き確率表の推定
    - $P(Y)$  ラベルに関する事前確率
    - 各特徴量に対して  $P(F_i|Y)$  (証拠変数)
    - これらの確率を併せて、モデルの**パラメータ**と呼び、 **$\theta$** と記述する
    - 現時点までは、何らかの不思議な方法によって、これらの値は既知である、とみなしていたが・・・
    - ・・・・実際には、訓練データをカウントして得られることが多い  
(まもなく、この方法も扱う)

# 例: 条件付き確率

$P(Y)$

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
0	0.1



$P(F_{3,1} = on|Y)$   $P(F_{5,5} = on|Y)$

1	0.01
2	0.05
3	0.05
4	0.30
5	0.80
6	0.90
7	0.05
8	0.60
9	0.50
0	0.80

1	0.05
2	0.01
3	0.90
4	0.80
5	0.90
6	0.90
7	0.25
8	0.85
9	0.60
0	0.80

# テキストの単純ベイズ分類器

## ■ Bag-of-words の単純ベイズ分類器:

- 特徴量:  $i$  番目の位置の単語  $W_i$
- 特徴量変数の条件のもとでラベルを予測 (スパムかどうか)
- 各特徴量はラベルのもとで条件付き独立
- 新しい点: 各  $W_i$  は同一分布

水曜 3限 の 講義 「人工知能」 を  
受講 してます。 . . .

↑  
 $i$  番目  $W_i$

## ■ 生成モデル: $P(Y, W_1 \dots W_n) = P(Y) \prod_i P(W_i | Y)$

文中の  $i$  番目の位置の  
単語、辞書中の  $i$  番目  
の単語ではない!

## ■ 互いに結びついた分布と bag-of-words

- 通常ならば、各変数はそれぞれの条件付き確率  $P(W_i | Y)$  に従う
- bag-of-words モデルでは
  - 各位置において同一分布
  - 全ての位置は同一の条件付き確率  $P(W | Y)$  を共有する
  - なぜこのような仮定をするか?
- “bag-of-words” と呼ばれるのは、モデルが単語(word)の順番や並べ替えに影響されないから (結局、出現頻度・回数のみを考慮している)

# 例: スパムフィルタ

- モデル:  $P(Y, W_1 \dots W_n) = P(Y) \prod_i P(W_i|Y)$
- パラメータは？

$P(Y)$

ham : 0.66
spam: 0.33

$P(W|\text{spam})$

the : 0.0156
to : 0.0153
and : 0.0115
of : 0.0095
you : 0.0093
a : 0.0086
with: 0.0080
from: 0.0075
...

$P(W|\text{ham})$

the : 0.0210
to : 0.0133
of : 0.0119
2020: 0.0110
with: 0.0108
from: 0.0107
and : 0.0105
a : 0.0100
...

- これらの確率表はどこから得られたか？

# スパムの例

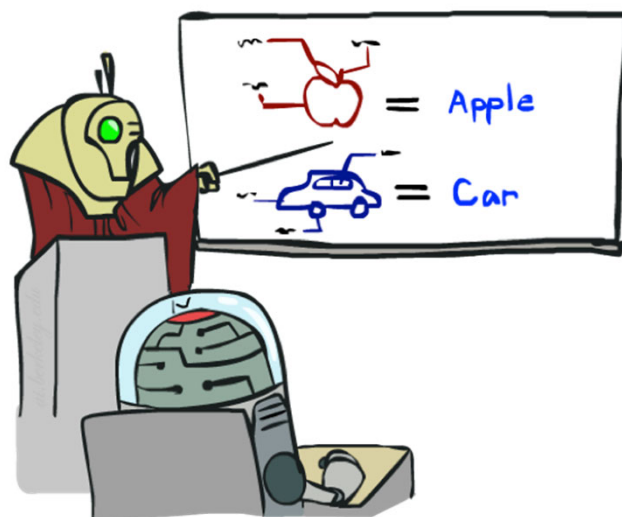
Word	P(w spam)	P(w ham)	Tot Spam	Tot Ham
(prior)	0.33333	0.66666	-1.1	-0.4

---

$$P(\text{spam} | w) = 98.9$$



# 訓練とテスト

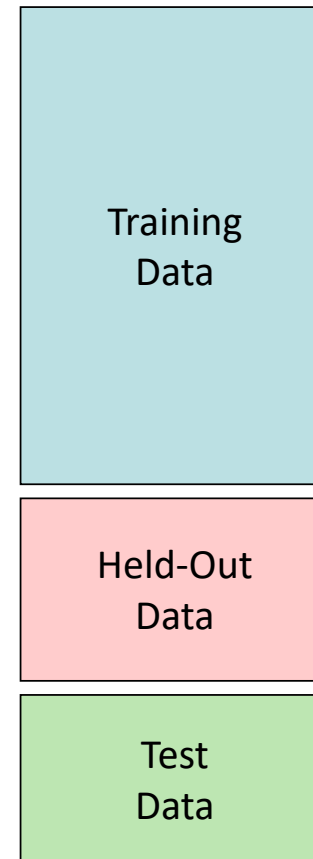


# 経験的なリスク最小化

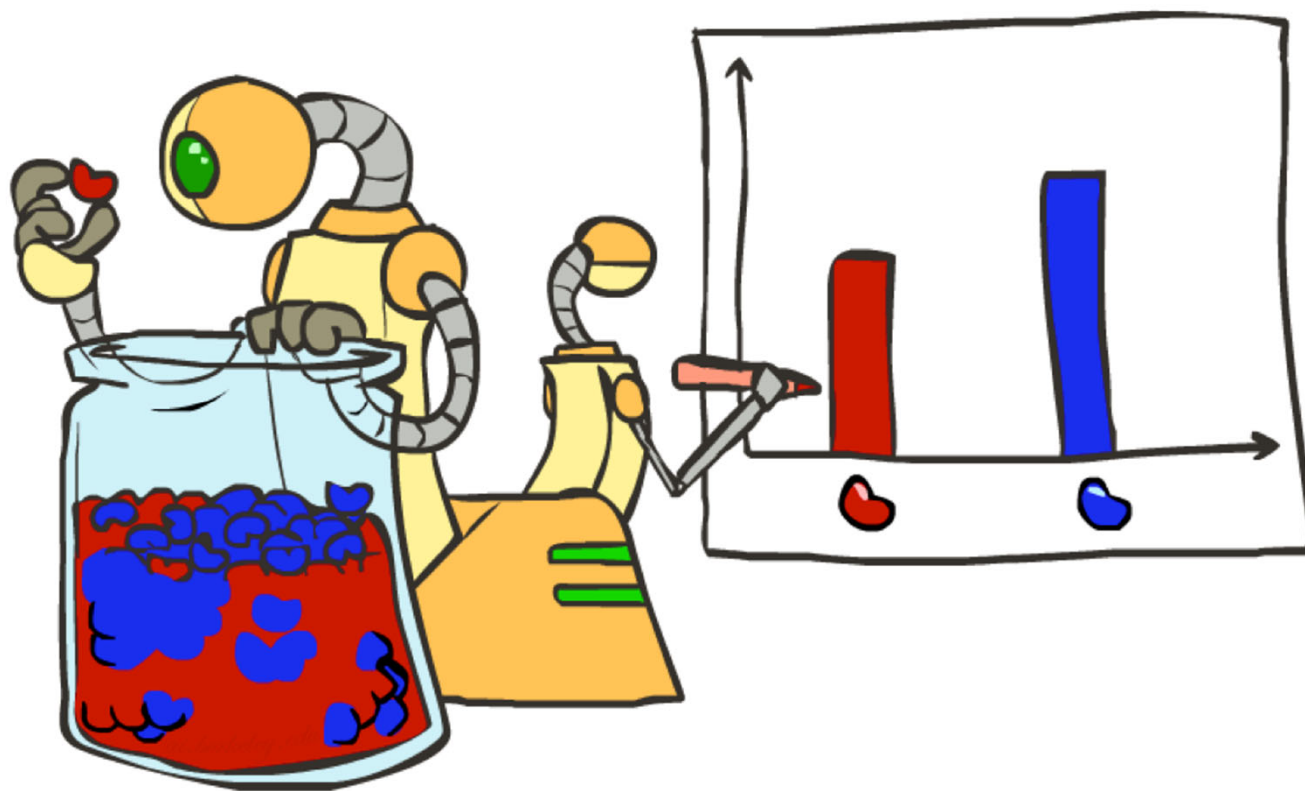
- 経験的なリスク最小化 (Empirical risk minimization)
  - 機械学習における基本原則
  - テストセットの真の分布において最良の性能のモデル（分類器）が欲しい
  - 真の分布は未知なので、実際の訓練セットに対する最良のモデルを選ぶ
  - 訓練セットにおいて最良のモデルを見出すのは最適化問題と言える
- 懸念すべき点：  
訓練セットへのオーバーフィッティング(過剰適合, overfitting)
  - 訓練セットをより大きくする  
(サンプリングの偏りがより少なくなり、訓練セットがテストセットにより似る)
  - 仮説の複雑さを制限する (正則化(regularization)や小さな仮説空間)

# 重要なコンセプト

- データ: ラベル付きインスタンス
  - 訓練セット (Training set)
  - ホールドアウトセット (Held out set)
  - テストセット (Test set)
- 特徴量: 各データ  $x$  を特徴づける属性値 (の集合)
- 実験のサイクル
  - 訓練セットによるパラメータの学習 (モデルの確率など)
  - (ホールドアウトセットによりハイパーパラメーターを調整)
  - テストセットによる正確さの計算
  - 非常に重要: 決してテストセットで「ピーク」にしない!
- 評価 (様々な指標がある)
  - 正解率(Accuracy): 正しく分類されたインスタンスの割合
- 一般化とオーバーフィッティング
  - テストデータでの分類性能を良くしたい
  - オーバーフィット: 訓練データによく適合するが一般化できない
  - 一般化とオーバーフィッティング (これから説明)



# パラメーター推定




# パラメータ推定

ランダム変数の分布を推定する

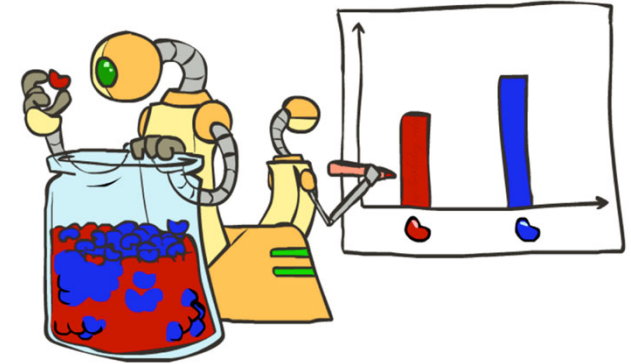
- 導出する: 人に聞く (なぜこれが難しいか?)
- 経験的に: 訓練データを利用 (学習!)
  - 結果  $x$  それぞれの値をとる **経験的比率** を数える

$$P_{\text{ML}}(x) = \frac{\text{count}(x)}{\text{total samples}}$$

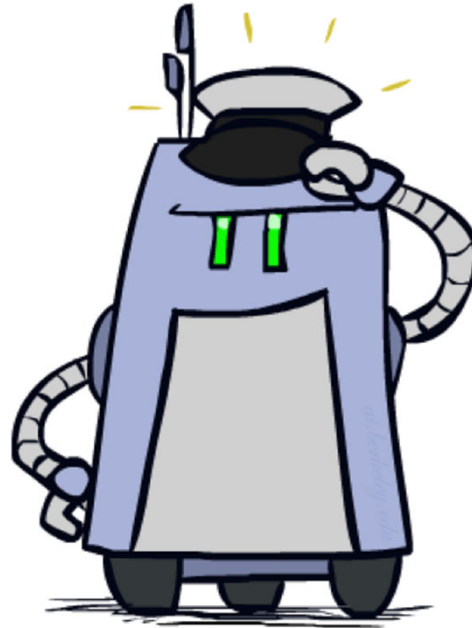
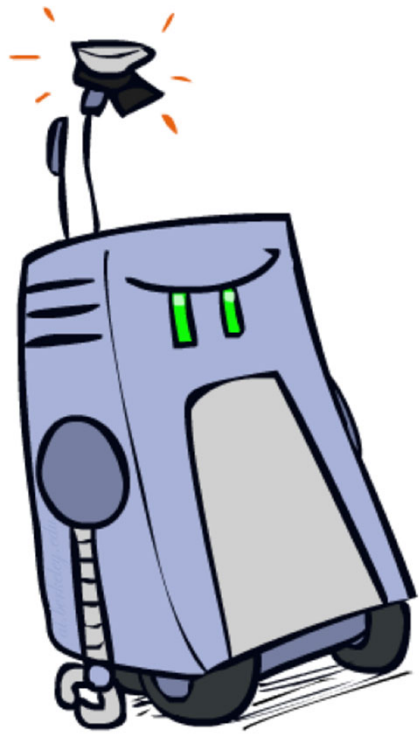

$$P_{\text{ML}}(\textcolor{red}{r}) = 2/3$$

= 最尤推定 (Maximum Likelihood) ( **このデータの尤度** を最大化する方法)

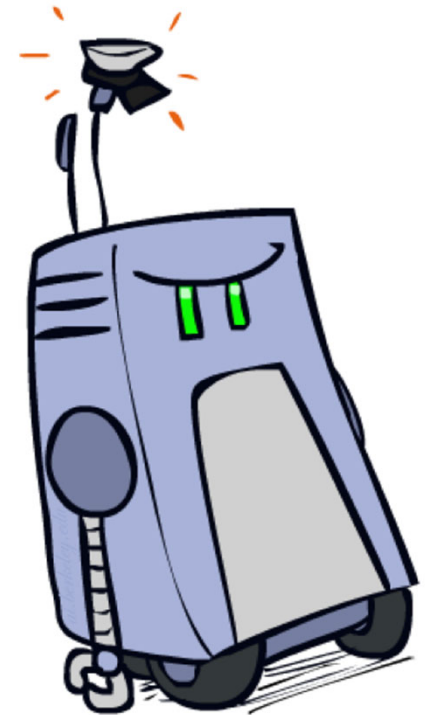
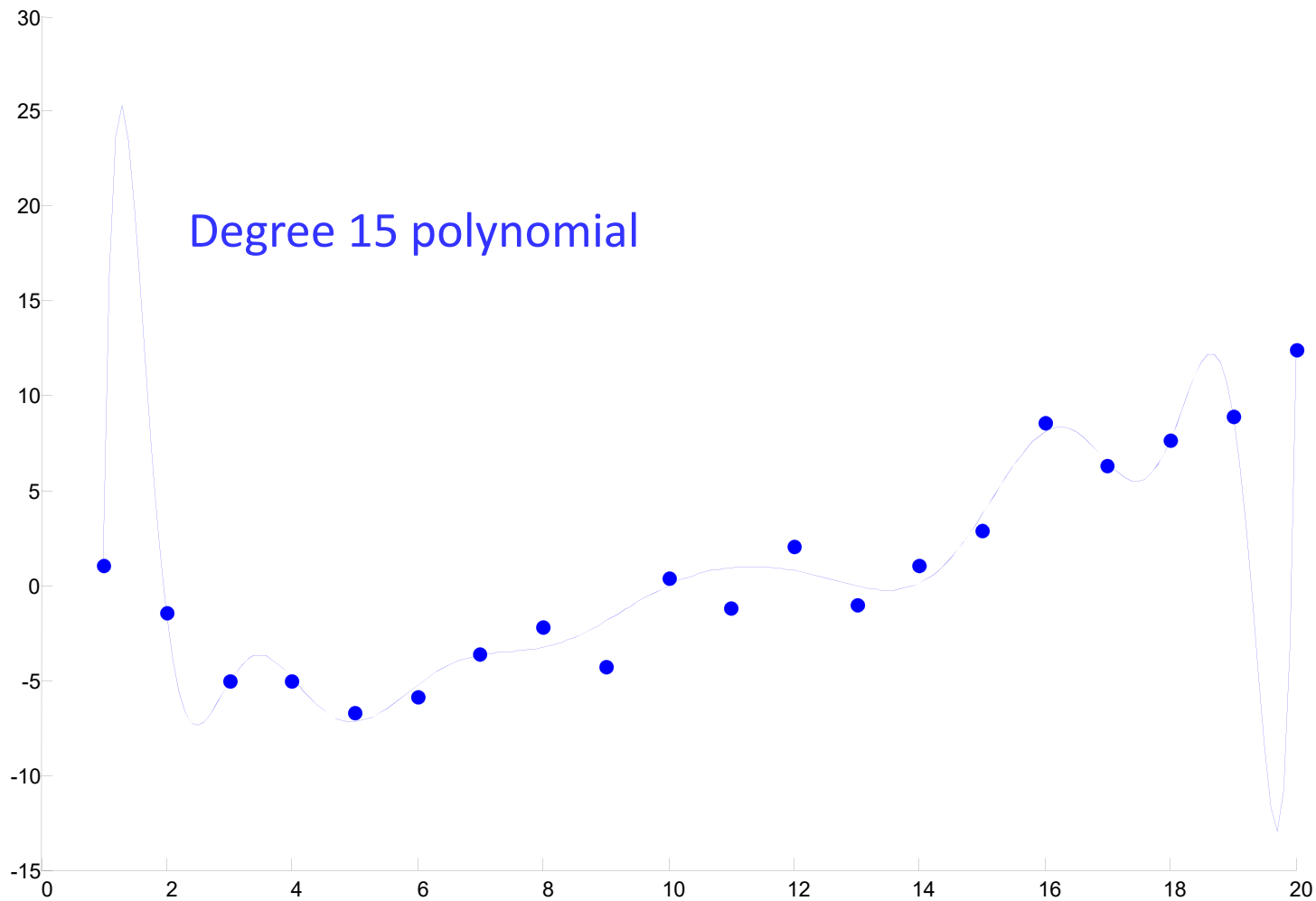
$$\theta_{\text{ML}} = \arg \max_{\theta} P(\mathbf{X}|\theta)$$



# 一般化とオーバーフィッティング

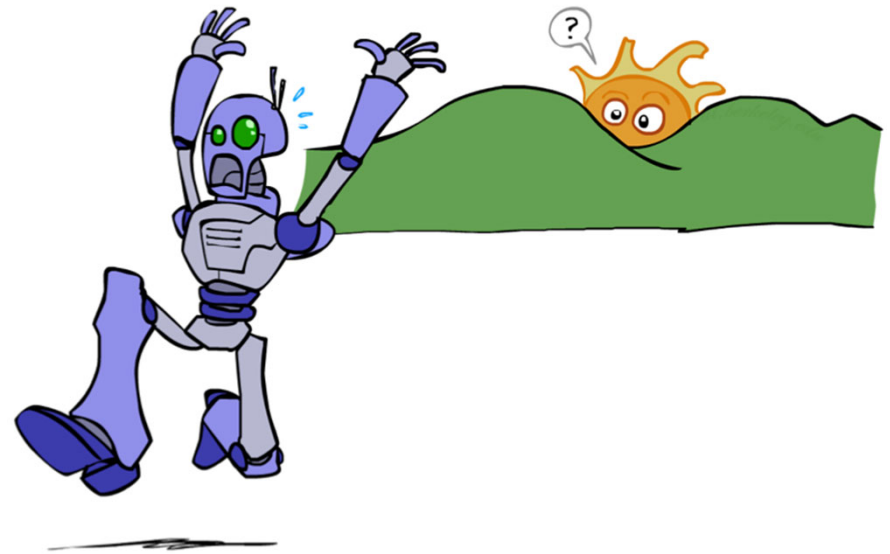
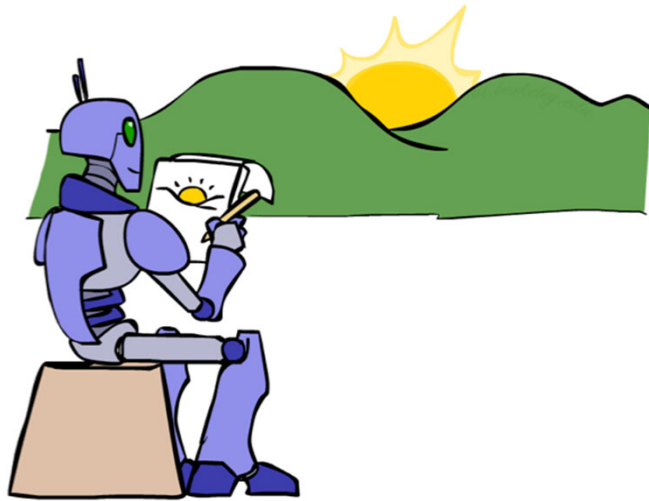


# オーバーフィッティング(Overfitting)



# 未知のイベント (Unseen Events)

---





# 例: オーバーフィッティング

$P(\text{features}, C = 2)$

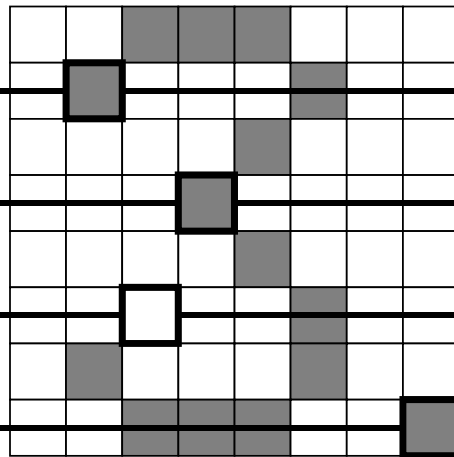
$$P(C = 2) = 0.1$$

$$P(\text{on}|C = 2) = 0.8$$

$$P(\text{on}|C = 2) = 0.1$$

$$P(\text{off}|C = 2) = 0.1$$

$$P(\text{on}|C = 2) = 0.01$$



$P(\text{features}, C = 3)$

$$P(C = 3) = 0.1$$

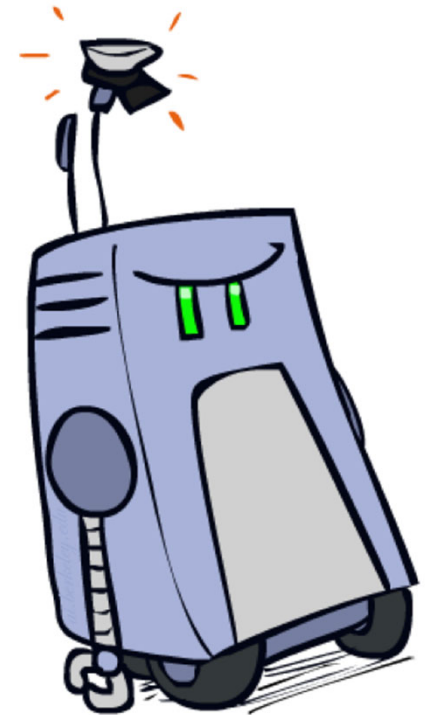
$$P(\text{on}|C = 3) = 0.8$$

$$P(\text{on}|C = 3) = 0.9$$

$$P(\text{off}|C = 3) = 0.7$$

$$P(\text{on}|C = 3) = 0.0$$

*2 wins!!*



# 例: オーバーフィッティング

- 確率の相対比較（オッズ比）による事後確率:
  - オッズ比が高い単語はどのような単語か？

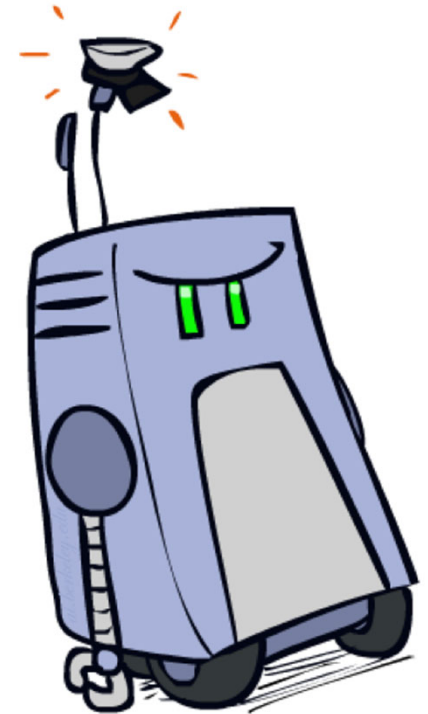
$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

south-west	:	inf
nation	:	inf
morally	:	inf
nicely	:	inf
extent	:	inf
seriously	:	inf
...		

$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

screens	:	inf
minute	:	inf
guaranteed	:	inf
\$205.00	:	inf
delivery	:	inf
signature	:	inf
...		

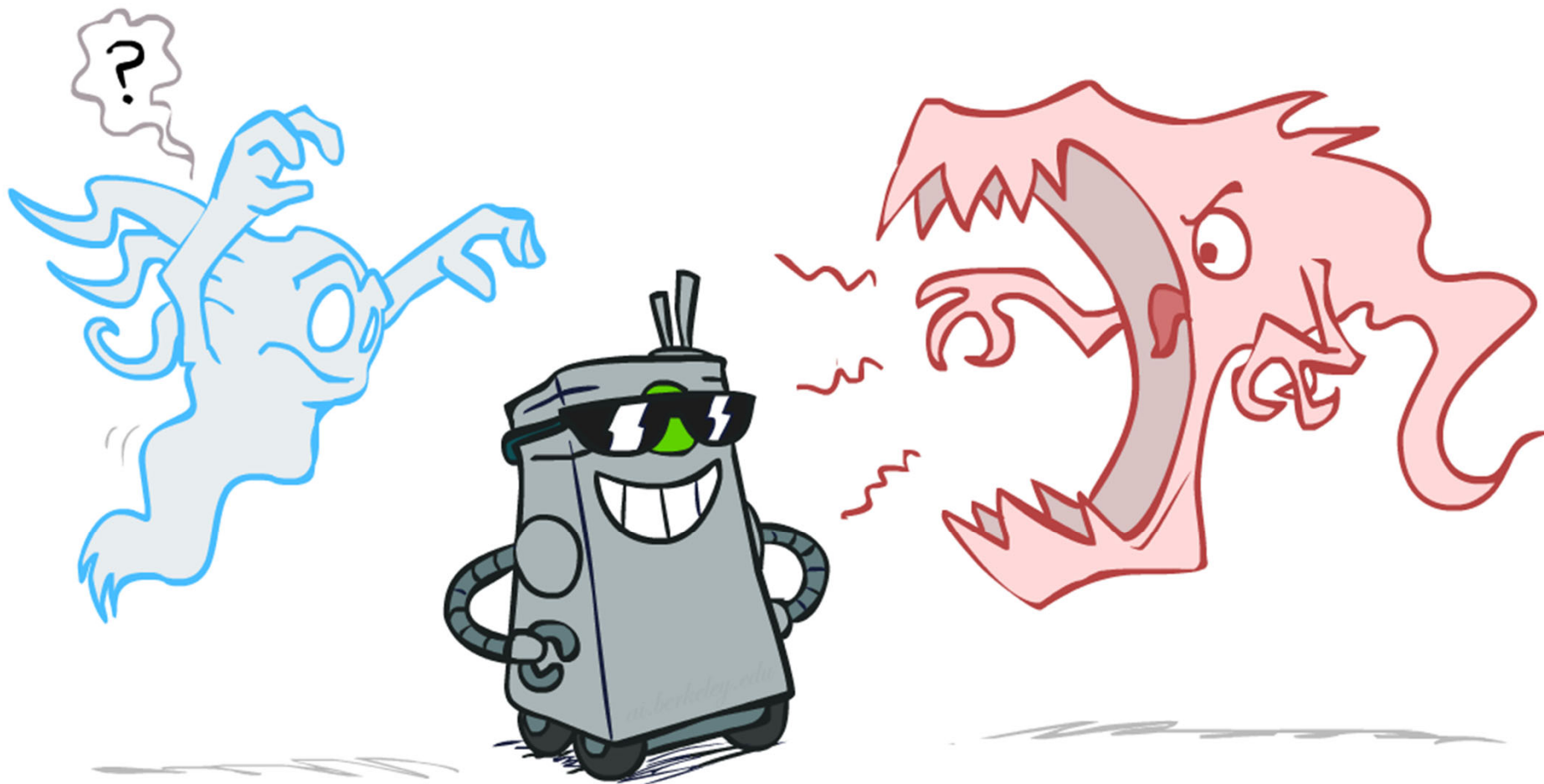
何が間違っていたか？



# 一般化とオーバーフィッティング

- 相対頻度のパラメータは訓練データにすぐにオーバーフィッティングする!
  - ピクセル(15,15) がONでラベルが3である訓練データが無くても、テストの際も無いとは限らない
  - “minute”という単語があれば 100% スпамである、とは言えない
  - “seriously”という単語があれば100% スпамでない、とも言えない
  - もし訓練セットに現れたことが無い単語があると？
  - 一般に、訓練データで見たことが無いからといって、確率がゼロであるとは言えない
- 極端な例を考えてみよう。メール全体で 1 つの特徴だとすると？ (文章IDなど)
  - 訓練データでは完全に上手く行く
  - 全く、一般化できない
  - bag-of-wordsの仮定をすることで少し一般化できるようになる。ただし十分ではない
- より良く一般化するためには: 推定を スムージング(smooth) あるいは 正則化(regularize) する

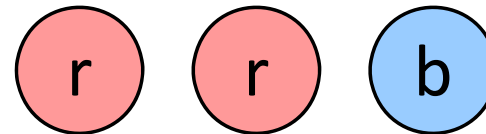
# スムージング (Smoothing)



# ラプラススムージング (Laplace Smoothing)

- ラプラスの推定:

- 全ての結果を、実際よりも1つ多く観測したとみなす



$$P_{LAP}(x) = \frac{c(x) + 1}{\sum_x [c(x) + 1]}$$

$$= \frac{c(x) + 1}{N + |X|}$$

$$P_{ML}(X) =$$

$$P_{LAP}(X) =$$

# ラプラススムージング (Laplace Smoothing)

- ラプラスの推定 (拡張):

- 実際よりも  $k$  回多く観測したとみなす

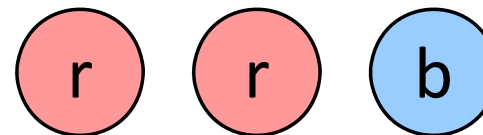
$$P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$$

- $k=0$  の場合は?
- $k$  は事前確率の **強さ** とよばれる

- 条件付き確率に対しては？

- 各条件を独立にスムージングする:

$$P_{LAP,k}(x|y) = \frac{c(x,y) + k}{c(y) + k|X|}$$



$$P_{LAP,0}(X) =$$

$$P_{LAP,1}(X) =$$

$$P_{LAP,100}(X) =$$

# 推定: 線形補完\*

- 実際には, ラプラススムージングは 以下のような  $P(X|Y)$  には 上手く行かないことが多い:
  - $|X|$  が非常に大きい場合
  - $|Y|$  が非常に大きい場合
- 他の選択肢: 線形補完
  - $P(X)$  をデータから経験的に得る
  - $P(X|Y)$  が、経験的に得られた  $P(X)$  と、あまりには違わないようにする
$$P_{LIN}(x|y) = \alpha \hat{P}(x|y) + (1.0 - \alpha) \hat{P}(x)$$
  - $\alpha$  が 0 や 1 の場合にはどういう意味になるか?

# スパムフィルタの例 ・ ・ ・ スムージング後

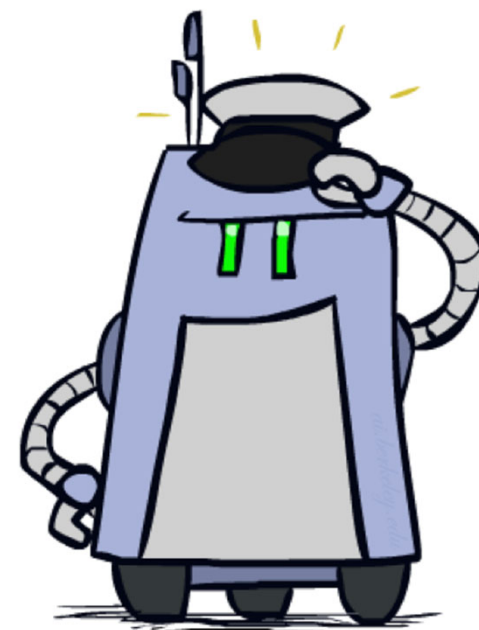
- 実際の分類問題において、スムージングは極めて重要
- 新しいオッズ比:

$$\frac{P(W|\text{ham})}{P(W|\text{spam})}$$

helvetica	:	11.4
seems	:	10.8
group	:	10.2
ago	:	8.4
areas	:	8.3
...		

$$\frac{P(W|\text{spam})}{P(W|\text{ham})}$$

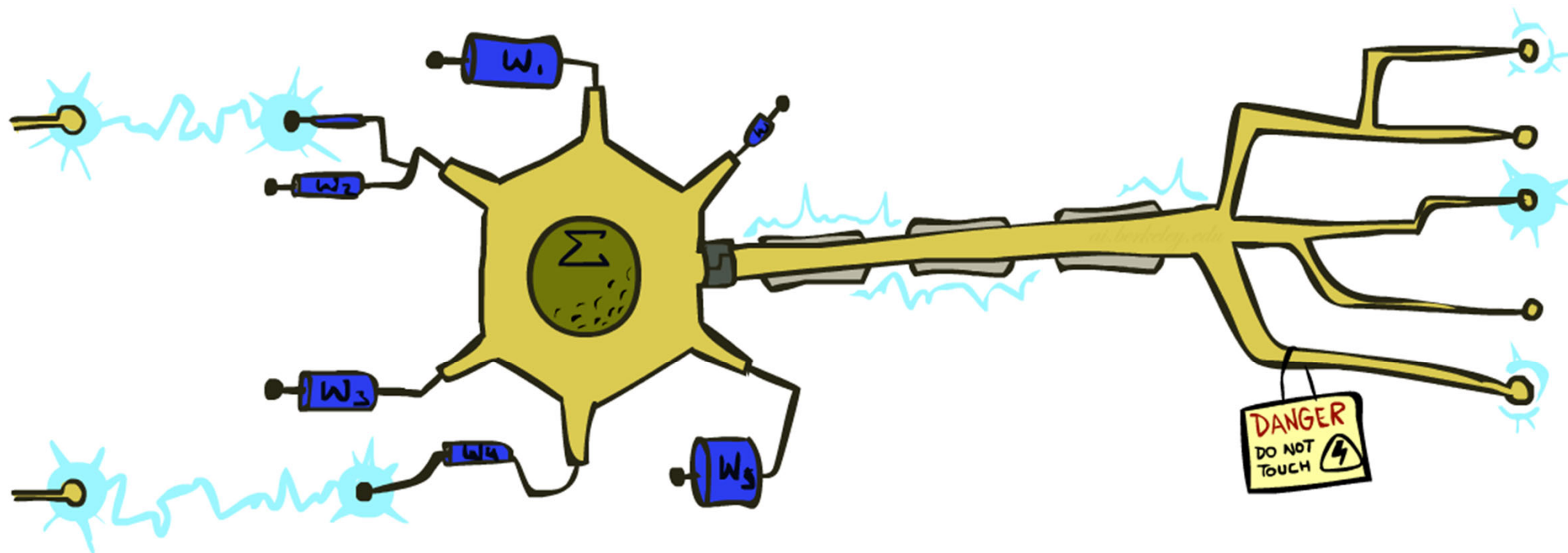
verdana	:	28.8
Credit	:	28.4
ORDER	:	27.2
<FONT>	:	26.9
money	:	26.5
...		



以前のものよりも意味がありそうですね



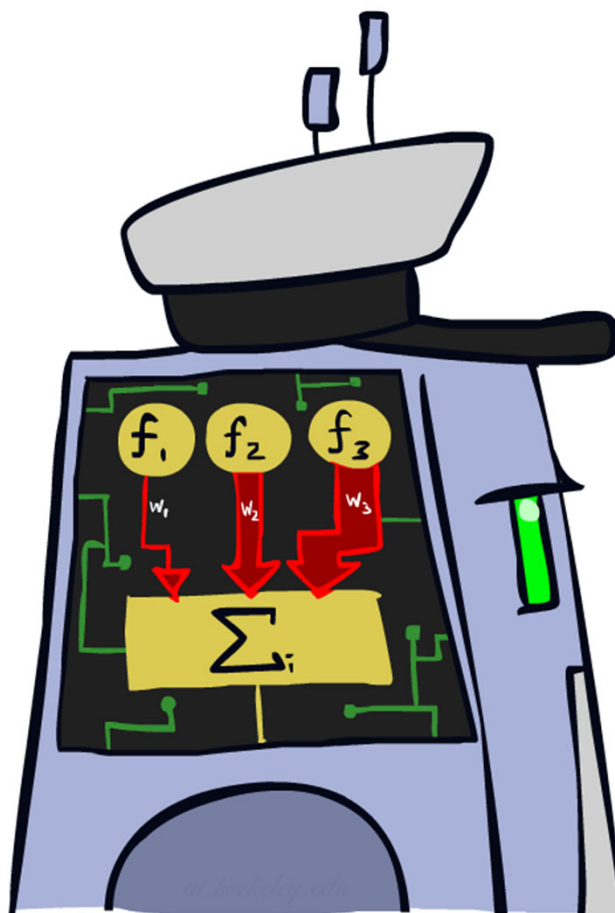
# パーセプトロン



# エラー駆動の分類 (Error-Driven Classification)



# 線形分類器 (Linear Classifiers)



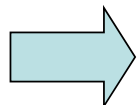
# 特徴ベクトル

$x$

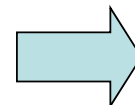
$f(x)$

$y$

```
Hello,  
  
Do you want free printer  
cartridges? Why pay more  
when you can get them  
ABSOLUTELY FREE! Just
```

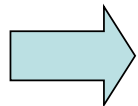


```
( # free      : 2  
  YOUR_NAME   : 0  
  MISSPELLED  : 2  
  FROM_FRIEND : 0  
  ... )
```

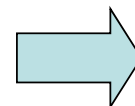


SPAM  
or  
+

2



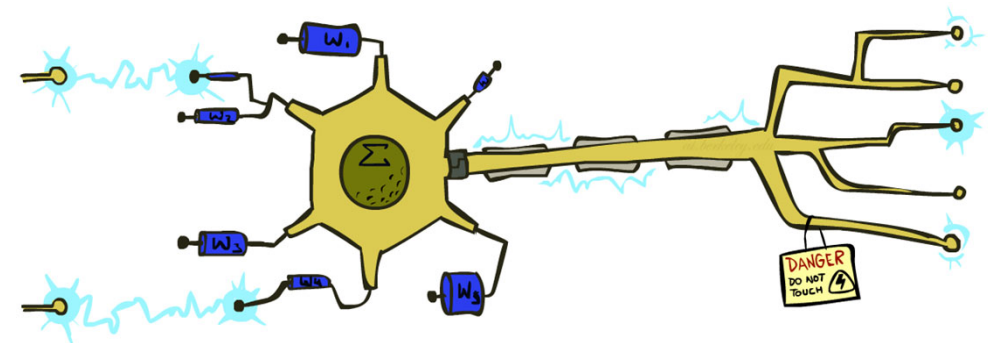
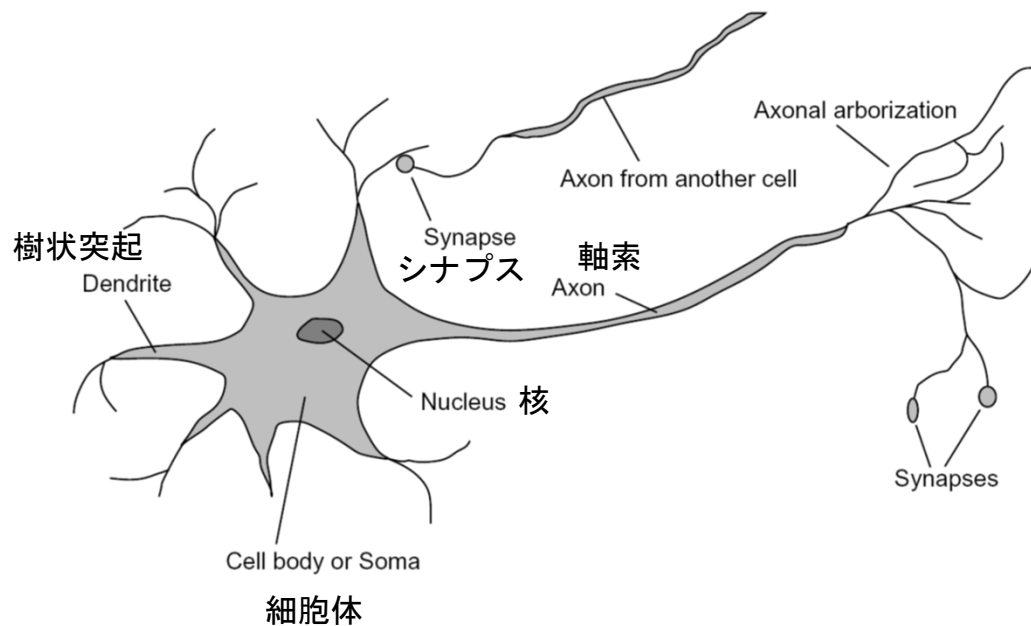
```
( PIXEL-7,12   : 1  
  PIXEL-7,13   : 0  
  ...  
  NUM_LOOPS    : 1  
  ... )
```



"2"

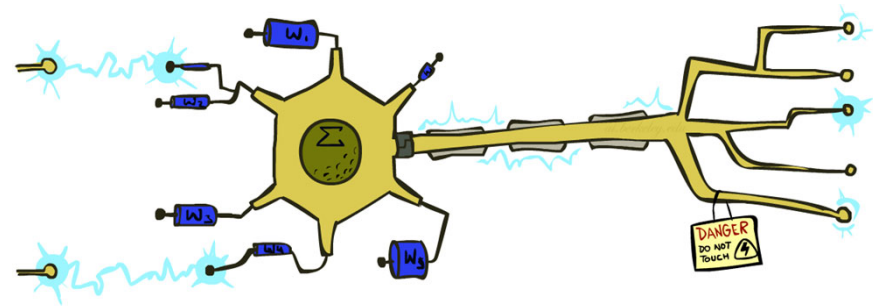
# (単純化した) 生物学

- 大まかな着想のもと: 人間のニューロン



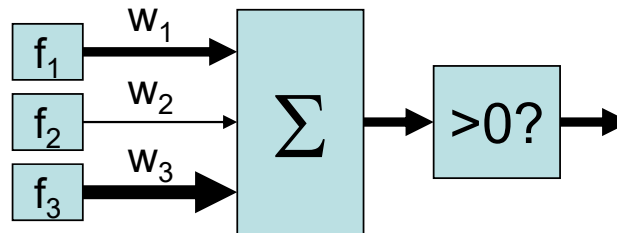
# 線形分類器

- 入力は **特徴の値**
- それぞれの特徴は **重み** を持つ
- 合計は **活性化 (activation)**



$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- もし活性化が:
  - 正なら, +1 を出力
  - 負なら, -1 を出力



# 重み

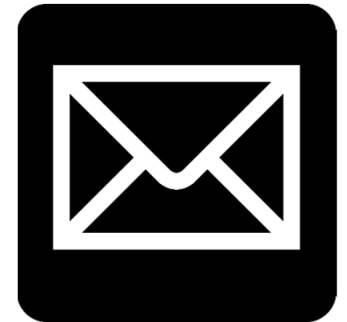
- 2クラス分類: 特徴量と重みベクトルを比較する
- 学習: 重みベクトルを例から見出す

```
# free      : 4  
YOUR_NAME   :-1  
MISPELLED   : 1  
FROM_FRIEND :-3  
...
```

$w$

$f(x_1)$

```
# free      : 2  
YOUR_NAME   : 0  
MISPELLED   : 2  
FROM_FRIEND : 0  
...
```



$f(x_2)$

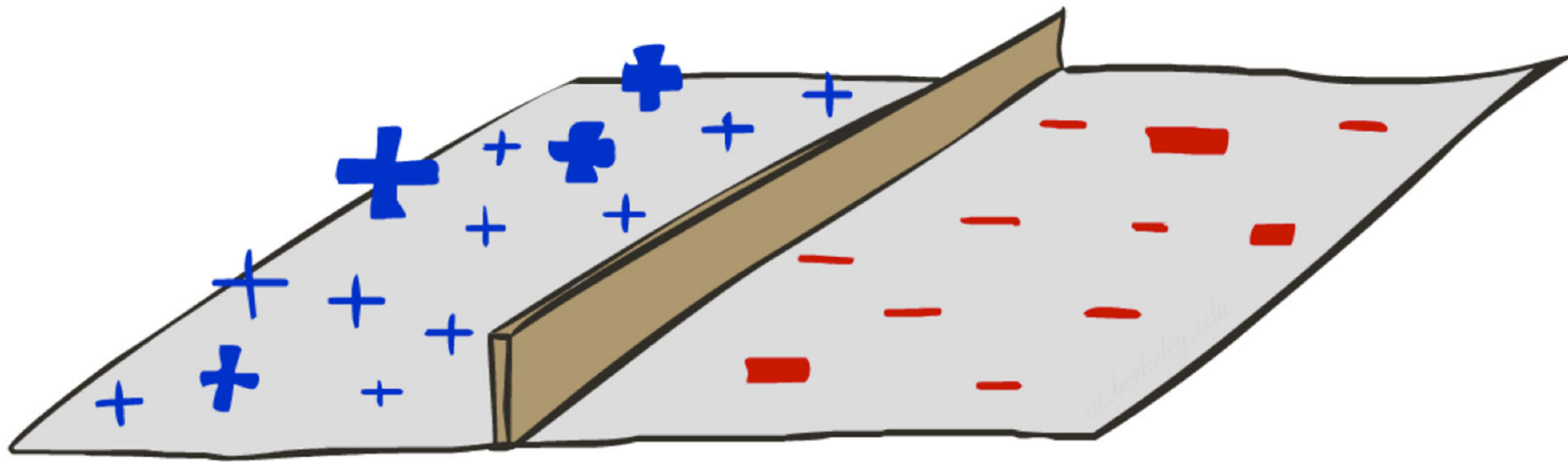
```
# free      : 0  
YOUR_NAME   : 1  
MISPELLED   : 1  
FROM_FRIEND : 1  
...
```



内積  $w \cdot f$  が正であれば、  
正のクラスに分類

# 決定ルール (Decision Rules)

---

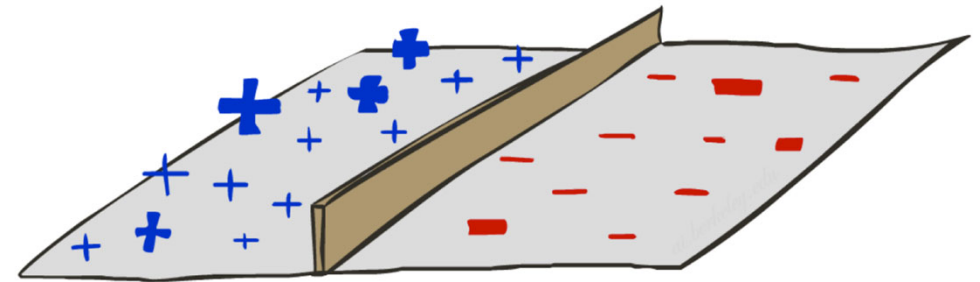




# 2クラス分類の決定ルール

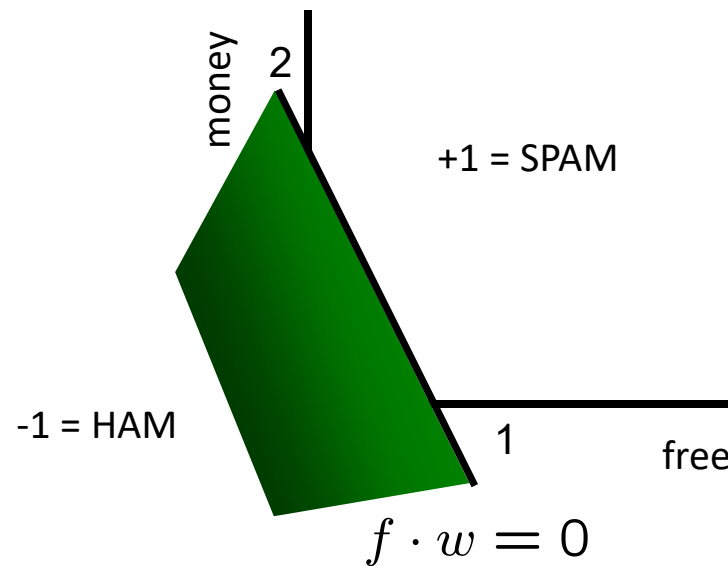
## ■ 特徴ベクトル空間において

- 各点が1つの例を表す
- 重みベクトルは超平面
- 片側が  $Y=+1$  に対応
- 反対側が  $Y=-1$  に対応



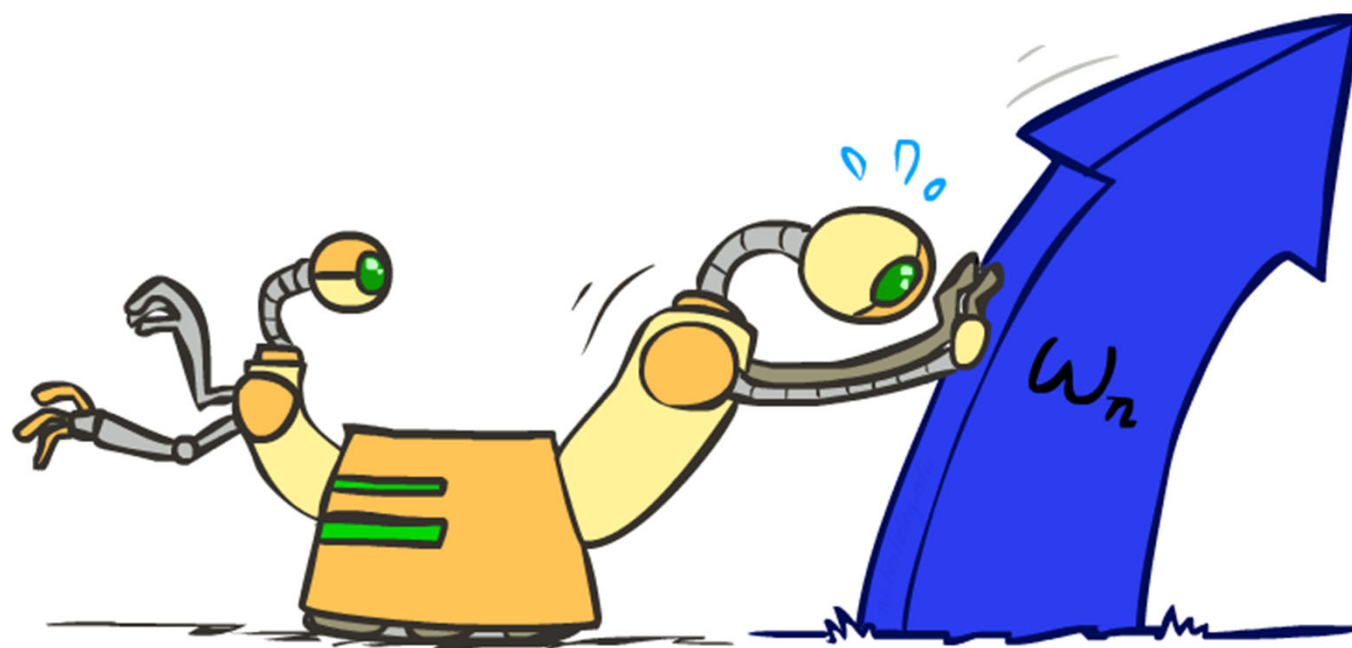
$w$

BIAS	:	-3
free	:	4
money	:	2
...		



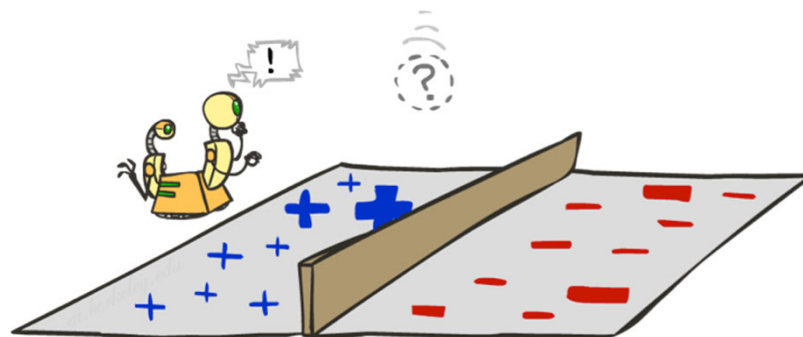
※ BIAS項・・・特徴ベクトルでは常に1の値

# 重みの更新

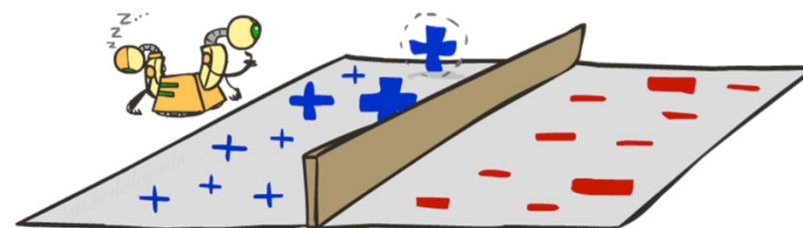


# パーセプトロンの学習: 2クラスの認識問題

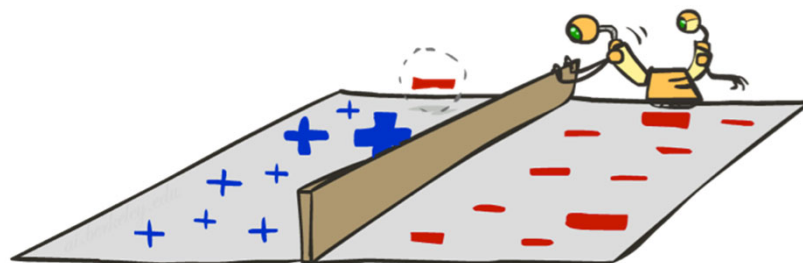
- 重み = 0 から開始
- 各訓練サンプルごとに:
  - 今の重みで分類



- 正しければ (i.e.,  $y=y^*$ ), 変化なし!



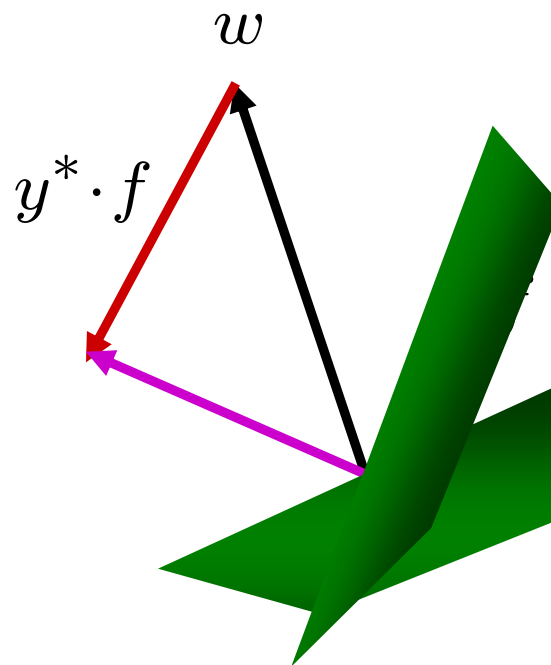
- 間違えたら: 重みベクトルを調整



# パーセプトロンの学習: 2クラスの認識問題

- 重み = 0 から開始
- 各訓練サンプルごとに:
  - 今の重みで分類
$$y = \begin{cases} +1 & \text{if } w \cdot f(x) \geq 0 \\ -1 & \text{if } w \cdot f(x) < 0 \end{cases}$$
  - 正しければ (i.e.,  $y=y^*$ ), 変化なし!
  - 間違えたら: 重みベクトルから特徴ベクトルを足し引きして調整。  
もし  $y^*$  が -1 なら引く。

$$w = w + y^* \cdot f$$



# パーセプトロン 演習

- 実際に、右の4サンプルから、重み  $w_1, w_2, w_3$  を更新してみよう

重み     $w_1$     $w_2$     $w_3$   
          0     0     0

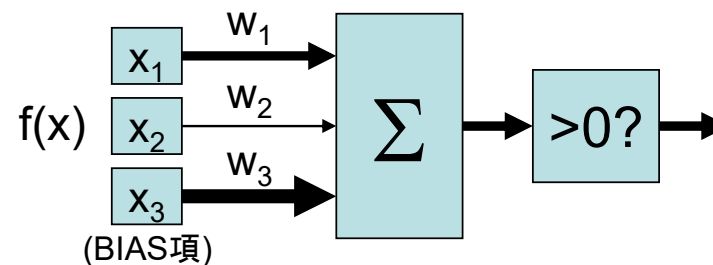
## 参考

- 重み = 0 から開始
  - 各訓練サンプルごとに:
    - 今の重みで分類
- $$y = \begin{cases} +1 & \text{if } w \cdot f(x) \geq 0 \\ -1 & \text{if } w \cdot f(x) < 0 \end{cases}$$
- 正しければ (i.e.,  $y=y^*$ ), 変化なし!
  - 間違えたら: 重みベクトルから特徴ベクトルを足し引きして調整。  
もし  $y^*$  が -1 なら引く。

$$w = w + y^* \cdot f$$

$f(x)$  のサンプル (OR関数)

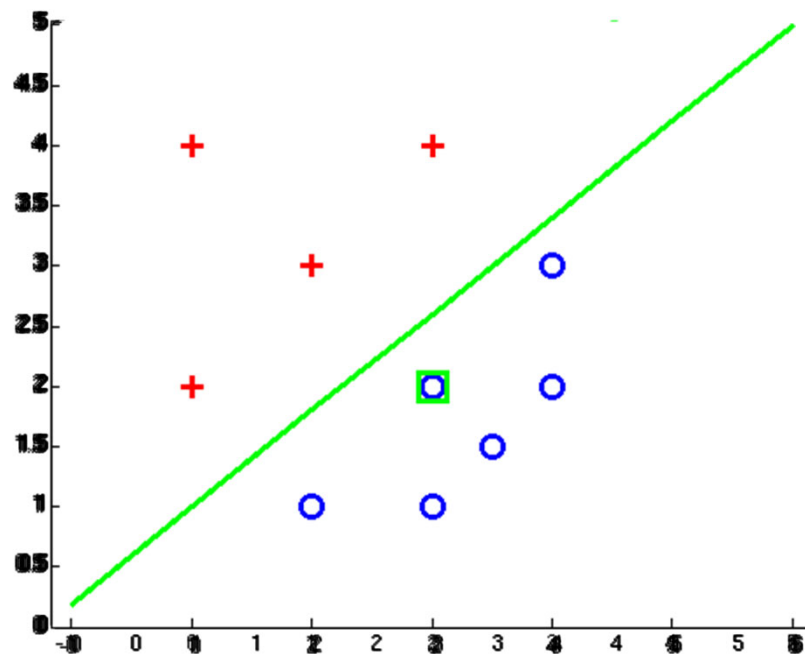
$x_1$	$x_2$	$y^*$
0	0	-1
0	1	1
1	0	1
1	1	1



BIAS項 常に 1 の値にする

# 例:パーセプトロン

- 分離可能な例

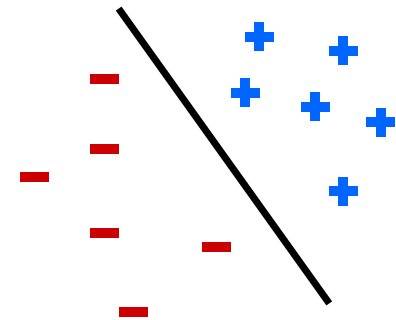


# パーセプトロンの特徴

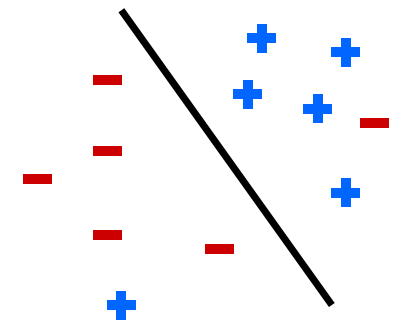
- 分離可能: 訓練セットを完全に正しく分類できるようなパラメータが存在するなら、分離可能という
- 収束性: 訓練データが分離可能であれば、パーセプトロンは最終的には収束（2クラスの場合）
- 誤りの上限 (Mistake Bound): 2クラス分類で、誤った分類の最大数は マージン あるいは 分離可能性 に依存

$$\text{mistakes} < \frac{k}{\delta^2}$$

(線形) 分離可能



(線形) 分離不可能



# 分類学習の比較

- 単純ベイズ分類器：

- モデルベースの学習
- クラス予測の確率を計算
- 特徴量の独立性についての強い仮定
- 訓練データを1度だけ処理 (カウントする)

- パーセプトロン・・・一種の「線形分類器」：

- エラー駆動の学習、汎化誤差の最小化
- データに関する仮定はあまりない
- 訓練データを何度も（何エポックも）処理
- たいていの場合、より正確
- 多くの改良・発展形 (e.g. SVM、ニューラルネットワーク、深層学習)  
※ 最も基本形のパーセプトロンを「単純パーセプトロン」と呼ぶこともある



# 引用文献

---

- S. Russell and P. Norvig, Artificial Intelligence A Modern Approach, 3rd edition, Pearson Education Limited, 2016.
- UC Berkeley CS188 Intro to AI -- Course Materials  
<http://ai.berkeley.edu/>