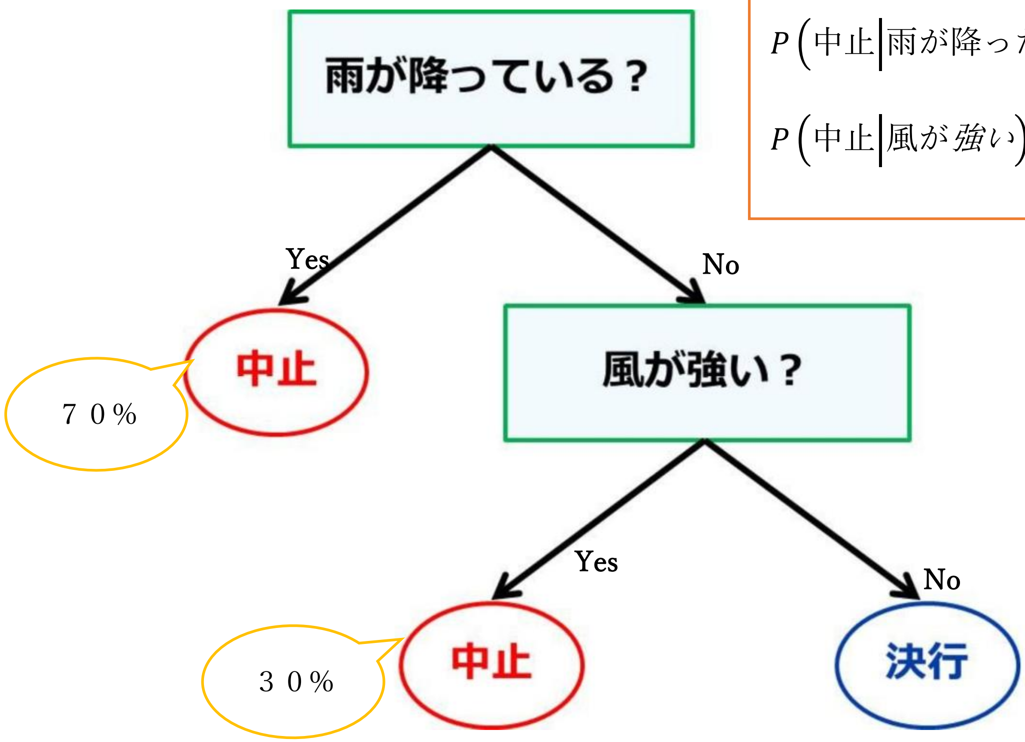
機械学習の入門講座！

大まかに分析モデルを理解してみる

決定木



$P(\text{中止}|\text{雨が降った})$ → 『雨が降った』後に『中止』になる確率

$P(\text{中止}|\text{風が強い})$ → 『風が強い事象』が発生した後に『中止』になる確率

出力 →

$P(\text{中止}|\text{雨が降った}) \quad = \quad ７０％$

$P(\text{中止}|\text{風が強い}) \quad = \quad ３０％$

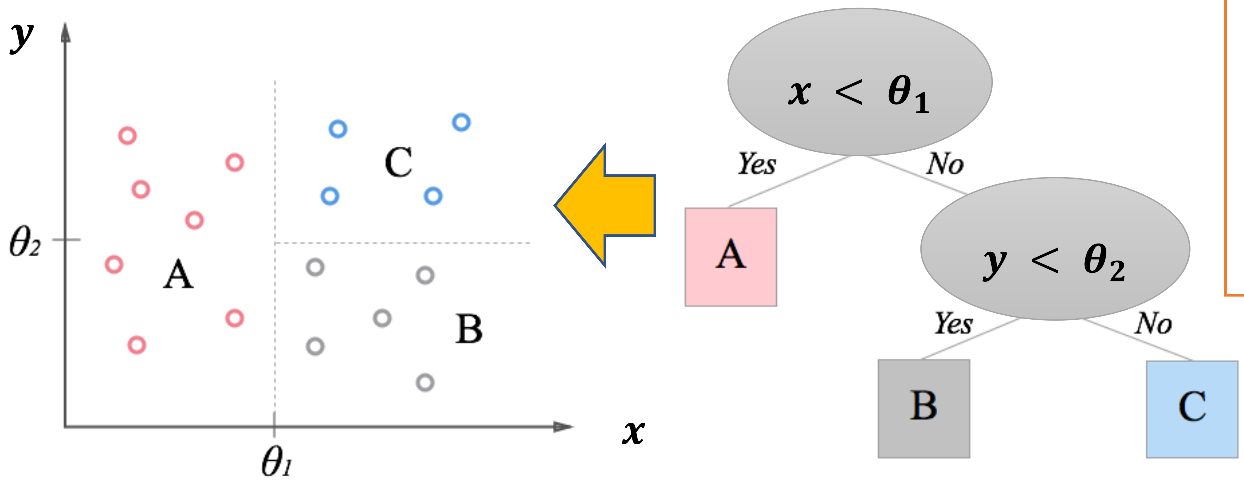
どれだけ細かく分類できたか！

ジニ不純度(未分類データの多さ　[0~1])

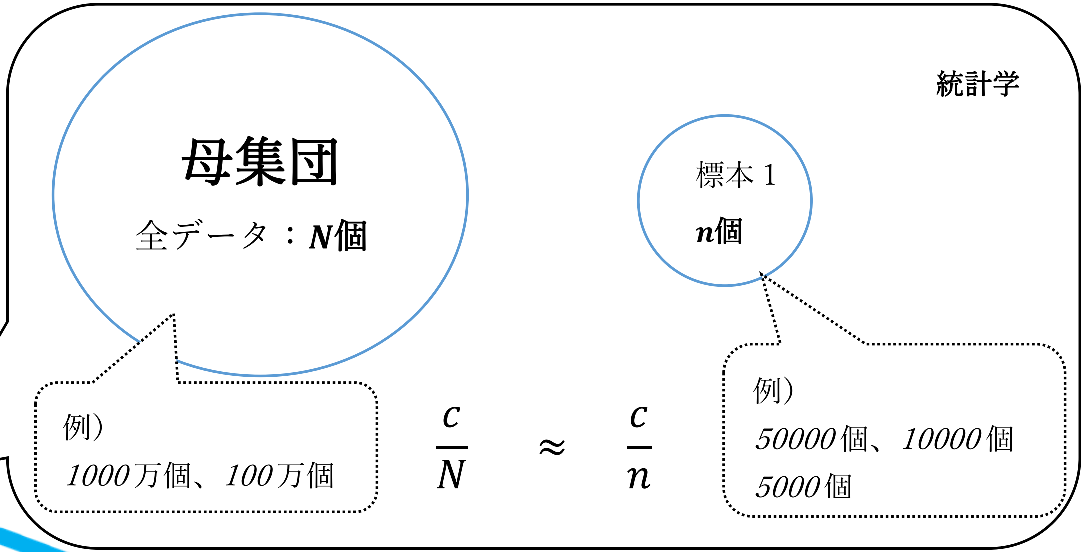
$$1-\sum_{i=1}^{n} P_i\left(i\text{番目の事象}|\text{全体}\right)^2$$

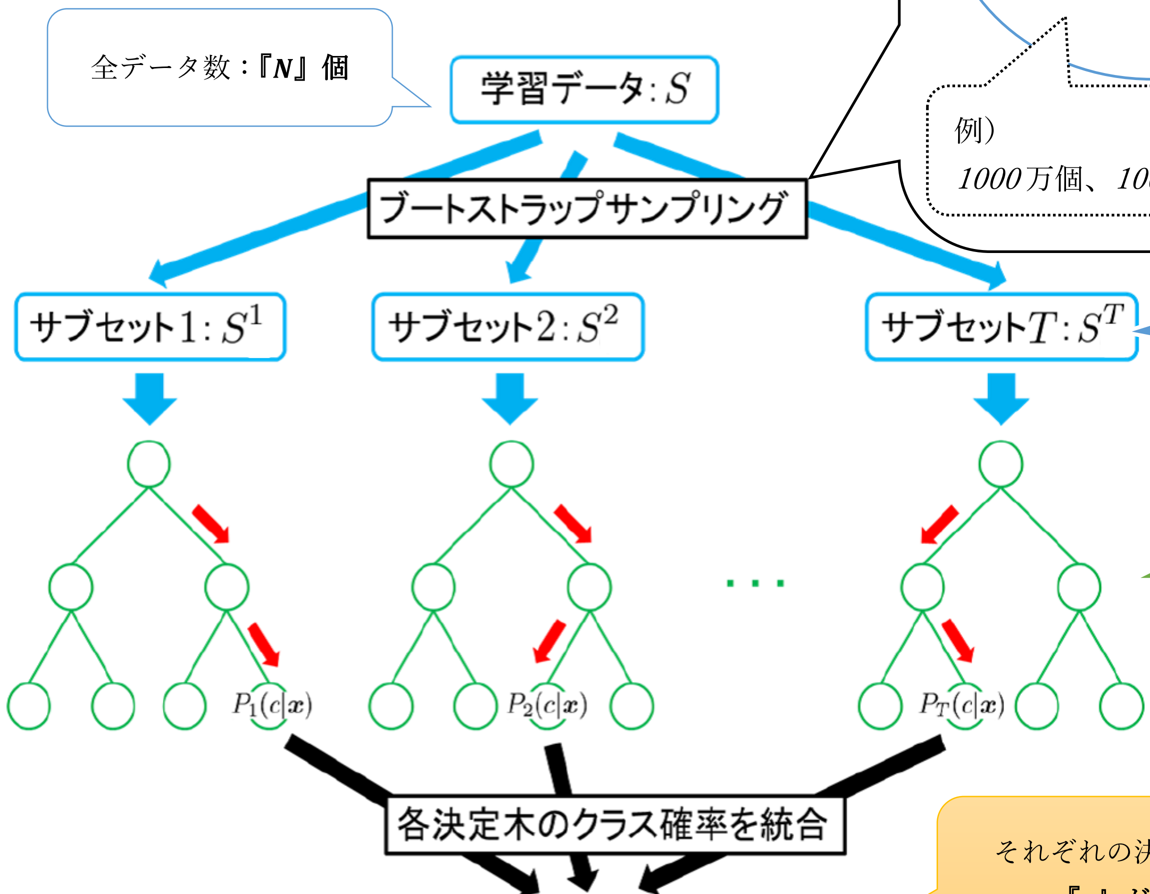
例） $P(\text{中止}|\text{雨})=\frac{7}{10}$, $P(\text{中止}|\text{強風})=\frac{3}{10}$, $P(\text{決行})=0$

$$1-\left[\left(\frac{7}{10}\right)^2+\left(\frac{3}{10}\right)^2+0\right]=\frac{42}{100}=0.42$$



ランダムフォレスト





$$P(c|\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T} P_t(c|\boldsymbol{x})$$

サンプルの標本を T 個準備し、
一つの標本に全体からランダムに『n』個とる

それぞれの標本(サブセット 1~T)で、
決定木で実施してみる。

それぞれの決定木から導出された、
『c』が起こる確率の平均値 ＝ 『c』の確立を多数決で決める

確率の期待値

決定木×ブースティング

バギング： 並列学習　　ex） ランダムフォレスト
訓練用にサンプル標本をを複数作成し、それぞれ独立に学習させて平均や多数決を行い分類精度を高める手法。

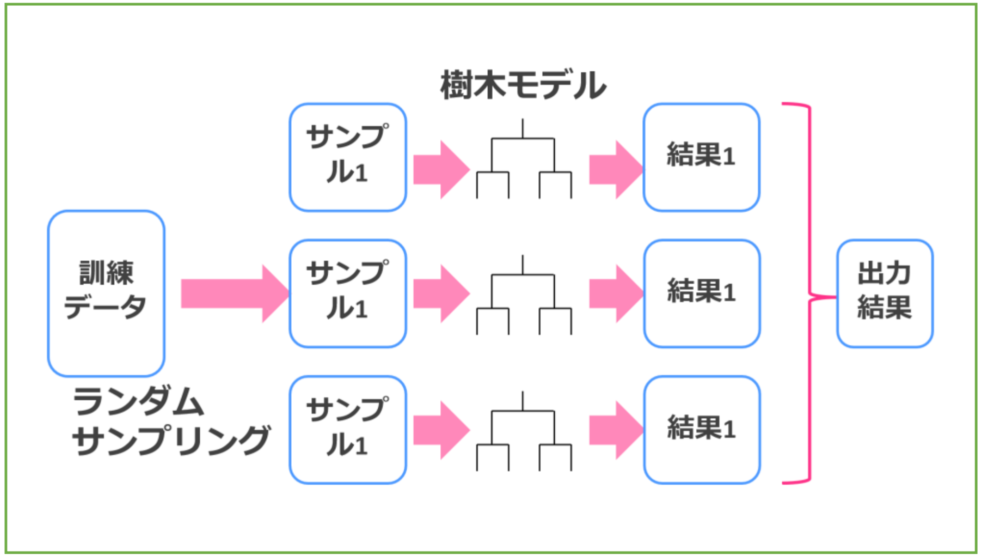
重要な葉だけ掘り下げる

ブースティング： 直列学習　　ex） LightGBM,　Xgboost
始めに弱学習器で予測を行い、誤分類のサンプルを訂正するように別の学習モデルをどんどん追加して予測精度を高める手法

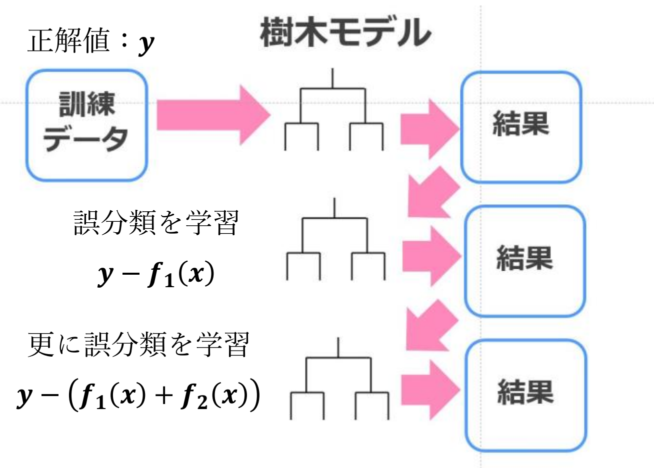
アンサンブル学習

複数の学習器(学習モデル)を用いて学習する方法

バギング



ブースティング

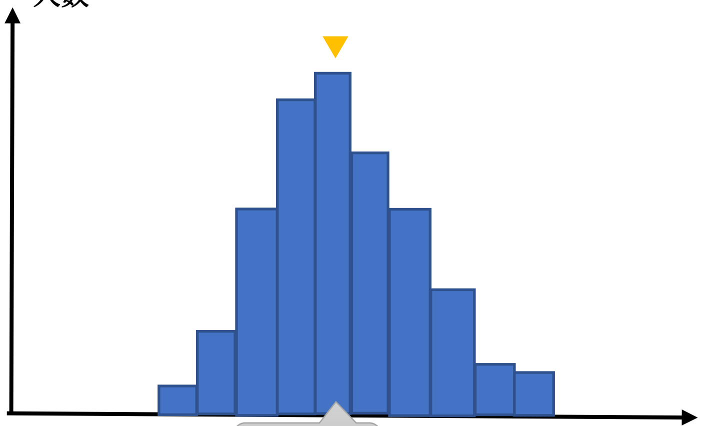


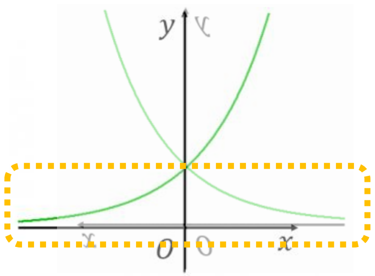
予測値： $f(x) = \sum_{i=1}^{n} f_i(x) = f_1(x) + f_2(x) + \cdots + f_n(x)$

誤差： $y - (f_1(x) + f_2(x) + \cdots + f_n(x))$

『LightGBM』や『Xgboost』は、
AI のコンテストでもよく使われる！

正規分布 1



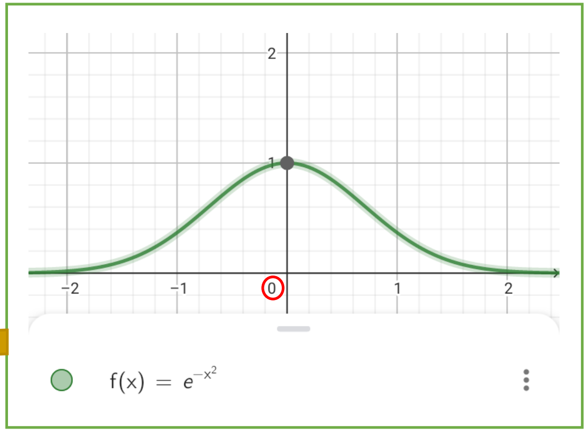


Y 軸で回転させると…

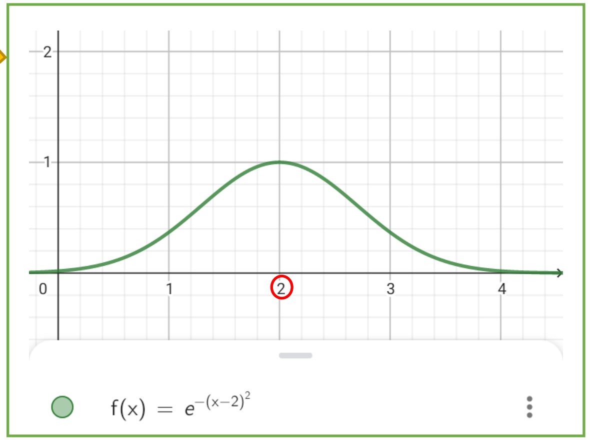
⊖(⊖×⊖) = ⊖

⊖(⊕×⊕) = ⊖

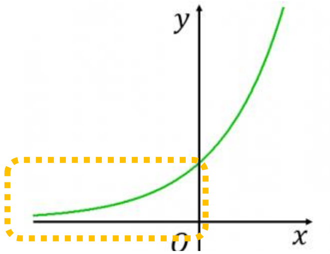
e^{-x^2}



$e^{-(x-2)^2}$



指数関数のグラフは指数が⊖の時に、限りなく０に近づいていく。



⊖乗は、中心から離れるほど小さくなる。

$$10^{-2} = \frac{1}{10^2} = \frac{1}{100} = 0.01$$

$$10^{-3} = \frac{1}{10^3} = \frac{1}{1000} = 0.001$$

$$10^{-4} = \frac{1}{10^4} = \frac{1}{10000} = 0.0001$$

復習！

$a^5 = a \cdot a \cdot a \cdot a \cdot a$　　$a^3 \times a^2 = (a \cdot a \cdot a)(a \cdot a) = a^{3+2} = a^5$

$(a^2)^4 = \underbrace{(a^2)\cdot(a^2)\cdot(a^2)\cdot(a^2)}_{4\text{個}} = \underbrace{(a \cdot a)(a \cdot a)(a \cdot a)(a \cdot a)}_{} = a^{2\times4} = a^8$

『ハンバーガー2 個のセット』を 4 つ買う
=全部で 8 個のハンバーガーを購入

⊖乗が必要な理由

分子　分母

$a^5 \div a^2 = \frac{a^5}{a^2} = \frac{\cancel{a} \cdot \cancel{a} \cdot a \cdot a \cdot a}{\cancel{a} \cdot \cancel{a}} = a^{5-2} = a^5 \times a^{-2} = a^3$

分子の指数を『2』減らす

0 乗が必要な理由

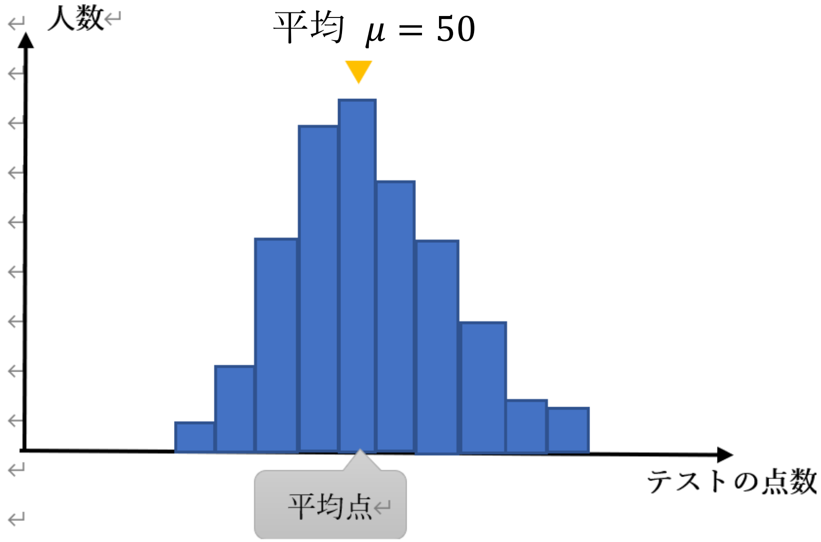
$a^5 \div a^5 = \frac{a^5}{a^5} = \frac{a \cdot a \cdot a \cdot a \cdot a}{a \cdot a \cdot a \cdot a \cdot a} = a^{5-5} = a^0 = 1$　⇒　$a^0 = 1$,

指数が分数の場合

$(a^{\frac{1}{2}})^2 = a^{\frac{1}{2}} \times a^{\frac{1}{2}} = a^{\frac{1}{2}+\frac{1}{2}} = a^1 \Leftrightarrow a^{\frac{1}{2}} = \sqrt{a}$

$(a^{\frac{1}{3}})^3 = a^1 \Leftrightarrow a^{\frac{1}{3}} = \sqrt[3]{a}$

正規分布 2



ある人の点数

平均点

$$\text{分散}\ (\sigma^2) = \frac{1}{n}\Sigma_{i=0}^{n}(x_i - \mu)^2$$

➡『中心からの差分』を平均したもの

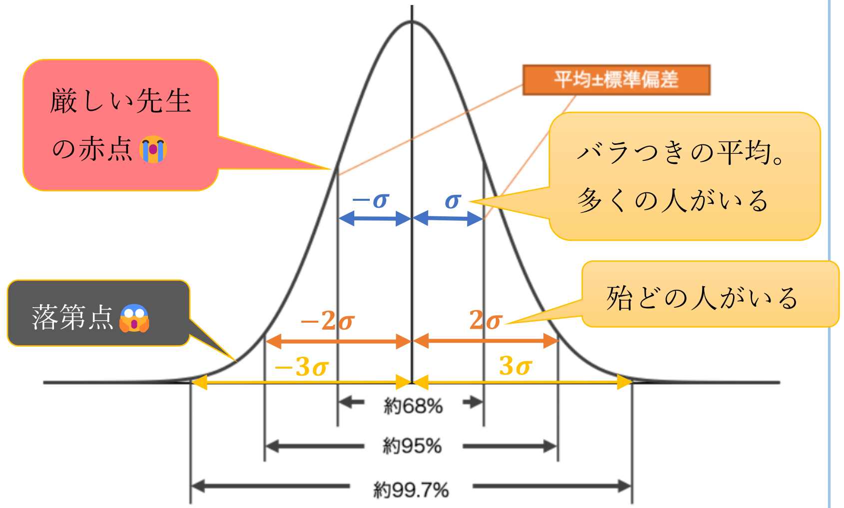
『バラつき度合い』
を平均したもの

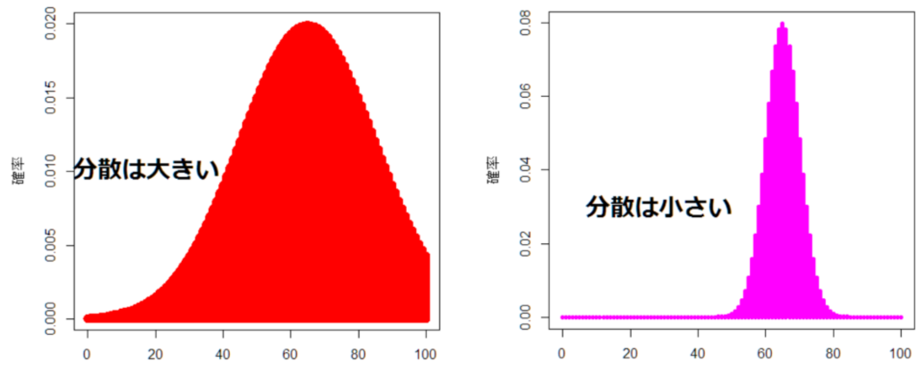
分散を root で囲ったものを**標準偏差($\boldsymbol{\sigma}$)**という

$$\sqrt{\sigma^2} = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(x-\mu)^2}$$

二乗から戻したイメージ。

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{(x-\mu)^2}{2\sigma}}$$

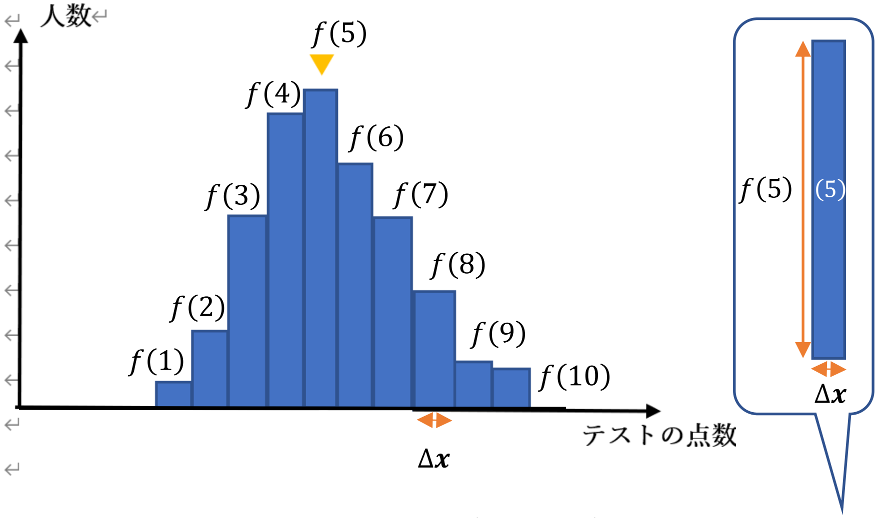


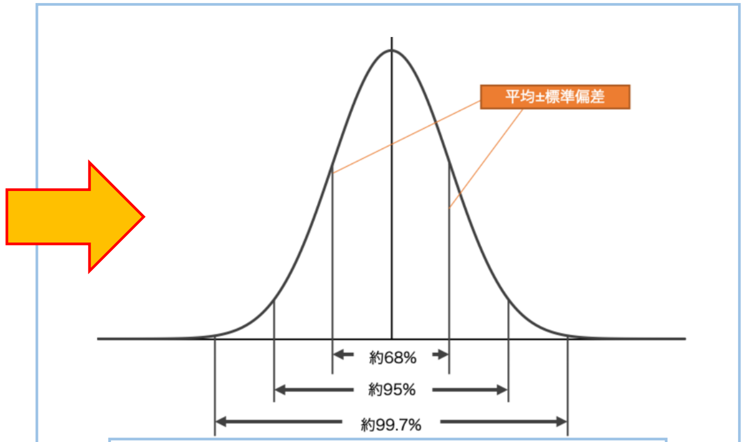


正規分布 3

関数の値が高いところほど、多くのデータが存在　＝『確率の分布を表している』

しかし、関数は連続量なので、0~100 までに無限に値が存在する。50 点をとる確率は、『1/ ∞ ≈ 0』となり確率を扱えない。そこで、下の最初のテストの点数の分布のグラフを見てみる。



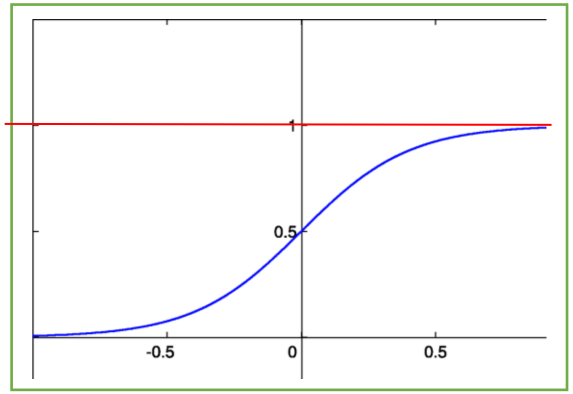


$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma}}$$

全体に対しての平均点グループ『(5)』(50~60 点)の割合は、**短冊の面積**で考えると

$$\frac{f(5)\cdot\Delta\boldsymbol{x}}{\sum_{i=1}^{10} f(i)\cdot\Delta\boldsymbol{x}} = \frac{f(5)\cdot\Delta\boldsymbol{x}}{f(1)\cdot\Delta\boldsymbol{x} + f(2)\cdot\Delta\boldsymbol{x} + \cdots + f(10)\cdot\Delta\boldsymbol{x}}$$

となり、面積で考えると割合を定量的に扱えるようになるのである！



$$sigmoid(x) \;=\; \frac{1}{1+e^{-x}}$$

0~1

このように、連続量で表すガウス分布でも、面積で考えることで、
確率を扱えるようになるのだ！
ガウス分布では、下記のように 1 になるように面積を定義している。

積分で『1』にする為の定数

$$\boxed{\frac{1}{\sqrt{2\pi\sigma}}}\int_{-\infty}^{\infty} e^{-\frac{(x-\mu)^2}{2\sigma}}\,dx \;=\; 1$$

イコールではないが、近似的にそう扱える

$$\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma}}$$

積分

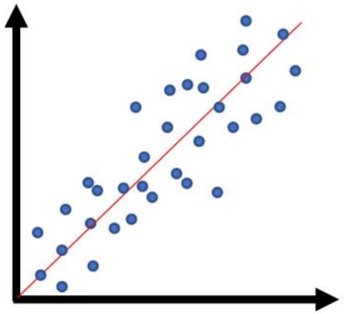
微分

$$sigmoid(x)$$

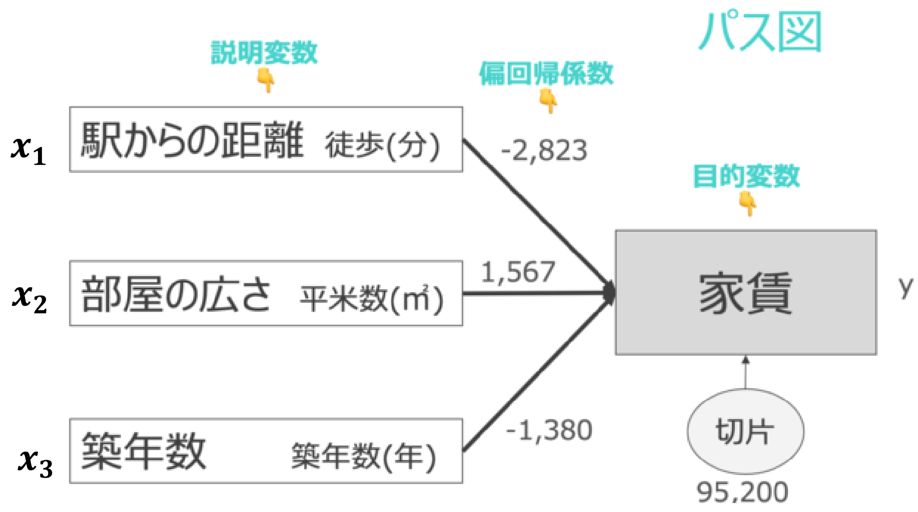
回帰分析

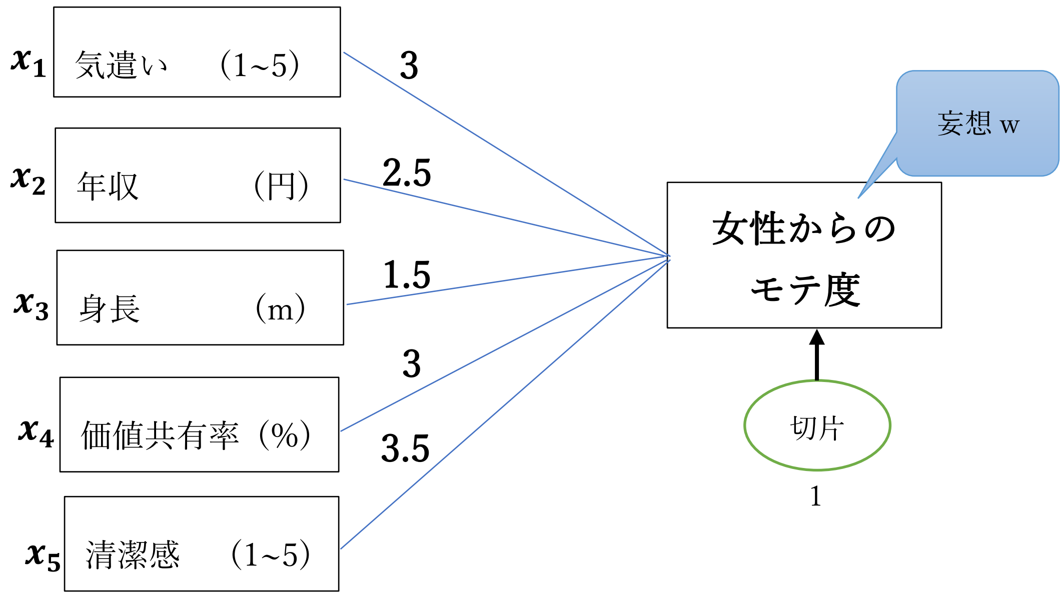
単回帰

$y = ax + b$



重回帰



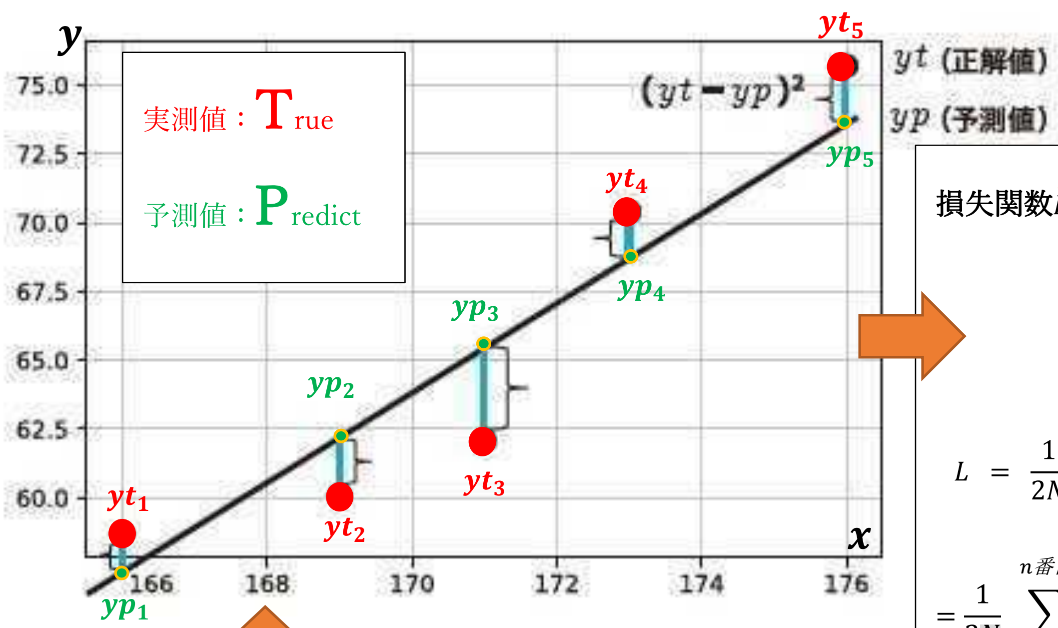


家賃：$y = -2823\boldsymbol{x_1} + 1567\boldsymbol{x_2} - 1380\boldsymbol{x_3} + 95200$

モテ度：$y = 3\boldsymbol{x_1} + 2.5\boldsymbol{x_2} + 1.5\boldsymbol{x_3} + 3\boldsymbol{x_4} + 3.5\boldsymbol{x_5} + 1$

最小二乗法

：「誤差の二乗」が最小になるようにしていく時の性質を利用して、最適な傾きを求める方法



誤差が大きいほど
損失関数は大きくなる

損失関数L　誤差の二乗の平均値

データの個数:N
＝実測値ytの個数
この例では５個

N 番目の実測値

N 番目の予測値

$$L = \frac{1}{2N}\sum_{1\,番目}^{n\,番目}\left(yt_{N\,番目} - yp_{n\,番目}\right)^2$$

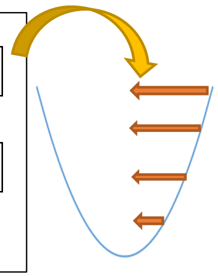
$$= \frac{1}{2N}\sum_{1\,番目}^{n\,番目}\left(yt_{N\,番目} - [ax_N + b]\right)^2$$

$$= \frac{1}{2\cdot 5}\{(yt_1 - yp_1)^2 + (yt_2 - yp_2)^2 + (yt_3 - yp_3)^2 + (yt_4 - yp_4)^2 + (yt_5 - yp_5)^2\}$$

損失関数が $\boxed{大}$ ＝ 修正 $\boxed{多}$

損失関数が $\boxed{小}$ ＝ 修正 $\boxed{少}$

とすれば良い！



$$y = a^{修正M回目}\cdot x + b$$

損失関数Lが最小値 ➡ **損失関数の微分 ＝ ほぼ０**（接線の傾きが０）

勾配降下法

$$\frac{\partial}{\partial U}\frac{\partial U}{\partial yp}U^2 = \frac{\partial}{\partial U}U^2\cdot\frac{\partial U}{\partial yp}$$

$$L = \frac{1}{2N}\sum_{1}^{N}(yt_N - yp_N)^2$$

$$\frac{\partial L}{\partial a} = \frac{\partial L}{\partial yp}\frac{\partial yp}{\partial a} = \frac{1}{2N}\cdot\frac{\partial}{\partial yp}(\underbrace{yt - yp}_{U})^2\cdot\frac{\partial}{\partial a}(ax_n + b)$$

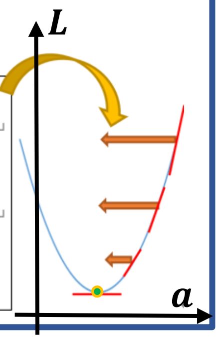
$$= \frac{1}{2N}\cdot\frac{\partial}{\partial U}U^2\cdot\frac{\partial}{\partial yp}(yt - yp)\cdot x_n$$

$$= \frac{1}{2N}\cdot 2(yt - yp)\cdot 1\cdot x = \frac{1}{N}\cdot(yt - yp)\cdot x_n$$

損失関数が $\boxed{大}$ ＝ 修正 $\boxed{多}$

損失関数が $\boxed{小}$ ＝ 修正 $\boxed{少}$

とすれば良い！



直線の傾きを修正

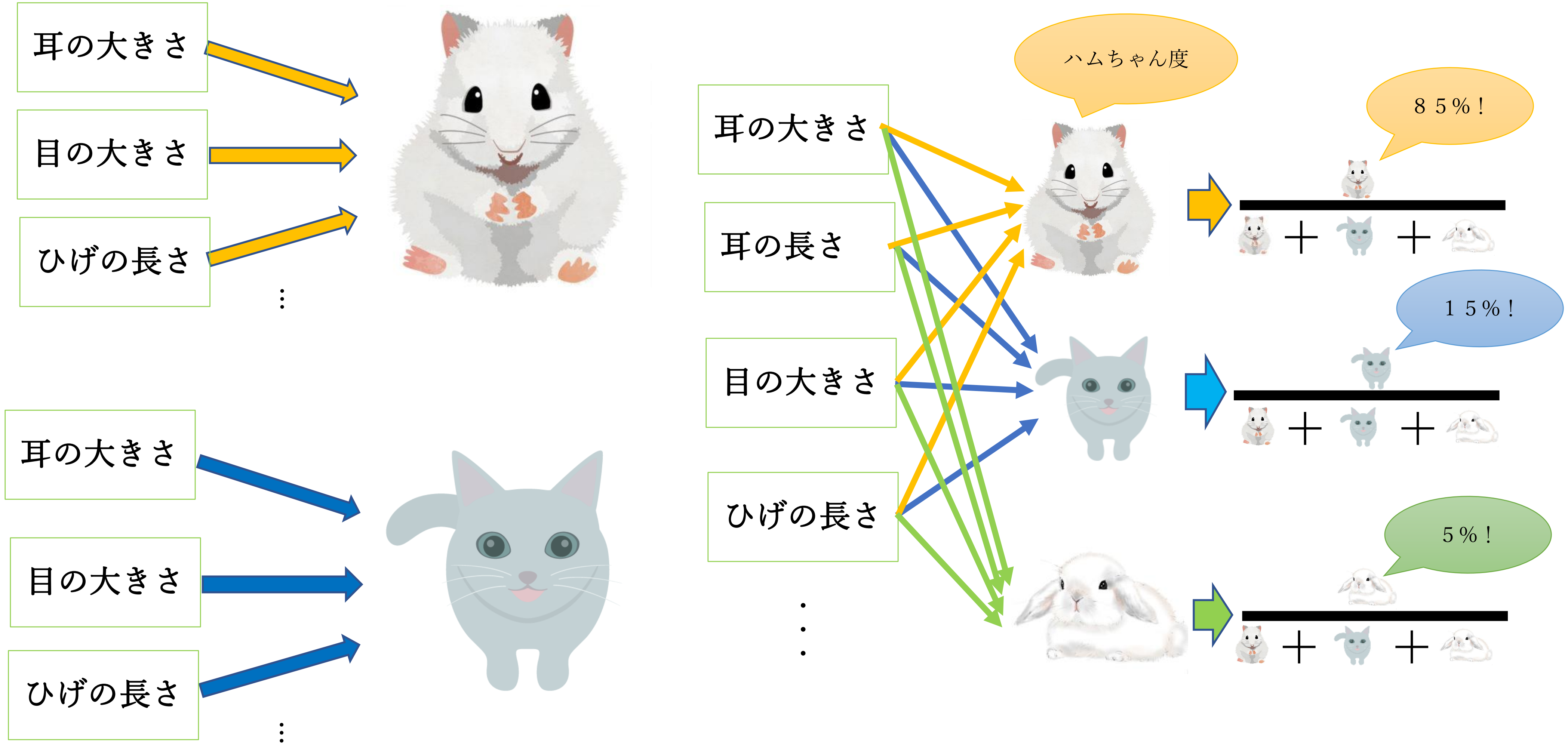
定数(学習率)
0.1~0.01

テキトーに決める！

$$a^{修正(M+1)回目} = a^{修正M回目} - \alpha\ \frac{1}{N}\sum_{1}^{n}(yt_n - yp_n)\,x_n$$

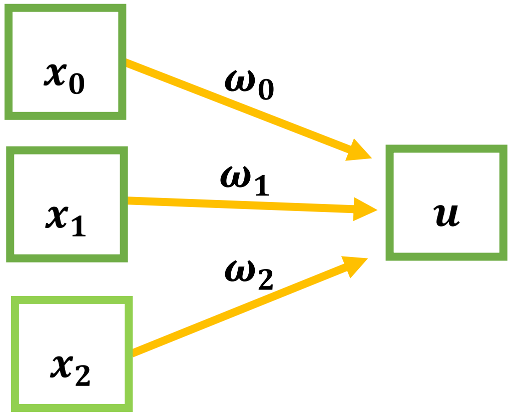
$$(yt_1 - yp_1)\,x_1 + (yt_2 - yp_2)\,x_2 + \cdots + (yt_5 - yp_5)\,x_5$$

多値分類１



その動物の度合い ＝【目の大きさ】＋【耳の大きさ】＋【ひげの長さ】＋ …

多値分類２



$$u = \omega_0 x_0 + \omega_1 x_1 + \omega x_2$$

$x_0 = 1$

$$\boxed{u = \omega_0 + \omega_1 x_1 + \omega_2 x_2}$$

$$\begin{cases} u_0 = \omega_{00} + \omega_{01} x_1 + \omega_{02} x_2 \\ u_1 = \omega_{10} + \omega_{11} x_1 + \omega_{12} x_2 \\ u_2 = \omega_{20} + \omega_{21} x_1 + \omega_{22} x_2 \end{cases}$$

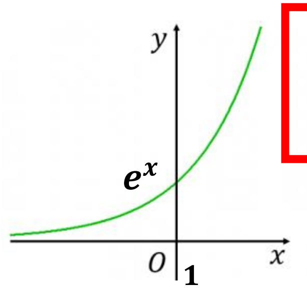
書き方を省略

$$\underset{\boldsymbol{u}}{\begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix}} = \underset{\boldsymbol{W}}{\boxed{\begin{pmatrix} \omega_{00} & \omega_{01} & \omega_{02} \\ \omega_{10} & \omega_{11} & \omega_{12} \\ \omega_{20} & \omega_{21} & \omega_{22} \end{pmatrix}}} \underset{\boldsymbol{x}}{\begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}}$$

変数を増やすと

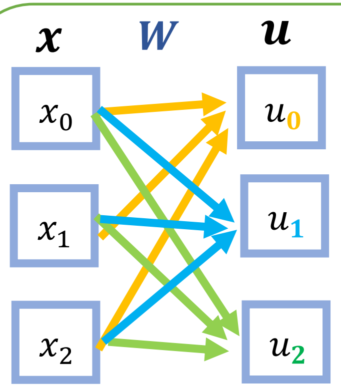
$$\begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_n \end{pmatrix} = \begin{pmatrix} \omega_{00} & \omega_{01} & \omega_{02} & \cdots & \omega_{0n} \\ \omega_{10} & \omega_{11} & \omega_{12} & \cdots & \omega_{1n} \\ \omega_{20} & \omega_{21} & \omega_{22} & \cdots & \omega_{2n} \\ \vdots & \vdots & & \ddots & \vdots \\ \omega_{n0} & \omega_{n1} & \omega_{n2} & \cdots & \omega_{nn} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

$exp(x)$ は e^x の事。



大きい値は、より大きく
小さい値は、より小さく

特徴が増幅される！



$$\frac{exp(u_0)}{exp(u_0) + exp(u_1) + exp(u_2)}$$

８５％！

$$\frac{exp(u_1)}{exp(u_0) + exp(u_1) + exp(u_2)}$$

１５％！

$$\frac{exp(u_2)}{exp(u_0) + exp(u_1) + exp(u_2)}$$

５％！

多値分類 3

$$\boldsymbol{u} = \boldsymbol{W}\boldsymbol{x}$$

$$\begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = \boxed{\begin{pmatrix} \omega_{00} & \omega_{01} & \omega_{02} \\ \omega_{10} & \omega_{11} & \omega_{12} \\ \omega_{20} & \omega_{21} & \omega_{22} \end{pmatrix}} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} \Leftarrow \begin{cases} u_0 = \omega_{00} + \omega_{01}x_1 + \omega_{02}x_2 \\ u_1 = \omega_{10} + \omega_{11}x_1 + \omega_{12}x_2 \\ u_2 = \omega_{20} + \omega_{21}x_1 + \omega_{22}x_2 \end{cases}$$

●SoftMax 関数

$$\boldsymbol{h(u_i)} = \frac{\boldsymbol{exp(u_i)}}{\sum_{i=0}^{2} \boldsymbol{exp(u_i)}} \qquad \boldsymbol{h(u_0)} = \frac{\boldsymbol{exp(u_0)}}{\boldsymbol{exp(u_0)} + \boldsymbol{exp(u_1)} + \boldsymbol{exp(u_2)}}$$

$$\boldsymbol{yp_i = h(u_i)} \quad , \quad \boldsymbol{yp_0 = h(u_0)}$$

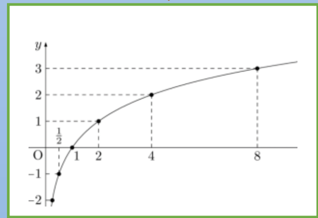
『$\log_2 x$』は、2 を何乗したら『x』になるかを示している。

$2^2 = 4 \Rightarrow \log_2 4 = 2$

$2^3 = 8 \Rightarrow \log_2 8 = 3$

$2^4 = 16 \Rightarrow \log_2 16 = 4$

$y = \log_2 x$



底が e の場合の表記

$\log_{\boxed{e}} x = \log x = \ln x$

●損失関数 L

『$\boldsymbol{yt_i \log(yp_i)}$』をエントロピーといい、熱力学の熱効率を計算する時に『熱が発散していく度合い』を表す指標として導入されたが、経済や情報などの分野でも応用された。

$$\boxed{\sum_{i=0}^{2} [yt_i \log(yp_i)]} = yt_0 \log(yp_0) + yt_1 \log(yp_1) + yt_2 \log(yp_2)$$

$yt_0 \log(yp_0) + yt_1 \log(yp_1) + yt_2 \log(yp_2)$

$yt_0 \log(yp_0) + yt_1 \log(yp_1) + yt_2 \log(yp_2)$

$$\frac{1}{M}\sum_{1}^{M}\left\{\boxed{\sum_{i=0}^{2}[yt_i \log(yp_i)]}\right\}_{M個目のデータ} = \frac{1}{M}\left[\left(\boxed{\sum_{i=0}^{2}[yt_i \log(yp_i)]}\right)_{1 個目のデータ} + \cdots + \left(\boxed{\sum_{i=0}^{2}[yt_i \log(yp_i)]}\right)_{M 個目のデータ}\right]$$

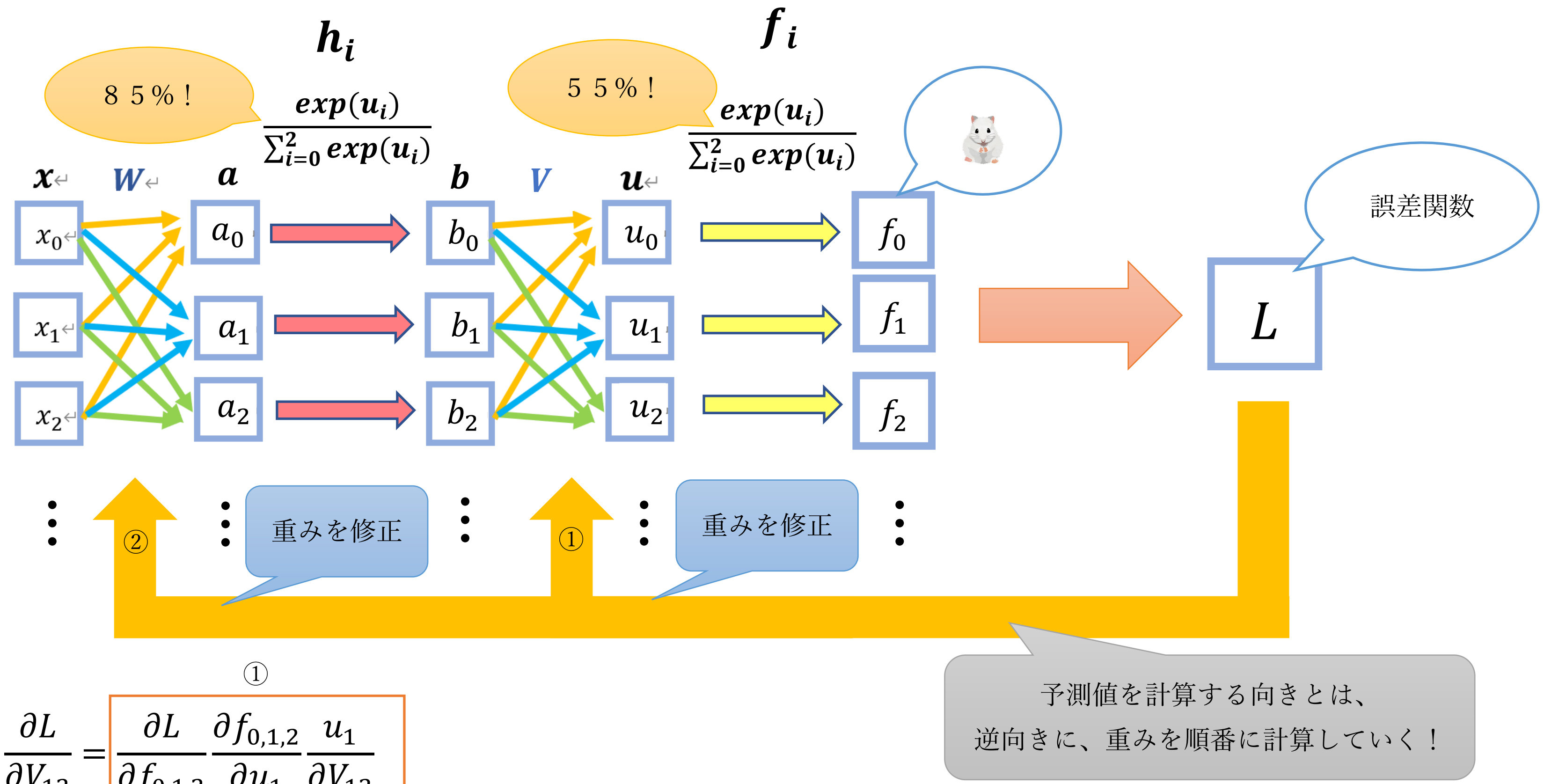
$$\frac{\partial L}{\partial \omega_{12}} = \frac{\partial L}{\partial yp_{0,1,2}} \frac{\partial yp_{0,1,2}}{\partial u_1} \frac{u_1}{\partial \omega_{12}}$$

計算していくと

長いので省略

$$\boxed{\omega_{ij}{}^{修正(k+1)回目}} = \omega_{ij}{}^{修正k回目} - \alpha \frac{1}{M}\left(\sum_{1}^{M}(\boldsymbol{yt_i} - \boldsymbol{yp_i})\, \boldsymbol{x_j}\right)_{M 個目のデータ}^{修正k回目}$$

ディープラーニングと誤差逆伝搬の概要



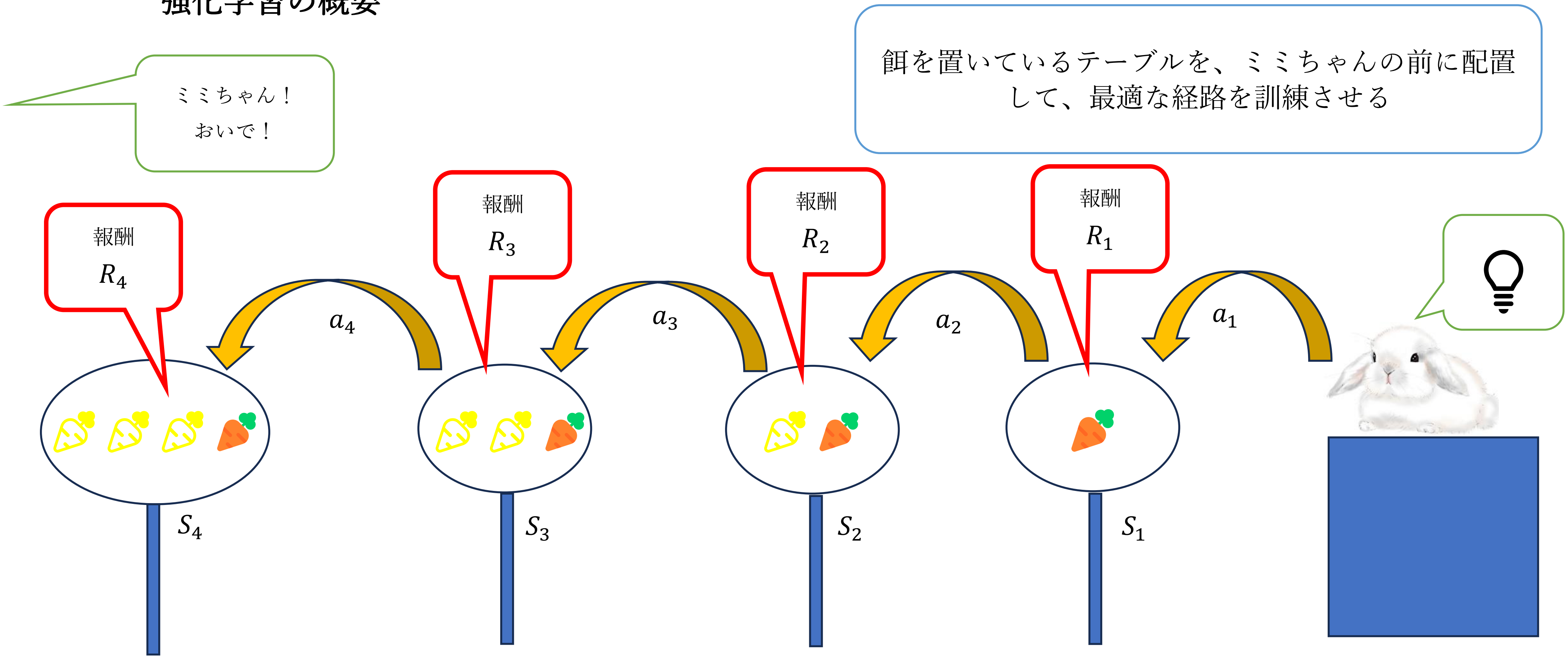
①

$$\frac{\partial L}{\partial V_{12}} = \boxed{\frac{\partial L}{\partial f_{0,1,2}} \frac{\partial f_{0,1,2}}{\partial u_1} \frac{u_1}{\partial V_{12}}}$$

② ①

$$\frac{\partial L}{\partial \omega_{12}} = \boxed{\frac{\partial L}{\partial f_{0,1,2}} \frac{\partial f_{0,1,2}}{\partial u_1} \frac{u_1}{\partial V_{12}}} \boxed{\frac{\partial V_{12}}{\partial b_{0,1,2}} \frac{\partial b_{0,1,2}}{\partial a_2} \frac{\partial a_2}{\partial \omega_{12}}}$$

強化学習の概要



Status： 状態
Action： 行動
Reward： 報酬

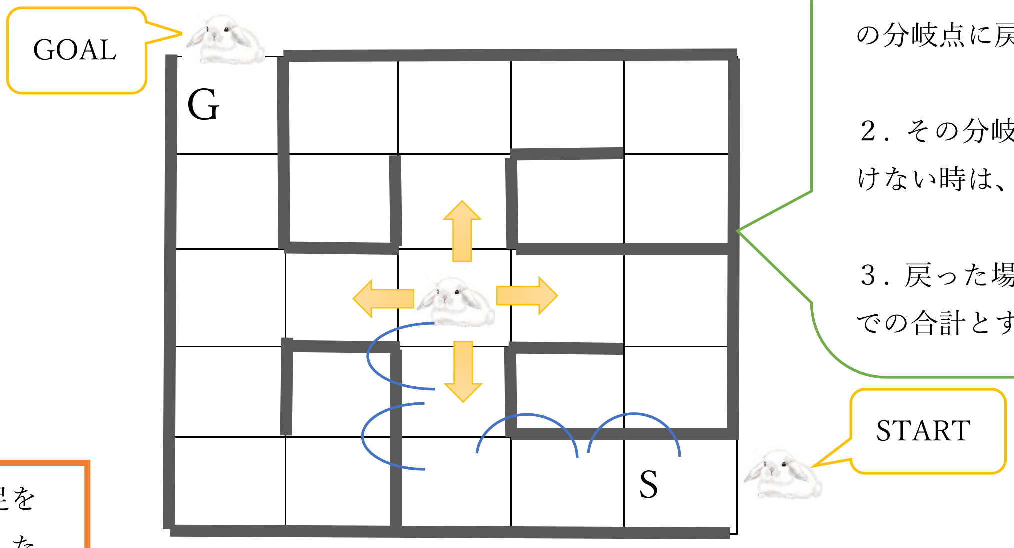
$R_1(S_1) =$　　　　$R_2(S_2) = \quad + 1$　　　　$R_3(S_3) = \quad + 2$　　　　$R_4(S_4) = \quad + 3$

色々試してみて、たどった経路の中で、<u>一番報酬を獲得した時の方策を採用する</u>

強化学習の概要２（状態価値関数V）

割引率
$0 \leq \gamma \leq 1$
報酬は無限に増加するので、
報酬の増加量を調整する。

$$V^{\pi}(s_t) = r_{t+1} + \boxed{\gamma}\, V^{\pi}(S_{t+1})$$



１．来た道を引き返すと行き止まりになる前の分岐点に戻る。

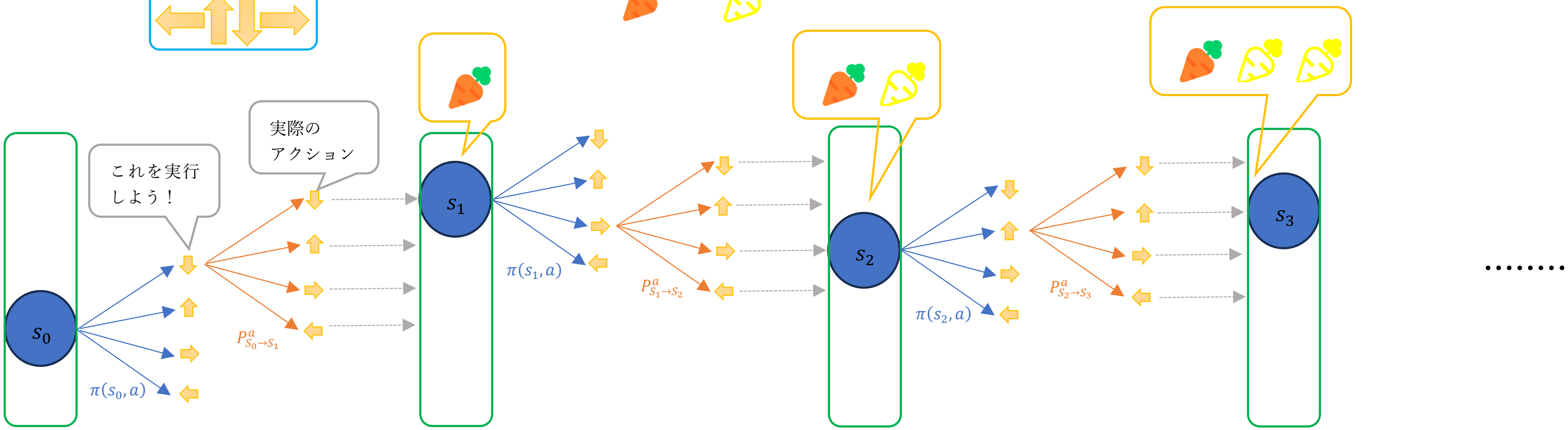
２．その分岐点から、来た道に戻らないといけない時は、さらにその前の分岐点に戻る。

３．戻った場合の報酬の合計は、戻った場所での合計とする。

各行動に対する頻度

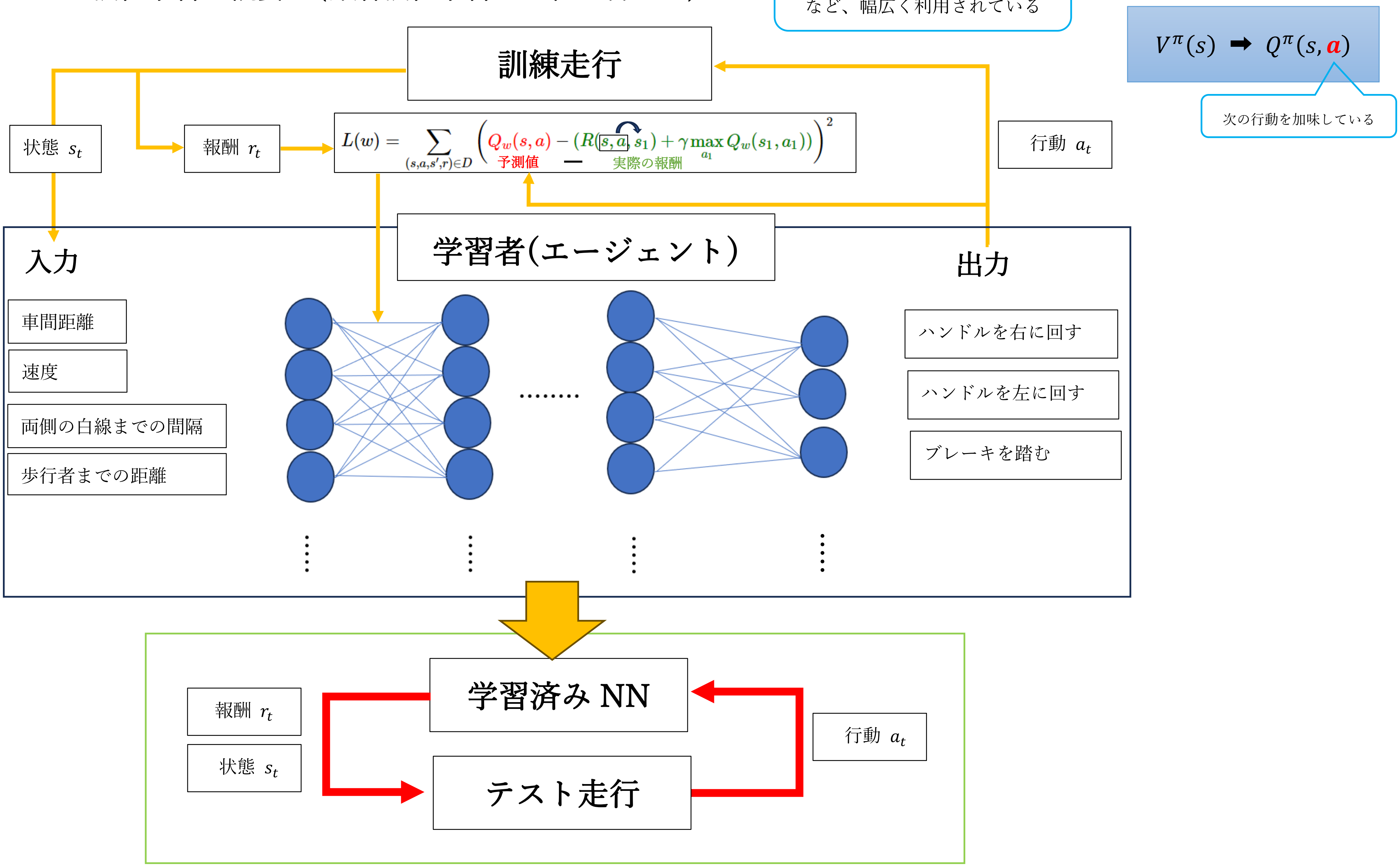
千鳥足を考慮した成功率

$$V^{\pi}(S) = \sum_{a=行動} \pi(s,a) \sum_{S'=次の位置} P^{a}_{S\to S'} \left[R^{a}_{S\to S'} + \gamma V^{\pi}(S') \right]$$



強化学習の概要 3（深層強化学習との組み合わせ）

ChatGPT や自動走行、囲碁 AI など、幅広く利用されている



k-means 法

0. データ数は n 個ある。

　　$i = 1, 2, \ldots, n$

1. クラスター中心となる点 Vg(重心点)を配置する個数 k を決める。

　　　$1, 2, \ldots, k$

2. k 個のクラスター中心となる点をランダムに配置する。

3. それぞれの座標P_i（x_i, y_i)から、それぞれのクラスター中心点 V_g（X_k, Y_k)の座標までの距離を求める。

$$\|P_i - V_g\|^2 = (x_i - X_k)^2 + (y_i - Y_k)^2$$

4. 『3』で求めた距離の中で、一番近い中心に紐づける。
　他の座標P_iでも、同様にする。

minimum(最小値)の略

$$arg\ \min\|P_i - V_g\|^2 = \arg\min\|(x_i + X_k)^2 + (y_i + Y_k)^2\|$$

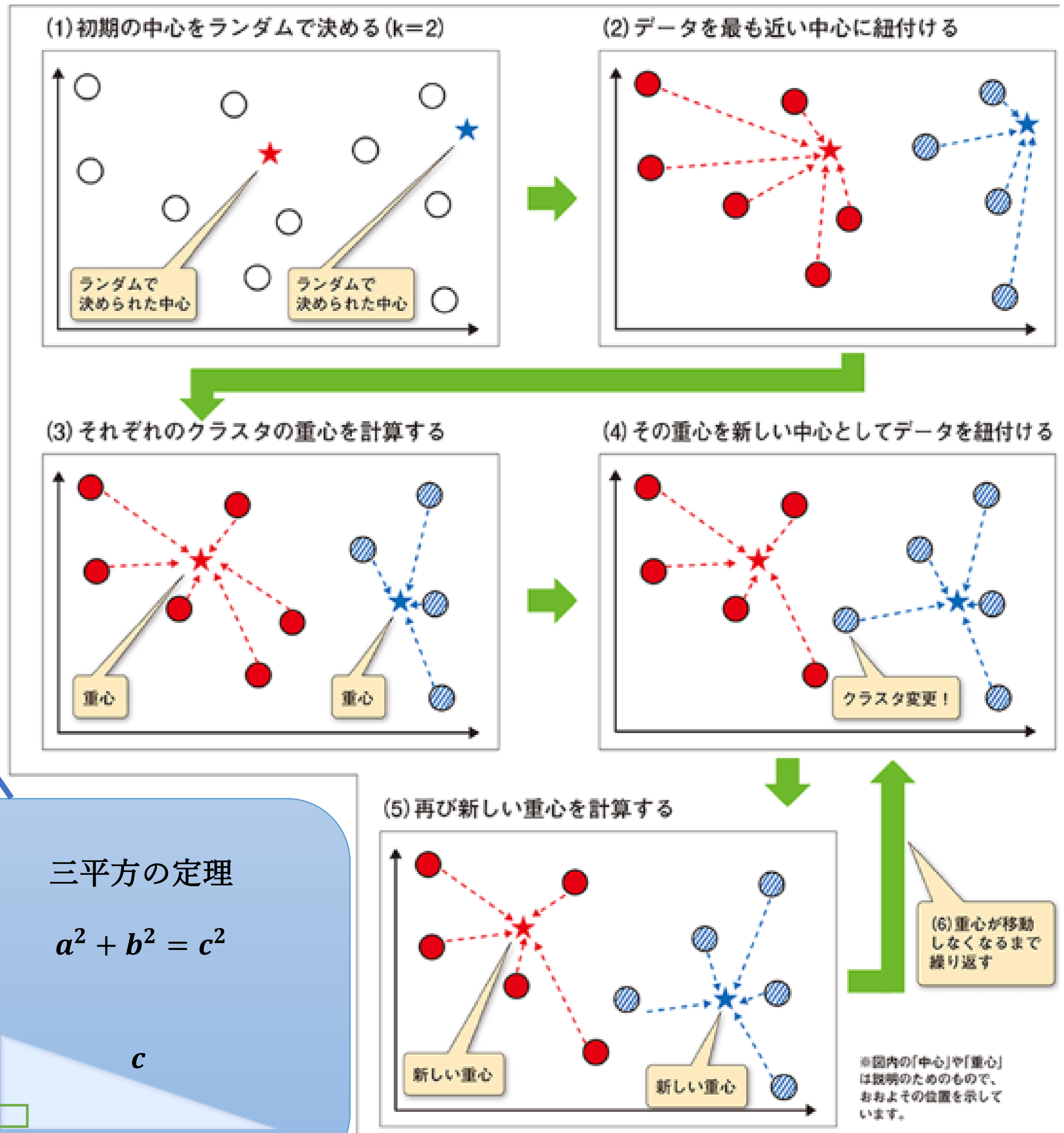
arg[条件式] で、『**条件式を満たすモノ**』と云う意味
argument；　議論、主張→ **論拠** →引数、偏角

5. 『4』で分類されたグループの重心となる、座標を見つける。

クラスターの x 座標の平均値$= \dfrac{x_1 + x_2 + x_3 + \cdots + x_n}{N}$

クラスターの y 座標の平均値$= \dfrac{y_1 + y_2 + y_3 + \cdots + y_n}{N}$

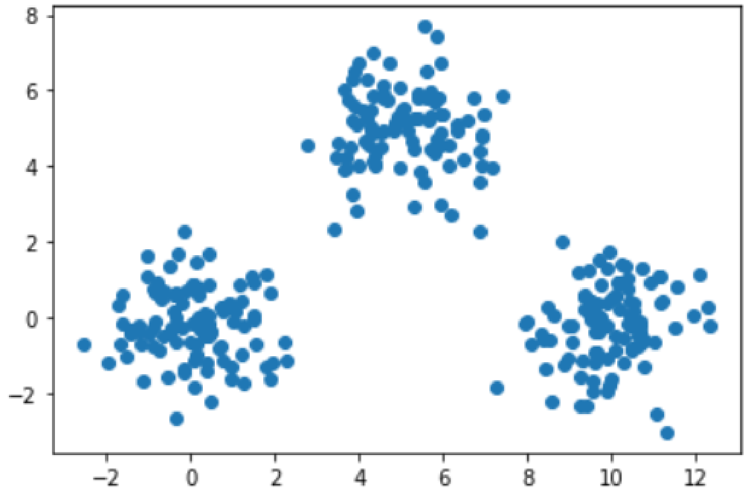
6. クラスター中心を『5』で求めた位置に移動する。
7. クラスター中心が移動しなくなるまで、『3』『4』『5』『6』を繰り返し行う。



三平方の定理

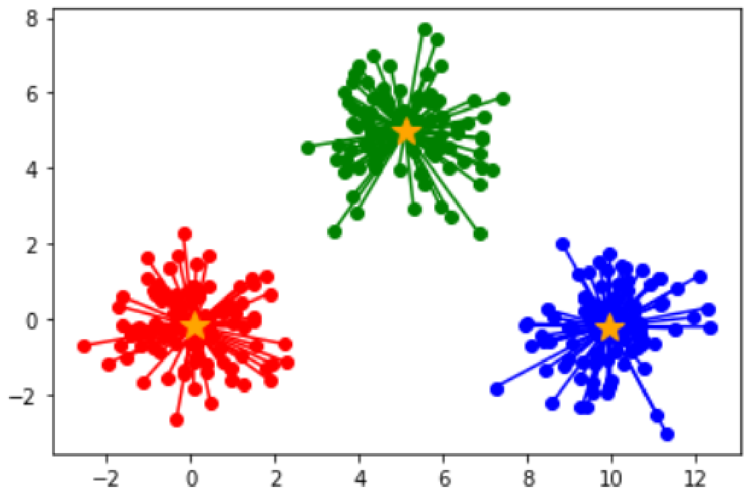
$$\boldsymbol{a^2 + b^2 = c^2}$$

a
c
b

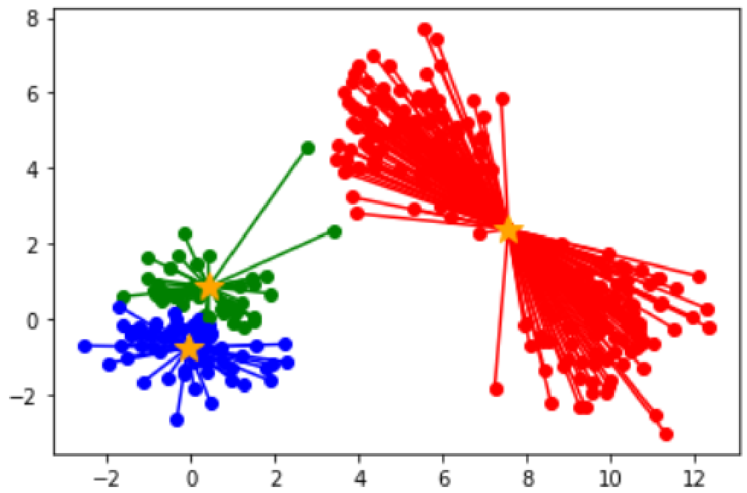


以下のように初期値によって、クラスタリングの結果が異なります。

(収束までの計算量も異なります)



良い初期値を利用した場合



悪い初期値を利用した場合

k-means++ 法

0. データ数は i 個ある。いくつのクラスターを作成するか決める。

$$i = 1, 2, \ldots, n \qquad , \qquad クラスター数 = 1, 2, \ldots, k$$

1. クラスター中心を以下の式に従って算出し、決定してくいく。
   データ同士の距離が遠い点をクラスター中心としていく。これを k 個揃うまで繰り返す。
   『下記の確率の式が高い点』＝『点と点の距離が遠い点』

$$\frac{D(x_i)}{\sum_1^n D(x_i)} = \frac{距離A}{距離A + 距離B + 距離C + 距離D}$$

2. 『1』でクラスター中心を k 個決定した後、そのクラスター中心をもとに
   【K-means】の処理を実行する

●k-means 法

1. それぞれの座標P_i（x_i, y_i)から、それぞれのクラスター中心点 Vg(X_k, Y_k)の座標
   までの距離を求める。

$$\|P_i - V_g\|^2 = (x_i + X_k)^2 + (y_i + Y_k)^2$$

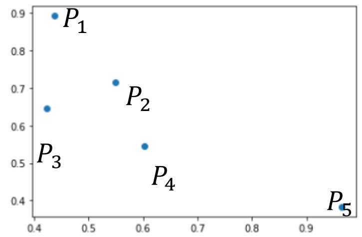
2. 『1』で求めた距離の中で、一番近いモノだけ線を引く。
   他の座標P_iでも、同様にする。

$$arg\ \min\|P_i - V_g\|^2 = \arg\min\|(x_i + X_k)^2 + (y_i + Y_k)^2\|$$

3. 『2』で分類されたグループの重心となる、座標を見つける。

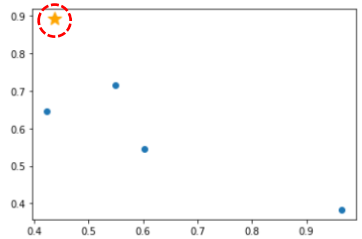
$$クラスター中心のx座標 = \frac{x_1 + x_2 + x_3 + \cdots + x_n}{N} \qquad クラスター中心のy座標 = \frac{y_1 + y_2 + y_3 + \cdots + y_n}{N}$$

4. クラスター中心を『3』で求めた位置に移動する。
5. クラスター中心が移動しなくなるまで、『1』『2』『3』『4』を繰り返し行う。

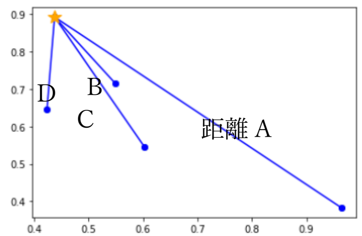


0. このデータ点を例に k-means++ を説明

$P_1(x_1, y_1)$
$P_2(x_2, y_2)$
$P_3(x_3, y_3)$
$P_4(x_4, y_4)$
$P_5(x_5, y_5)$

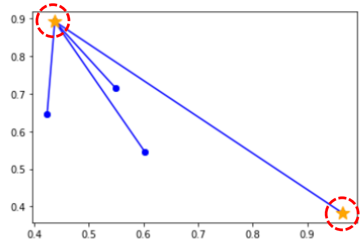


1. [データ点] からランダムに 1 つ選び、[クラスタ中心] とする



2. [データ点] と [一番近いクラスタ中心] の距離を求める

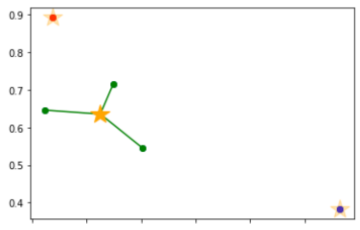
※複数の [クラスタ中心] が存在する場合は、一番近いものを 1 つ選択



3. 距離が遠いデータ点を選ぶ (確率が高い)

※各データ点が選ばれる確率は以下

(データ点の距離)^2/(各データ点の距離の合計)^2



4. k 個の [クラスタ中心] を選ぶまで、2, 3 を繰り返す

5. k-means 法で k 個のクラスタリングを行う
※ここからの手順は k-means と k-means++ で同じで