『データサイエンスの考え方
―社会に役立つ AI × データ活用のために―』

（小澤 誠一・齋藤 政彦　共編）

問の解答

◆注意◆

- 一部の計算問題を除き，以下は，解答の一例を示したものです．
- 本解答例は，「著作権法」によって，著作権等の権利が保護されています．本書の複製権・翻訳権・上映権・譲渡権・公衆送信権（送信可能化権を含む）は著作権者が保有しています。本解答例の全部または一部につき無断で転載，複写複製，電子的装置への入力等をされると，著作権等の侵害となる場合がありますので，ご注意ください．

第 2 章　アルゴリズムとデータ構造

問 2.1　アルゴリズム 2.5 の計算量は，4 行目の計算の回数（繰り返し回数）から算出できる．繰り返し回数は $p \times r \times q$ であることから，計算量のオーダーは $O(pqr)$ となる．

問 2.2　2 行 2 列の行列を 2 次元配列 A で表し，その行列式を求めるアルゴリズムを次に示す．

アルゴリズム 1 ■ 2 行 2 列の行列の行列式を求める

Input: 2 次元配列 A

Output: 配列 A で表された 2 行 2 列の行列の行列式

1: **return** $A[0][0] * A[1][1] - A[0][1] * A[1][0]$

また，このアルゴリズムの計算量のオーダーは $O(1)$ である．

問 2.3　動作例を右に示す．なお，下線部はデータを比較する箇所，そのうち二重下線部は比較の結果データを交換する箇所である．

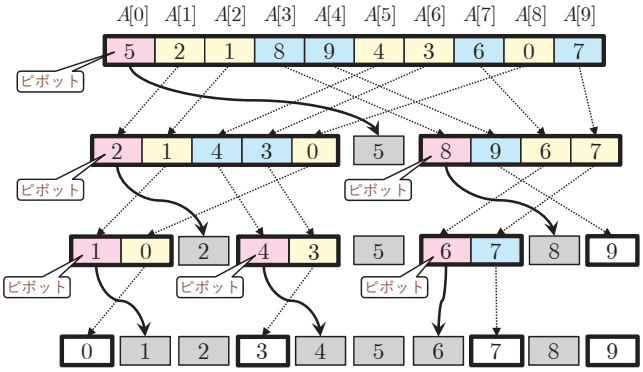
5 2 1 8 9 4 3 6 0 7
2 5 1 8 9 4 3 6 0 7
2 1 5 8 9 4 3 6 0 7
2 1 5 8 9 4 3 6 0 7
2 1 5 8 9 4 3 6 0 7
2 1 5 8 4 9 3 6 0 7
2 1 5 8 4 3 9 6 0 7
2 1 5 8 4 3 6 9 0 7
2 1 5 8 4 3 6 0 9 7
2 1 5 8 4 3 6 0 7 **9**
1 2 5 8 4 3 6 0 7 **9**
1 2 5 8 4 3 6 0 7 **9**
1 2 5 8 4 3 6 0 7 **9**
1 2 5 4 8 3 6 0 7 **9**
1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

問 2.4　アルゴリズム 2.8 の計算量は，3 行目のデータ比較の回数（繰り返し回数）から算出できる．繰り返し回数は

$$(N-1)+(N-2)+\cdots+1=\frac{(N-1)N}{2}=\frac{1}{2}N^2-\frac{1}{2}N$$

であることから，計算量のオーダーは $O(N^2)$ となる．

問 2.5　動作例を次に示す．なお，この解答例は図 2.11 に準拠したものであり，実際にアルゴリズム 2.9 に従って実行した場合には，ピボットの前後にデータを移動させる方法が異なるため，これとは違った動作をすることに留意のこと．



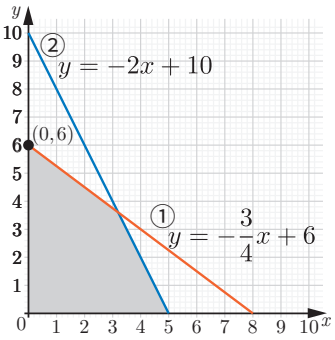
第 3 章　システム最適化

問 3.1　標準化した制約式 (3.16)，(3.17) に非基底変数 $x = 0$，$\lambda_1 = 0$ を代入すると，

$$
\begin{aligned}
& 3x + 4y + \lambda_1 = 24 \\
\Rightarrow\ & 4y = 24 \\
\Rightarrow\ & y = 6 \\
& 2x + y + \lambda_2 = 10 \\
\Rightarrow\ & y + \lambda_2 = 10 \\
& y = 6 \text{ を代入すると} \\
\Rightarrow\ & \lambda_2 = 4
\end{aligned}
$$

以上より，$x = 0$，$y = 6$，$\lambda_1 = 0$，$\lambda_2 = 4$ となり，実行可能基底解 $(x,\ y) = (0,\ 6)$ を得る．

図 3.2(a) に，$(x,\ y) = (0,\ 6)$ をプロットすると，次のようになる．



問 3.2　目的関数の式 (3.22) に，$(x,\ y) = (3.2,\ 3.6) = \left(\dfrac{16}{5}, \dfrac{18}{5}\right)$ を代入すると，

$$
\begin{aligned}
z &= (6 - x)x + \left(\frac{35}{3} - y\right) y \\
&= \left(6 - \frac{16}{5}\right) \times \frac{16}{5} + \left(\frac{35}{3} - \frac{18}{5}\right) \times \frac{18}{5} \\
&= 38
\end{aligned}
$$

となる．非線形計画問題の最適値（$z = 40.25$）よりも，目的関数の値が 2.25 だけ小さい解が得られた．

問 3.3　目的関数の (3.8) に，$(x,\ y) = (3,\ 3)$ を代入すると，

$$
\begin{aligned}
z &= 24x + 16y \\
&= 24 \times 3 + 16 \times 3 \\
&= 120
\end{aligned}
$$

となる．整数計画問題の最適値（$z = 128$）よりも，目的関数の値が 8 だけ小さい解が得られた．

第 5 章　教師なし学習

問 5.1

(1) K-平均法におけるクラスタ中心の学習は，クラスタ中心をランダムに配置することから始まり，そこからクラスタへの割り当てとクラスタ中心の更新を繰り返す．そこで，クラスタ中心が更新されるにつれてその損失関数の値が小さくなる理由を考える．そのためには，具体例を図を描きながら考えるとよい．例として，図 1 にあるような平面上の 12 個の点からなるデータを考える．ここで，これら平面上の点にそれぞれ $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{12}$ と名付け，このデータを 2 つのクラスタにクラスタリングする．

まずは，K-平均法のアルゴリズムが具体的にどのようなものか図を示しながら解説する．ただし，今回は 2 つのクラスタに分けるので $K = 2$ である．

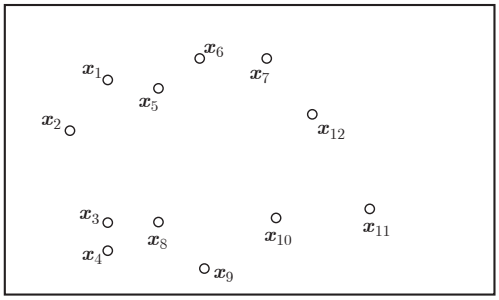


図 1 ■ 平面上の 12 個の点

[Step 1] このアルゴリズムは，クラスタ中心をランダムに配置することから始まる．今回は $K = 2$ なので，2 個のクラスタ中心を配置する．例えばランダムに配置した結果，図 2 左のようになったとする．ただし，図の星型の点がクラスタ中心である．このクラスタ中心に $\boldsymbol{c}_1, \boldsymbol{c}_2$ とそれぞれ名付けておく．

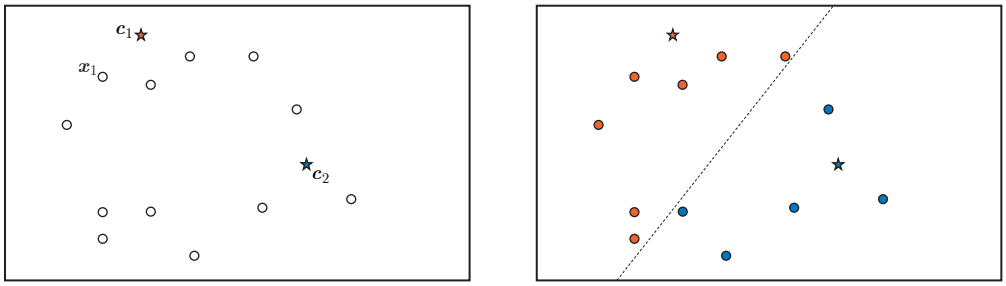


図 2 ■ $\boldsymbol{c}_1, \boldsymbol{c}_2$ によるクラスタリング

[Step 2] クラスタ中心が決まれば，それに応じて，$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{12}$ の各点をクラスタへ割り当てることができる．クラスタ中心 $\boldsymbol{c}_1, \boldsymbol{c}_2$ に対応するクラスタをそれぞれ C_1, C_2 と名付ける．例えば，$\boldsymbol{x}_1$ は C_1, C_2 どちらのクラスタに割り当てられるか？　この問いに答えるためにはまず，「$\boldsymbol{x}_1$ に最も近いクラスタ中心は $\boldsymbol{c}_1, \boldsymbol{c}_2$ どちらか？」を考える．

図 2 左を見れば，$\boldsymbol{c}_1$ の方が $\boldsymbol{c}_2$ よりも近そうである．この場合，$\boldsymbol{x}_1$ は $\boldsymbol{c}_1$ に対応するクラスタ C_1 に割り当てられる．同様にして，他の点についてもクラスタへの割り当てを行う．今回はクラスタ中心は 2 点なので，この 2 点に関する垂直二等分線を引けば，各点がどちらのクラスタに属するかがすぐにわかる (図 2 右の破線)．このクラスタリングの結果，$\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_7\}$ が C_1 に属し，$\{\boldsymbol{x}_8, \boldsymbol{x}_9 \ldots, \boldsymbol{x}_{12}\}$ が C_2 に属することになる．図 2 右では，クラスタ C_1, C_2 いずれに属するかで点の色を分けている．

[Step 3] 割り当てが終われば，次はクラスタ中心の更新を行う．まず $\boldsymbol{c}_1$ の更新を考える．新しいクラスタ中心は，C_1 に属する点の平均をとることで得られる．ここで，平面上に xy 座標を導入し，点をベクトルとして考える．つまり今回，C_1 に属している点は $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_7$ の 7 点なので，これらのベクトルの和をとり，その後 $\dfrac{1}{7}$ 倍することで，新しいクラスタ中心が得られる．この新しいクラスタ中心を $\boldsymbol{c}_1'$ とする．このクラスタ中心の更新を数式で表すと，次のようになる．

$$\boldsymbol{c}_1' = \frac{1}{7}\left(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \cdots + \boldsymbol{x}_7\right)$$

次に $\boldsymbol{c}_2$ の更新を考える．C_2 に属している点は $\boldsymbol{x}_8, \boldsymbol{x}_9, \ldots, \boldsymbol{x}_{12}$ の 5 点であるから，新しいクラスタ中心を $\boldsymbol{c}_2'$ とすると，$\boldsymbol{c}_2'$ は次の数式によって求めることができる．

$$\boldsymbol{c}_2' = \frac{1}{5}\left(\boldsymbol{x}_8 + \boldsymbol{x}_9 + \cdots + \boldsymbol{x}_{12}\right)$$

図 3 左では更新前のクラスタ中心が，右では更新後のクラスタ中心が描かれている．

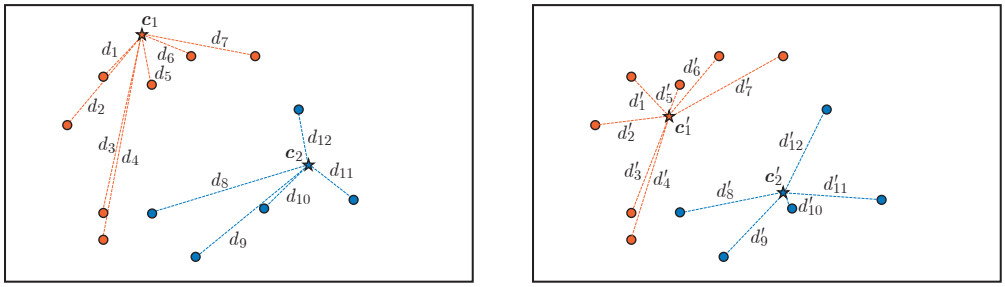


図 3 ■ クラスタ中心の更新

[Step 4] クラスタ中心が更新されたので，次は，それに応じて各点のクラスタへの割り当てを再び行う．新しいクラスタ中心 $\boldsymbol{c}_1', \boldsymbol{c}_2'$ に対応するクラスタをそれぞれ C_1', C_2' と名付ける．点 $\boldsymbol{c}_1', \boldsymbol{c}_2'$ に関する垂直二等分線（図 4 左の破線）を引くことで，各点がどちらのクラスタに属するかがわかる．点 $\boldsymbol{x}_8$ は，以前はクラスタ C_2 に属していた（図 4 左）が，新しいクラスタリングでは，点 $\boldsymbol{x}_8$ の割り当てはクラスタ C_1' になり，割り当てが変化している（図 4 右）．

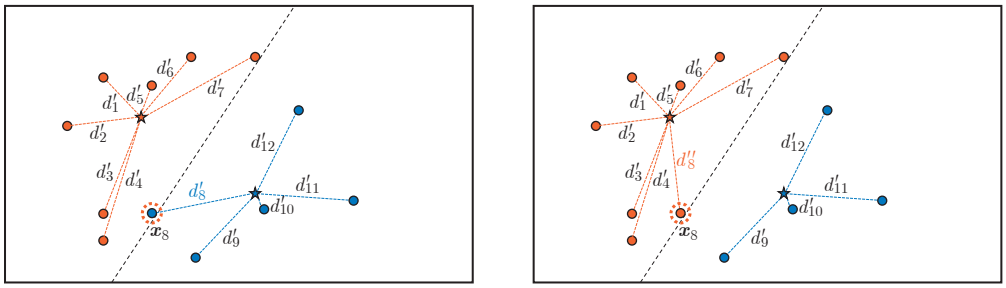


図 4 ■ $\boldsymbol{c}_1', \boldsymbol{c}_2'$ によるクラスタリング

[Step 5] クラスタの割り当てが変わったので，再びクラスタの中心の更新を行う．これが完了すれば，更新されたクラスタ中心に関するクラスタへの割り当てを行う．クラスタの割り当てが変化しなくなるまで，この操作を繰り返す．

以上が，K-平均法のアルゴリズムの解説である．

ここから，本題である，この更新で損失関数の値は小さくなっているかどうか，について考える．まず，更新前のクラスタ中心である $\boldsymbol{c}_1, \boldsymbol{c}_2$ の損失関数の値 $J(\boldsymbol{c}_1, \boldsymbol{c}_2)$ は次のようになる．

$$J(\boldsymbol{c}_1, \boldsymbol{c}_2) = d_1^2 + d_2^2 + \cdots + d_{12}^2$$

ただし，$d_1, d_2, \ldots, d_{12}$ は図 3 左に示すクラスタ中心と点との間の距離を表す．

更新後のクラスタ中心である $\boldsymbol{c}'_1, \boldsymbol{c}'_2$ の損失関数の値 $J(\boldsymbol{c}'_1, \boldsymbol{c}'_2)$ は，次のようになる．

$$J(\boldsymbol{c}'_1, \boldsymbol{c}'_2) = (d'_1)^2 + (d'_2)^2 + \cdots + (d'_7)^2 + (d''_8)^2 + (d'_9)^2 + \cdots + (d'_{12})^2$$

ただし，$d'_1, d'_2, \ldots, d''_8, \ldots, d'_{12}$ は図 4 右に示すクラスタ中心と点との間の距離を表す．

ここでは，次の不等式を確かめることが目標になる．

$$J(\boldsymbol{c}_1, \boldsymbol{c}_2) > J(\boldsymbol{c}'_1, \boldsymbol{c}'_2)$$

この不等式の成立は，次のようにして確かめることができる．

$$\begin{aligned} J(\boldsymbol{c}_1, \boldsymbol{c}_2) &= d_1^2 + d_2^2 + \cdots + d_{12}^2 \\ &> (d'_1)^2 + (d'_2)^2 + \cdots + (d'_8)^2 + \cdots + (d'_{12})^2 \\ &> (d'_1)^2 + (d'_1)^2 + \cdots + (d''_8)^2 + \cdots + (d'_{12})^2 \\ &= J(\boldsymbol{c}'_1, \boldsymbol{c}'_2) \end{aligned}$$

1 つ目の不等号の成立は，次のように説明される．図 3 左には，$d_1, d_2, \ldots, d_{12}$ が，各破線の長さとして書き込まれ，図 3 右には，$d'_1, d'_2, \ldots, d'_{12}$ が，各破線の長さとして書き込まれている．これらの図を見比べると，

$$d_1^2 + d_2^2 + \cdots + d_{12}^2 > (d'_1)^2 + (d'_2)^2 + \cdots + (d'_{12})^2$$

であることが直感的に読み取れる．

次に，2 つ目の不等号の成立は，次のように説明される．左辺の d'_8 が右辺では d''_8 に置き換わっていることに注意してほしい．d'_8 は $\boldsymbol{x}_8$ と $\boldsymbol{c}'_2$ の間の距離を意味し，d''_8 は $\boldsymbol{x}_8$ と $\boldsymbol{c}'_1$ の間の距離を意味する．図 4 では，$\boldsymbol{x}_8$ は破線で表された垂直二等分線の左側にあるが，これは，$\boldsymbol{x}_8$ が $d'_8 > d''_8$ を満たす位置にあることを意味する．このことから

$$(d'_1)^2 + (d'_2)^2 + \cdots + (d'_8)^2 + \cdots + (d'_{12})^2 > (d'_1)^2 + (d'_1)^2 + \cdots + (d''_8)^2 + \cdots + (d'_{12})^2$$

であることがわかる．

以上より，$J(\boldsymbol{c}_1, \boldsymbol{c}_2) > J(\boldsymbol{c}'_1, \boldsymbol{c}'_2)$ が示され，クラスタ中心の更新によって損失関数の値は小さくなることがわかる．

(2) 問題の図 5.4 のような顧客データの散布図には，図 5 のように，大きく 3 つのグループがあるように見える．説明をわかりやすくするために，図 5.4 のデータを表にしたものが，表 1 である．ただし，顧客にはそれぞれ A さん，B さん，... と名付けた．図 5 では，I さん，J さん，C さんの表す点を吹き出しで示したが，I さんと J さんは近く，I さんと C さんは遠く離れているように見える．$K = 3$ とした K-平均法によって，このデータをうまくクラスタリングすることができるか？　実際，K-平均法でのクラスタリングの結果は，図 6 のようになる．このクラスタリングでは，I さんと C さんが同じクラスタに入り，J さんは I さんと異なるクラスタに属している．これは直感と合わないと思う人もいるだろうが，なぜそのようになったのか，以下で考える．

表 1 ■ ある企業の顧客データ

	A さん	B さん	C さん	D さん	E さん	F さん
購入回数	20	18	17	18	16	14
購入金額	1100	4000	4500	1000	1700	250
	G さん	H さん	I さん	J さん	K さん	L さん
購入回数	13	11	7	6	3	2
購入金額	500	1500	5000	3500	3000	4000

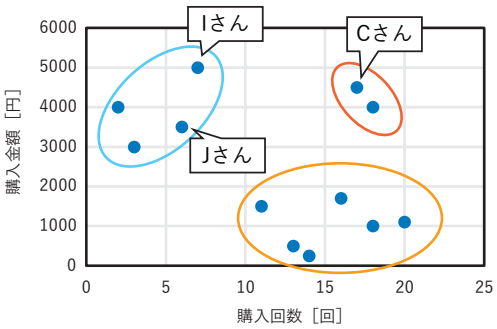


図 5 ■ 直感的なクラスタリング

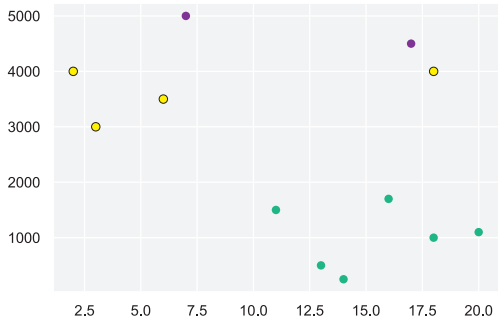


図 6 ■ K-平均法によるクラスタリング

ポイントは，購入回数のばらつきと購入金額のばらつきに，大きな差があるということである．実際，購入回数の標準偏差はおおよそ 5.95 だが，購入金額の標準偏差はおおよそ 1605.00 であり，ばらつきが大きく異なる．K-平均法では，2 点の間の距離を測り，この距離が小さいと，その 2 点は同じクラスタに属しやすくなる．そこで，図 7 左に着目すると，視覚的には I さんと C さんは遠くにあるように見えるが，実際に距離を測ると，おおよそ 500 であることがわかる．一方，視覚的には I さんと J さんは近くにあるように見えるが，実際に距離を測ると，おおよそ 1500 であることがわかる．したがって（直感と反して），I さんと C さんは近く，I さんと J さんは遠いということになる．縦軸の数値は，横軸の数値に比べて大きくばらついている．散布図を，縦軸と横軸のスケールを合わせて書くと，縦長の図になってしまうため，縦軸のスケールを大きくすることで，図が縦長にならないようにしている．しかし，縦軸と横軸のスケールが異なることによって，視覚的に近く見えても実際の距離は離れているということが起こり得るのである．

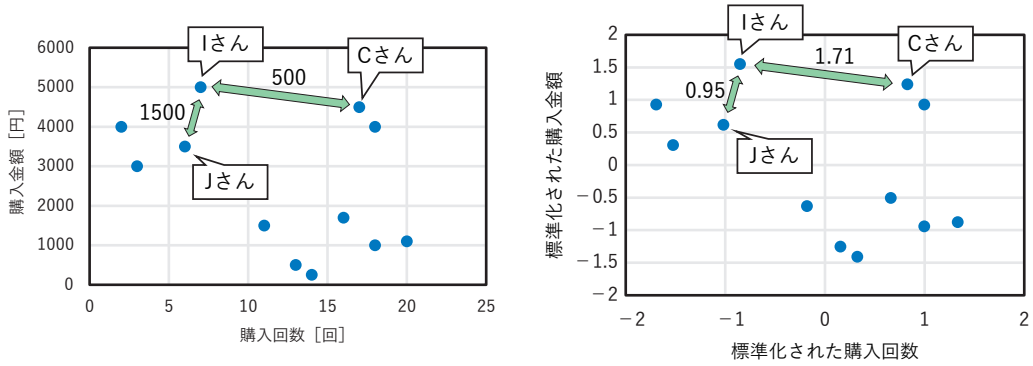


図 7 ■ I さんと C さん，I さんと J さんの間の距離

この視覚的な直感に合うようなクラスタリングを行うためには，例えば購入回数と購入金額の標準化を行ってから，K-平均法を実行することが考えられる．

購入回数と購入金額の標準化を行った結果は，表 2 のようになる．購入回数の標準化は，各購入回数の数値から購入回数の平均を引き，それを購入回数の標準偏差で割ることで計算できる．また，購入金額の標準化は，各購入金額の数値から購入金額の平均を引き，それを購入金額の標準偏差で割ることで計算できる．この標準化した後のデータの散布図が，図 7 右である．実際に I さんと C さんの間の距離を測ると，おおよそ 1.75 であることがわかり，一方，I さんと J さんの間の距離を測ると，おおよそ 0.95 であることがわかる．標準化された後では（直感どおり），I さんと C さんは遠く, I さんと J さんは近いということになる．

表 2 ■ 標準化されたデータ

	A さん	B さん	C さん	D さん	E さん	F さん
標準化された購入回数	1.33	0.99	0.82	0.99	0.65	0.32
標準化された購入金額	−0.87	0.93	1.24	−0.93	−0.50	−1.4
	G さん	H さん	I さん	J さん	K さん	L さん
標準化された購入回数	0.15	−0.18	−0.85	−1.02	−1.52	−1.69
標準化された購入金額	−1.24	−0.62	1.55	0.62	0.30	0.93

問 5.2 対数尤度関数を最大化する $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ が尤度関数も最大化する理由は，対数関数 $y = \ln(x)$ が単調増加関数だからである．単調増加関数であることから，$y_1 \leq y_2$ ならば，$\ln(y_1) \leq \ln(y_2)$ であり，逆に $\ln(y_1) \leq \ln(y_2)$ であれば，$y_1 \leq y_2$ である．

ここで，対数尤度関数を $f(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ と表すことにする．ガウスモデルの場合は

$$f(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = N(\boldsymbol{x}_1 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \times N(\boldsymbol{x}_2 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \times \cdots \times N(\boldsymbol{x}_N \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

となり，対数尤度関数は $\ln(f(\boldsymbol{\mu}, \boldsymbol{\Sigma}))$ である．もし，$\ln(f(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)) \leq \ln(f(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2))$ であれば，$f(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \leq f(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ が成り立つ．ここで, 対数尤度関数を最大化するパラメータを $\boldsymbol{\mu}_{\max}, \boldsymbol{\Sigma}_{\max}$ とおくと，任意のパラメータ $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ に対して

$$\ln(f(\boldsymbol{\mu}, \boldsymbol{\Sigma})) \leq \ln(f(\boldsymbol{\mu}_{\max}, \boldsymbol{\Sigma}_{\max}))$$

が成り立ち，これから

$$f(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \leq f(\boldsymbol{\mu}_{\max}, \boldsymbol{\Sigma}_{\max})$$

がわかる．よって $\boldsymbol{\mu}_{\max}, \boldsymbol{\Sigma}_{\max}$ は尤度関数も最大化することがわかる．

次に，対数尤度関数のメリットについて考える．尤度関数は

$$f(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = N(\boldsymbol{x}_1 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \times N(\boldsymbol{x}_2 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \times \cdots \times N(\boldsymbol{x}_N \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

のように積の形であるが，対数尤度関数にすることで積の形が和の形に変わる．

$$\begin{aligned}\ln(f(\boldsymbol{\mu}, \boldsymbol{\Sigma})) &= \ln(N(\boldsymbol{x}_1 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \times N(\boldsymbol{x}_2 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) \times \cdots \times N(\boldsymbol{x}_N \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})) \\ &= \ln(N(\boldsymbol{x}_1 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})) + \ln(N(\boldsymbol{x}_2 \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})) + \cdots + \ln(N(\boldsymbol{x}_N \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}))\end{aligned}$$

尤度関数または対数尤度関数を最大化したい場合，よく用いられるのは，この関数の導関数（微分）を考えることである．積の形をした $f(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ を微分するよりも，和の形をした $\ln(f(\boldsymbol{\mu}, \boldsymbol{\Sigma}))$ を微分する方が計算は簡単であり，これが 1 つ目のメリットである．しかし，ガウスモデルを考える場合，さらなるメリットがある．そこでは，対数関数 $y = \ln(x)$ が指数関数 $y = \exp(x)$ の逆関数であること，つまり $\ln(\exp(x)) = x$ が成り立つことがポイントになる．これを使えば，$\ln(N(\boldsymbol{x}_n \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}))$ $(n = 1, 2, \ldots, N)$ は次のように簡単な形になる．

$$
\begin{aligned}
&\ln\left(\frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}}\exp\left(-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})\right)\right)\\
=&\ln\left(\frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}}\right)+\ln\left(\exp\left(-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})\right)\right)\\
=&-\ln(2\pi)-\frac{1}{2}\ln(|\boldsymbol{\Sigma}|)-\frac{1}{2}(\boldsymbol{x}_n-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_n-\boldsymbol{\mu})
\end{aligned}
$$

最後の等号において，対数関数 $y=\ln(x)$ が指数関数 $y=\exp(x)$ の逆関数であることを用いた．この形になることにより，微分の計算がさらに簡単になる．

問 5.3 p.94 の図 5.11 の分布では，2 つの三日月形のクラスタがあることが視覚的に読み取れる．実際に，混合ガウス分布を用いたクラスタリングは，視覚的に読み取ったクラスタリングと一致するだろうか？

実際に混合ガウスモデルを使ってクラスタリングを行った結果は，図 8 のようになる．クラスタは色によって区別している．この結果から，混合ガウス分布を用いたクラスタリングは，視覚的に読み取ったクラスタリングと一致しないことがわかる．

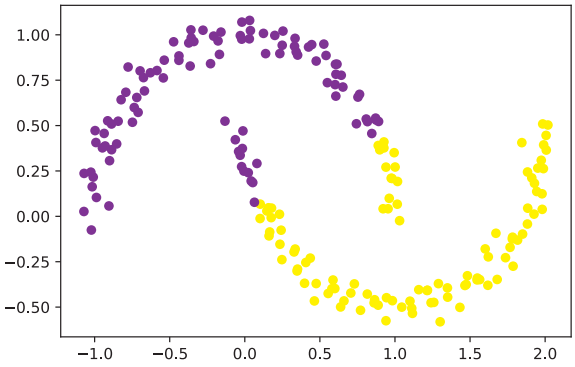


図 8 ■ 混合ガウスモデルによるクラスタリング

次に，一致しなかった理由について考える．図 8 では，点が三日月型に密集している．混合ガウスモデルを用いたクラスタリングでは，まず確率モデルを作成する．そして，確率モデルとして混合ガウスモデルを選択する．図 8 に当てはまるような確率モデルとしては，2 つの三日月の周辺では値（密度）が大きくなり，それ以外の場所では値（密度）が小さくなるような関数が理想的である．$K=2$ の場合の混合ガウスモデルは，ピークが 2 つある（p.90 の図 5.10(a) 参照）．基本的には，そのピーク周辺ではおおよそガウス分布の形をしていると考えられる．ガウスモデルはピーク周辺に密度が大きい場所が円形または楕円形に広がる．つまり，$K=2$ の混合ガウスモデルの場合，密度の大きい場所は 2 か所あり，それぞれ密度が大きい場所が円形または楕円形に広がると考えられる（p.90 の図 5.10(b) 参照）．実際，図 8 に混合ガウスモデルを当てはめた結果は，図 9 のようになる．このモデルでの密度が最も大きい場所は，2 つの濃色の楕円で表される．この図から，実際のデータの密度が大きい場所（三日月）とモデルの密度が大きい場所（楕円）が一致していないことが読み取れる．このことから，混合ガウス分布を用いたクラスタリングは，視覚的に読み取ったクラスタリングと一致しなくなる．

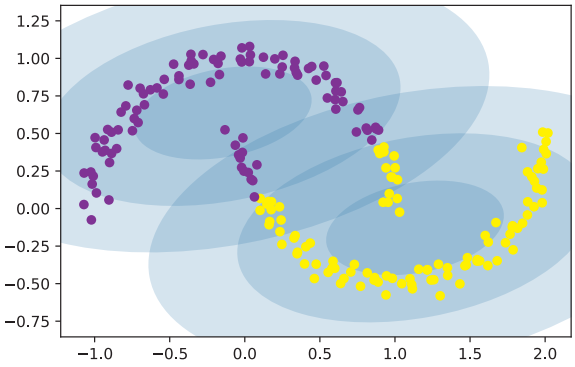


図 9 ■ 混合ガウスモデルの当てはめ

問 5.4 自己組織化マップにおける参照ベクトルの学習アルゴリズムの疑似コードとして，例えば次のアルゴリズム 2 が考えられる．

アルゴリズム 2 ■ 自己組織化マップ

Input: データ点 $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
Output: 参照ベクトル $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$

1: **for** $m = 1$ **to** M **do**
2: $\quad \boldsymbol{w}_m$ にランダムな数値を代入
3: **end for**
4: **while** D があらかじめ定められた基準を上回る **do**
5: $\quad$ **for** $m = 1$ **to** M **do**
6: $\quad\quad \boldsymbol{w}'_m = \boldsymbol{w}_m$
7: $\quad$ **end for**
8: $\quad$ **for** $i = 1$ **to** N **do**
9: $\quad\quad m^* = \mathrm{argmin}_{m \in \{1,2,\ldots,K\}} \|\boldsymbol{x}_i - \boldsymbol{w}_m\|^2$
10: $\quad\quad C$ に z_{m^*} の近傍にあるノードのノード番号を格納
11: $\quad\quad$ **for** $j = 0$ **to** $|C| - 1$ **do**
12: $\quad\quad\quad \boldsymbol{w}_{C[j]} = \boldsymbol{w}_{C[j]} + \alpha(\boldsymbol{x}_i - \boldsymbol{w}_{C[j]})$
13: $\quad\quad$ **end for**
14: $\quad\quad$ 配列 C を空にする
15: $\quad$ **end for**
16: $\quad \alpha$ を 0 に近づける
17: $\quad \{\boldsymbol{w}'_1, \ldots, \boldsymbol{w}'_M\}$ と $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$ を比較，その差を数値化し，その値を D に代入
18: **end while**
19: **return** $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$

以下に，この疑似コードを解説する．

[Step 1] 参照ベクトルの学習アルゴリズムは，各ノードの参照ベクトルをランダムに設定することから始まる．これに対応するのが，1 行目と 2 行目である．

[Step 2] 次に，$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ の中から 1 つベクトルを取り出し，これを $\boldsymbol{x}_i$ として $m^* = \mathrm{argmin}_{m \in \{1,2,\ldots,M\}} \|\boldsymbol{x}_i - \boldsymbol{w}_m\|^2$ を求めるが，これは 9 行目で行う．

[Step 3] 本文では，ノード z_{m^*} を勝者ノードと述べた．そして，勝者ノードの近傍のノード（p.103 の図 5.19 参照）を取り出し，取り出されたノードがもつ参照ベクトルを，$\boldsymbol{x}_i$ に近づくように修正する．これについてはまず，10 行目で勝者ノードの近傍のノード番号を配列 C に格納している．

11 行目から 13 行目では，取り出されたノードの参照ベクトルを $\boldsymbol{x}_i$ に近づくように修正する．ここで，α は $0 < \alpha < 1$ を満たす数値としてあらかじめ定めておく．

12 行目の $\boldsymbol{w}_{C[j]} + \alpha(\boldsymbol{x}_i - \boldsymbol{w}_{C[j]})$ の意味については，図 10 を参照すること．$\boldsymbol{w}_{C[j]}$ は取り出されたノードの参照ベクトルを意味する．この図から，$\boldsymbol{w}_{C[j]} + \alpha(\boldsymbol{x}_i - \boldsymbol{w}_{C[j]})$ は $\boldsymbol{w}_{C[j]}$ を $\boldsymbol{x}_i$ に近づけたものであることが読み取れる．11 行目から 13 行目は，j に関する for 文であるから，取り出された各ノードの参照ベクトルを全て修正することになる．

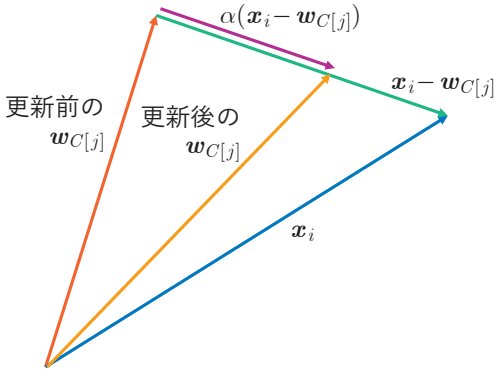


図 10 ■ $\boldsymbol{w}_{C[j]} + \alpha(\boldsymbol{x}_i - \boldsymbol{w}_{C[j]})$ の意味

[Step 4] 以上により取り出された $\boldsymbol{x}_i$ について，参照ベクトルの修正を行う．次に，$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ の内の他のベクトルに対して同じような参照ベクトルの修正を行う．これは，8 行目から 15 行目までの i についての for 文で，$\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ の各ベクトルに対して参照ベクトルの修正を自動で行う．

[Step 5] ひととおり参照ベクトルの修正が終わると，まず 16 行目で α を 0 に近づける．これは学習を安定させるための工夫である．

とりあえず α は 12 行目で使われ，どのくらい $\boldsymbol{x}_i$ に近づけるのかを意味する数値であるが，この数値を学習が進むにつれて変化させる，ということを知っていればよい．

最後に，更新前の参照ベクトル $\{\boldsymbol{w}'_1, \ldots, \boldsymbol{w}'_M\}$ と更新後の参照ベクトル $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_M\}$ を比較し，この前後の差がほとんどなくなるまで学習を繰り返すことになる．

第 6 章　教師あり学習

問 6.1

$$\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}_a \\ \boldsymbol{y}_b \\ \boldsymbol{y}_c \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, \qquad \boldsymbol{X} = [\boldsymbol{x}_a,\ \boldsymbol{x}_b,\ \boldsymbol{x}_c]^T = \begin{bmatrix} 1 & 2 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}$$

となるので,

$$\begin{aligned} \boldsymbol{W} &= \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{Y} \\ &= \begin{bmatrix} 14 & 7 \\ 7 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 2 & 3 \\ 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \\ &= \frac{1}{35}\begin{bmatrix} 16 & -8 \\ -7 & 21 \end{bmatrix} = \begin{bmatrix} 0.457 & -0.229 \\ -0.200 & 0.600 \end{bmatrix} \end{aligned}$$

問 6.2　ヒントより,

$$\begin{aligned} \frac{\partial y_n}{\partial w_d} &= (1-y_n)y_n \cdot \frac{\partial u_n}{\partial w_d} \\ &= (1-y_n)y_n \cdot x_{nd} \quad (d = 0,\ 1,\ 2) \end{aligned}$$

$u_n = w_0x_{n0} + w_1x_{n1} + w_2x_{n2} = \boldsymbol{x}_n^T\boldsymbol{w}$,　$x_{n0} = 1$ であることに注意．上式をヒントの式に代入すると,

$$\frac{\partial E_n(\boldsymbol{w})}{\partial w_d} = (t_n - y_n)x_{nd}$$

(6.28) 式から,

$$\begin{aligned} \frac{\partial E(\boldsymbol{w})}{\partial w_d} &= -\sum_{n=1}^{N}\frac{\partial E_n(\boldsymbol{w})}{\partial w_d} \\ &= -\sum_{n=1}^{N}(t_n - y_n)x_{nd} \end{aligned}$$

また，w_d の更新式は,

$$w_d^{(t+1)} = w_d^{(t)} + \eta\sum_{n=1}^{N}(t_n - y_n)x_{nd}$$

となる．

第 7 章　確率モデル・確率推論

問 7.1　多くの例が存在するが，例えば，分類問題としては，迷惑メールのフィルタリングや医療診断など．迷惑メールのフィルタリングでは，迷惑メールに含まれやすい文字列が非迷惑メールにも含まれる場合があるため，迷惑メールか非迷惑メールかという分類と文字列との間に不確実さが存在する．この不確実さを確率モデルとして取り扱うことより，より精緻な推定が可能となる．

問 7.2　まず，例題 7.1 で求めた検査方法 1 での陽性結果を受け取った段階での事後確率を，検査方法 2 を行う前の段階での事前確率とみなして $P(\text{感染}) = 0.67$ とする．次に，例題 7.1 と同様にベイズの定理を用いることで，検査方法 1 での陽性結果に引き続き検査方法 2 での陽性結果を得た段階での事後確率として，$P(\text{感染} \mid \text{陽性}) \approx 0.99$ が得られる．

問 7.3　まず，事前確率がラプラス分布に従う場合を考える．事後確率の負の対数として得られる

$$E_l(w_1, w_2) = \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - w_1\phi_1(x_i) - w_2\phi_2(x_i))^2 + \lambda\left(|w_1| + |w_2|\right)$$

に注目し，この目的関数を最小化する係数 w_1, w_2 を考える．ここで，$E_l(w_1, w_2)$ の第 1 項は基底関数によるデータの再構成誤差を示し，第 2 項は係数の正則化を示す．図 11(a) の破線と太線は，それぞれ，$E_l(w_1, w_2)$ の第 1 項と第 2 項の等高線を示す．第 2 項による正則化の効果が，原点を中心とするひし形になることに注意すると，両者の接点である黒丸が求められる係数となり，一方の係数がゼロ $(w_1 = 0)$ となる結果が得られている．すなわち，ラプラス分布に基づく正則化が図 11(a) の太線に示す「尖った」効果をもつことにより，スパースな解が得られている．

一方，事前確率がガウス分布に従う場合には，

$$E_g(w_1, w_2) = \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - w_1\phi_1(x_i) - w_2\phi_2(x_i))^2 + \mu\left(w_1^2 + w_2^2\right)$$

となる．$E_g(w_1, w_2)$ の第 2 項による正則化の効果が，原点を中心とする円となることで，図 11(b) のように，両方の係数が非ゼロとなり，スパースではない解が得られている．

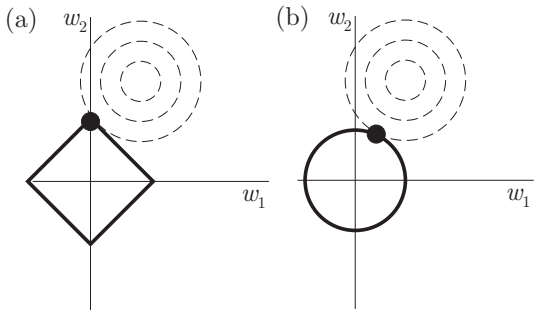


図 11 ■ 事後確率最大化法と事前分布

問 7.4　例えば，医療診断は，損失の非対称性が存在する例として挙げられる．すなわち，病気でないと診断したが実際には病気である場合の損失は，病気であると診断したが実際には病気でない場合の損失と比べて，大きいと考えられる．

第 8 章　強化学習

問 8.1

- 状態：車の速度，傾き，周囲の物体との距離，駐車枠との位置関係
- 行動：アクセル／ブレーキの踏力，ハンドルの角度・角速度
- 報酬：周囲の物体との距離／車の速度変化／駐車枠のとの位置関係に応じた正の報酬（即時報酬），周囲の物体と衝突に対して（大きな）負の報酬

問 8.2

- 小さすぎる場合：現在の Q 関数の推定値を重視して更新が小幅になり，収束に時間がかかる
- 大きすぎる場合：行動によって得られる報酬を重視し，報酬に含まれるノイズに対して敏感になる．パラメータの収束がスムーズでなくなり，時間がかかる場合がある．

問 8.3　分散が小さくなれば，毎回の行動によって得られる報酬が真の行動価値 $Q^*(a)$ に近づく（$Q_t(a)$ の収束が速くなる）ため，最適な ε はより小さくなる．

第 9 章　情報センシング

問 9.1　ランニングフォームを測定し，疲れたときにフォームが悪くなってないか，走行時の左右の足のバランスが悪くないかを教えてくれるようなサービスを考えてみる．ランニングフォームにおいて大事なのは，

(1) ヒザの曲がり具合とタイミング
(2) 着地圧 (足圧) の大きさと左右バランス

といったところだと仮定すると，上記 (1)(2) が取得するべき状況である．それらを認識するには，(1) の場合は例えば膝サポータに曲げセンサを仕込んでおけば取得できそうである．(2) は例えば靴のインソールに圧力センサを置けばよい．

問 9.2　以下，「気をつけなければいけないこと：その理由」のように記す．

- 目的とするべき状況を明確に定義する：明確でなければセンサも決まらないし，求める精度や粒度が決まらない．
- 適切なセンサを選択する：求める状況が認識できるのか，使用に耐え得るのか，電池はもつのか，壊れないか，といった要素で求めるクオリティに合ったセンサを使っていなければ，価値のあるデータが取得できない．
- 計測がうまくできているかわかりやすい状況にしておく：計測しているつもりで実はできていないということがあれば，計測作業が無駄になる．
- 取得データの前処理を行う：データにノイズや外れ値が入っていると適切に学習できなかったり，求めたサービスが実現できなかったりするため．

第 10 章　画像解析・深層学習

問 10.1　アルファベットは A から Z まで 26 文字，大文字と小文字があるので，クラスは最低 52 個必要．

認識に用いる特徴量は，文字認識の場合は，例えば，各クラス（A，B，C などの 文字）の文字を構成する直線の数や角度，穴の数など．

※**補足**　ほかにも，多くの特徴量が考えられ多岐にわたる．文字認識の性能は特徴の抽出と選択に左右される．

問 10.2　1 画素あたりのデータ量は 8 bit であり，8 bit = 1 byte である．したがって，全画素のデータ量は，

$$644 \times 592 \times 1 = 381,248\,\text{byte}$$

第 11 章　時系列データ解析・音声解析

問 11.1

$$\frac{5.0\,\text{秒} \times 16,000\,\text{sample/秒} \times 16\,\text{bit/sample}}{8\,\text{bit/byte}} = 160,000\,\text{byte}$$

問 11.2　音声中のある時刻の発話内容を推定する際，単方向 RNN はその時刻より過去の情報を参照して推論を行うのに対して，双方向 RNN は過去に加えて未来の情報も参照して推論を行う．そのため，一般に単方向 RNN より双方向 RNN の方が音声認識精度が高い．

しかし，単方向 RNN ではユーザが発話途中でもリアルタイムで音声認識（ストリーミング音声認識）が可能なのに対して，未来の情報が必要な双方向 RNN では，ユーザが発話を終了してからでないと音声認識が開始できない．

以上の議論より，それぞれの長所・短所は次のようにまとめられる．

- 単方向 RNN
  - 長所：ストリーミング音声認識が可能．
  - 短所：音声認識精度が双方向 RNN に比べて劣る．
- 双方向 RNN
  - 長所：音声認識精度が単方向 RNN に比べて優れている．
  - 短所：ストリーミング音声認識が困難．

第 12 章　テキスト解析

問 12.1

- CC-100: Monolingual Datasets from Web Crawl Data
  `http://data.statmt.org/cc-100/`
  多言語（116 言語）．論文（CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data，Guillaume Wenzek，et．al．Proceedings of The 12th Language Resources and Evaluation Conference，4003–4012，2020）のために収集された言語別に Web サイトから収集されたデータ．ローマ字表記されたテキストデータや Web サイトの URL やデータ収集日時などの情報が含まれる．
- Wikipedia: データベース（日本語版）
  `https://ja.wikipedia.org/wiki/Wikipedia:データベースダウンロード`
  日本語．日本語版 Wikipedia の記事データベース．全ページの最新のテキストデータが含まれる．
- Brown Corpus
  `http://www.nltk.org/nltk_data/`
  正式名称 Brown University Standard Corpus of Present-Day American English．英語．1967 年にブラウン大で発表された最初の電子的コーパス．15 のジャンルからなる 500 件（1,014,312 語）のテキストが含まれている．単語には品詞タグが付与されている（URL は NLTK（Natural Language Toolkit）用のデータダウンロードサイト）．
- Large Movie Review Dataset
  `http://ai.stanford.edu/~amaas/data/sentiment/`
  英語．スタンフォード大学で研究のために収集された，映画批評サイト IMDb のレビューテキストのデータ．各テキストには肯定的・否定的を示す 1/0 のラベルが付けられている．
- Reuters-21578 Text Categorization Collection
  `http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html`
  英語．1987 年に配信された Reuter のニュース速報のテキスト 21,578 件．カテゴリごとに分類されている．
- Project Gutenberg
  `https://www.gutenberg.org`
  主に英語．60,000 を超える著作権フリーの電子書籍データベース．電子書籍として読みやすい EPUB や HTML でのデータ以外に，テキストデータ（UTF-8）も提供されている．
- 青空文庫
  `https://www.aozora.gr.jp`
  主に日本語．著作権法上の著作権保護期間の切れた著作と，切れていないがインターネット上での利用が許諾されている著作が含まれている．著作権保護の切れた作品についてはファイル形式の変更なども認められており，テキスト解析に利用できる．
- SQuAD (The Stanford Question Answering Dataset)
  `https://rajpurkar.github.io/SQuAD-explorer/`
  英語．100,000 件を超える質問文と回答文の組と 50,000 件を超える回答不可能な質問のセット．

その他，次のサイトで，タスクごとに利用可能なコーパスについて調べることができる．

- List of datasets for machine-learning research
  `https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research`

- Google Dataset Search
  `https://datasetsearch.research.google.com`
- Datasets for Natural Language Processing
  `https://machinelearningmastery.com/datasets-natural-language-processing/`

問 12.2

- MeCab
  `https://taku910.github.io/mecab/`
  日本語．京都大学と NTT コミュニケーション研究所の共同研究プロジェクトによって開発されたオープンソースの形態素解析ツール．スタンドアロンツールであるが，Python や Ruby，Java などの言語で動作するラッパーがあり，プログラム中からも利用可能である．
- GiNZA
  `https://megagonlabs.github.io/ginza/`
  日本語．Python 用の形態素解析ツール．形態素の抽出以外に，固有表現抽出や，依存構造解析の機能を提供する．
- JUMAN++
  `https://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN%2B%2B`
  日本語．表記揺れに対応したスタンドアロンの形態素解析ツール．
- spaCy
  `https://spacy.io`
  多言語．Python 用の形態素解析ツール．形態素の抽出，品詞のタグ付け，原型への変換，固有表現抽出の機能を提供する．
- NLTK (Natural Language Toolkit)
  `https://www.nltk.org`
  多言語．Python 用の自然言語処理ツールキット．その一部として形態素解析，品詞のタグ付け，原型への変換，固有表現抽出などの機能を提供する．

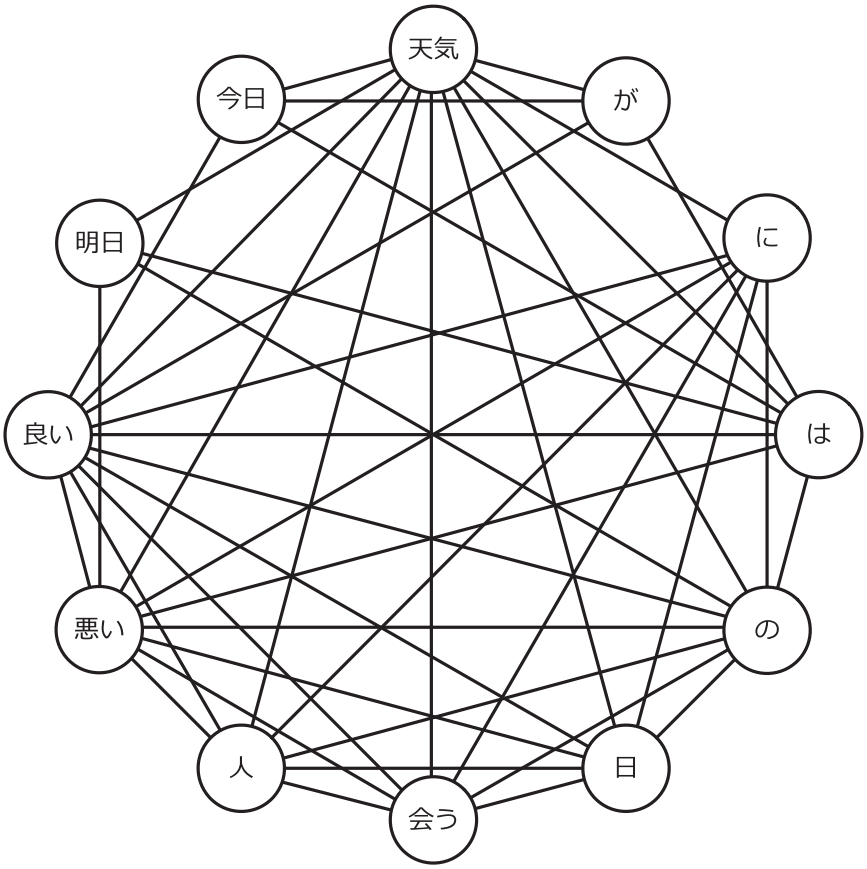
問 12.3 log の底を 10 としたとき，文 2 における TF および IDF は次のように計算される．なお，ここでは $1/3 = 0.33$ としている．

	天気	今日	は	良い	晴れ
TF	0	0.33	0.33	0	0.33
IDF	0.3	0	0	0.3	0.3

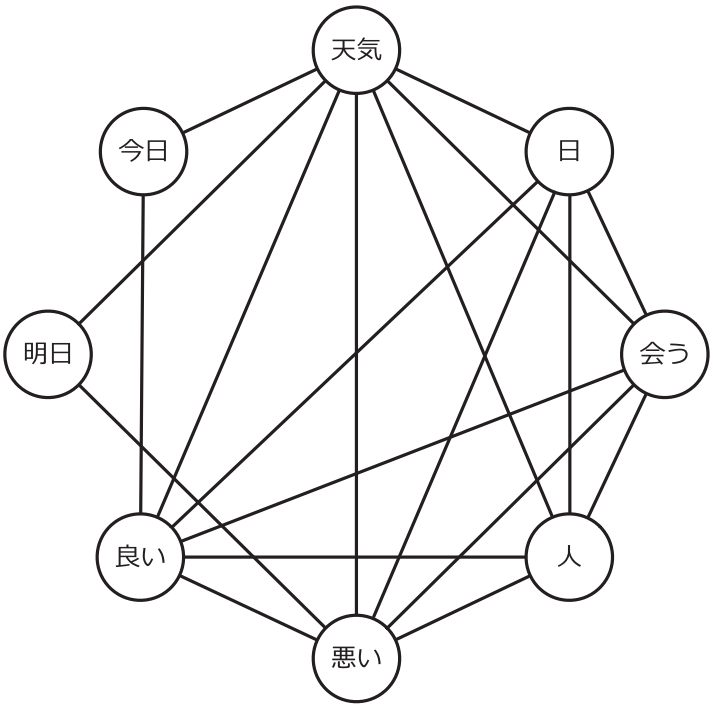
したがって，TF-IDF は次のように求められる．

	天気	今日	は	良い	晴れ
TF-IDF	0	0	0	0	0.099

問 12.4　全ての語（トークン）を含めると次のようになる．

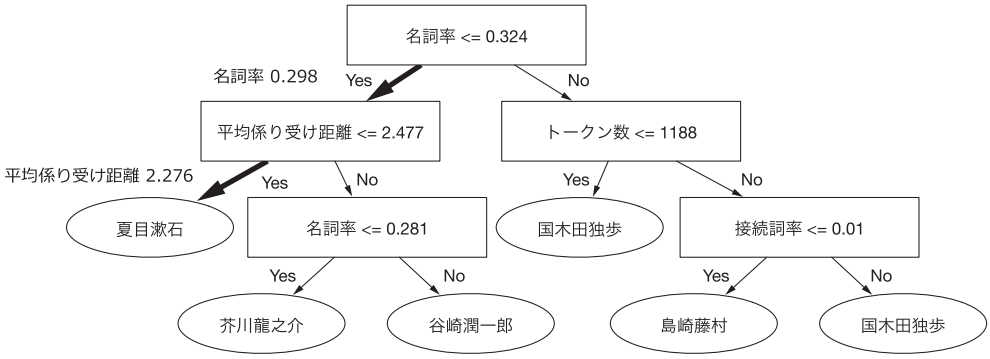


助詞を除いた場合は以下のようになる．



問 12.5

- 坊ちゃん
  名詞率 0.298，平均係り受け距離 2.276 より, 以下のようなノードを経由して,「夏目漱石」となる.



以下同様に

- 細雪 $\longrightarrow$ 谷崎潤一郎
- 酒中日記 $\longrightarrow$ 国木田独歩

となる.

第 13 章　情報セキュリティ

問 13.1　アメリカの Yahoo!社で約 30 億件の個人情報が 2013 年に流出していたことが 2017 年に判明した．また，エクアドルで国民ほぼ全員を含む約 2000 万件の個人情報が流出したと 2019 年にエクアドル政府が発表した．これらは不正アクセスの事案である．

また，アメリカの Dyn 社の DNS サーバが攻撃を受け，同社の顧客企業のサービスにアクセスしにくくなる等の障害が 2016 年 10 月に発生するなどの分散型サービス妨害（Distributed Denial of Service; DDoS）攻撃の事案もある．

2017 年 5 月には，行政，民間企業，医療機関等の多くの組織に世界各国で影響を及ぼしたマルウェア WannaCry の感染被害が発生した．感染したシステムはファイルを暗号化され，その復号鍵と引き換えに金銭を要求する脅迫文が表示されるランサムウェアによる攻撃の事案である．

※**補足**　ここでは海外の事例をいくつか並べたが，日本国内でもサイバー攻撃の被害は発生している．

問 13.2　日本年金機構は 2015 年に年金加入者約 125 万件の情報が流出したと公表した．「『厚生年金基金制度の見直しについて（試案）』に関する意見」という件名の 1 通のメールを職員が開封することから始まり，マルウェアに感染した機構内の複数の PC が個人情報を外部に送信した．

※**補足**　この事案の調査結果報告は，日本年金機構の Web サイトに掲載されている．
`https://www.nenkin.go.jp/oshirase/topics/2015/20150721.files/press0820.pdf`

問 13.3　有料道路の料金所で停止することなく料金支払いを可能にする電子料金収受システム（ETC）は，ETC 車載器と料金所ゲートが無線通信により決済データをやり取りする．このように車両内の端末と道路の設備が通信をすることで，渋滞回避支援や災害時支援のサービスを提供するような交通システムの高度化が進んでいる．道路や料金所などのインフラ側の通信相手が正規のサービス提供者であることが確認できないと，利用者の知らないうちに不正なデータをやり取りして被害に遭うようなことが考えられる．

問 13.4　リアルな世界の印鑑証明書の発行は，まず，登録する印鑑と本人確認書類を市区町村の窓口に持参し，印鑑登録を申請する．本人確認が完了したら印鑑登録証が発行され，それを用いて印鑑証明書を発行してもらうことができる．検証者は提示された印鑑の印影と印鑑証明書に印刷された印影とを見比べることで，その印鑑を利用する人が確かに存在し，その印鑑の持ち主であると市区町村により確認された人であることがわかる．バーチャルな世界の電子証明書はリアルな世界の印鑑証明書に対応し，電子証明書を発行する認証局はリアルな世界の市区町村役場に対応する．

第 14 章　プライバシー保護技術

問 14.1　「＜企業名＞　プライバシーポリシー」でウェブ検索すると，おおかたの企業のページは見つかる．例として，（企業ではないが）首相官邸と内閣府の該当ページを上記の検索語でウェブ検索すると，

【首相官邸】

`https://www.kantei.go.jp/jp/policy/privacy_policy.html`

【内閣府】

`https://www.cao.go.jp/notice/privacy.html`

が見つかる．

問 14.2　ユーザ ID かパスワードが間違いであるとのメッセージを出す．

問 14.3　住所を削除すればよい．

問 14.4　右辺が d_1 となることを示す．

$$\begin{aligned} n\overline{\mathcal{D}} - (n-1)\overline{\mathcal{D}^*} &= n\left(\frac{1}{n}\sum_{i=1}^{n} d_i\right) - (n-1)\left(\frac{1}{n-1}\sum_{i=2}^{n} d_i\right) \\ &= \sum_{i=1}^{n} d_i - \sum_{i=2}^{n} d_i \\ &= d_1 \end{aligned}$$

問 14.5　フィルタ係数の小数点以下 n 桁以降を無視することにする．このとき，フィルタ係数に 10^n を掛けて得られる整数部の回数だけ，入力の暗号文を加算する．これによって，入力とフィルタ係数の乗算結果に 10^n 倍したものが得られるため，これを繰り返すことで畳込み演算の結果が得られる．