

教師あり学習と教師なし学習（高校生・大学初年次向け）

最小二乗法・勾配降下法の次に

Planned and edited by Yoshio Okamoto / Written by Claude Opus 5.5 (Anthropic)

Ver. 0.1 October 4, 2026

※ 以下の内容はAIが生成したものであり、一応編者がチェックしていますが、間違いや思い込みが含まれている場合があることはご注意ください。

© 2026 Yoshio Okamoto. 教育目的での利用は自由です（出典を明記してください）。

はじめに

この資料は、最小二乗法と勾配降下法を学んだ人が次に進むための整理ノートです。機械学習（machine learning）の最も基本的な2つの枠組みである教師あり学習と教師なし学習を、代表的な手法を例にとって説明します。

節	内容	代表的な手法
1	機械学習の分類と全体像	—
2～3	教師あり学習（回帰）	線形回帰（最小二乗法の復習）
4	教師あり学習（分類）	ロジスティック回帰
5	汎化と過学習	多項式回帰，正則化
6～7	教師なし学習（クラスタリング）	k -means 法
8	教師なし学習（次元削減）	主成分分析（PCA）
9	まとめ	—
付録	数式による説明，Python コード	—

読み方 本文は、高校の数学（三角関数，指数・対数，微分，数列の和 $\sum$ ，データの分散と共分散）を使って読めるように書きました。ところどころ高校の範囲を少し越えるところがありますが，その場ではことばで説明を添えています。大学で学ぶ道具（行列，偏微分による導出，固有値など）が必要になる部分は，**脚注**と**巻末の付録A～F**にまとめました。付録は読み飛ばしても本文の流れは追えます。例題の数値はすべて実際に計算した値です（付録Fのプログラム）。

記号の約束

- データの個数を n ，データの番号を添字 i ($i = 1, 2, \dots, n$) で表します。 x_i は i 番目のデータの値です。
- $\sum_{i=1}^n a_i$ は $a_1 + a_2 + \dots + a_n$ のことです（数学Bの数列の和と同じ記号です）。
- $\hat{y}$ （ハット）は**モデルによる予測値**， $\bar{x}$ （バー）は**平均**を表します。
- $\log$ は自然対数（底が $e = 2.718\dots$ の対数）です。

1 機械学習の分類

1.1 「学習」とは何をすることか

最小二乗法は、次のような手順でした。

1. モデルを決める（例：直線 $y = wx + b$ ）
2. モデルの良し悪しを測る関数を決める（例：残差の2乗の和）
3. その関数が最小になるように、パラメータ（ w , b ）を決める

機械学習の多くの手法は、この3段階の形をしています。2の関数を**損失関数**（loss function, または目的関数）と呼び、3の最小化に勾配降下法などを使います。

機械学習 = データから、損失関数を最小にするパラメータを求めること

手法ごとに違うのは、**どんなデータを使い、何を損失とするか**です。この違いによって、機械学習は大きく分類されます。

1.2 データに「正解」があるかどうか

	教師あり学習	教師なし学習
使うデータ	入力と正解の組 (x_i, y_i)	入力だけ x_i
目的	入力から正解を予測する規則を得る	データの中の構造を見つける
損失の基準	予測と正解のずれ	データ自身のまとまり具合・ばらつき
代表的な課題	回帰, 分類	クラスタリング, 次元削減
例	身長から体重を予測する	客を購買傾向でグループに分ける

教師あり学習（supervised learning）の「教師」とは、各データに付いている**正解**（ラベル, label）のことです。先生が問題と答えをセットで与えてくれる状況にたとえられます。

教師なし学習（unsupervised learning）では正解が与えられません。データだけを見て、「似たもの同士のまとまり」や「データが主にどの方向にばらついているか」といった**構造**を見つけ出します。

なお、これらとは別の枠組みとして、行動の結果として得られる「報酬」を手掛かりに学習する**強化学習**（reinforcement learning）がありますが、この資料では扱いません。

2 教師あり学習の枠組み

2.1 問題設定

教師あり学習では、入力と正解の組が n 組与えられます。

$$(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

x_i を**入力**（特徴量, feature）, y_i を**出力**（正解, ラベル）と呼びます。入力は1つの数とは限らず、「身長と年齢」のように複数の数の組（ベクトル）のこともあります。

目標は、パラメータを含むモデル $\hat{y} = f(x)$ を用意し、予測 $\hat{y}_i = f(x_i)$ が正解 y_i に近くなるようにパラメータを決めることです。「近さ」は損失関数

$$L = \sum_{i=1}^n (i \text{ 番目のデータについての, 予測 } \hat{y}_i \text{ と正解 } y_i \text{ のずれ})$$

で測ります。1個のデータについてのずれをどう測るかは、手法によって変わります。

2.2 回帰と分類

教師あり学習は、出力 y の種類によって2つに分かれます。

- ・ **回帰** (regression) : y が連続的な数値 (体重, 気温, 地震の規模など)
- ・ **分類** (classification) : y がいくつかの種類の中からどれか (合格/不合格, 犬/猫/鳥など)

回帰の代表が**線形回帰** (第3節), 分類の代表が**ロジスティック回帰** (第4節) です。

3 回帰：線形回帰（最小二乗法の復習）

3.1 例題：身長から体重を予測する

5人の身長 x (cm) と体重 y (kg) のデータがあります。

人	1	2	3	4	5
身長 x	150	160	165	170	180
体重 y	45	52	58	62	70

モデルを直線 $\hat{y} = wx + b$ とし、損失を残差 (正解と予測の差) の2乗の和

$$L(w, b) = \sum_{i=1}^n (y_i - (wx_i + b))^2$$

とします。これが**最小二乗法** (least squares method) です。

3.2 解

L は w と b の2つの変数の関数です。そこで、 b を定数とみなして w で微分したものと、 w を定数とみなして b で微分したものを、どちらも0とおきます¹。これを解くと (計算は付録 A.1) ,

$$w = \frac{S_{xy}}{S_{xx}}, \quad b = \bar{y} - w\bar{x}$$

が得られます。ここで $\bar{x}$, $\bar{y}$ は平均,

$$S_{xx} = \sum_i (x_i - \bar{x})^2, \quad S_{xy} = \sum_i (x_i - \bar{x})(y_i - \bar{y})$$

です。 S_{xx} を n で割ると x の分散, S_{xy} を n で割ると x と y の共分散なので, w は「共分散 ÷ x の分散」と言い換えることもできます。

¹このように、ほかの変数を定数とみなして1つの変数で微分することを**偏微分** (partial differentiation) といい、 $\partial L / \partial w$ のように書きます。山の斜面にたとえると、 $\partial L / \partial w$ は東西方向の傾き、 $\partial L / \partial b$ は南北方向の傾きです。両方の傾きが0になる地点が谷底 (最小の点) の候補です。

例題では $\bar{x} = 165$, $\bar{y} = 57.4$, $S_{xx} = 500$, $S_{xy} = 425$ なので,

$$w = \frac{425}{500} = 0.85, \quad b = 57.4 - 0.85 \times 165 = -82.85$$

となります. つまり「身長が 1 cm 高いと, 体重はおよそ 0.85 kg 重い」という規則が得られました. 身長 175 cm の人の体重は $0.85 \times 175 - 82.85 = 65.9$ kg と予測されます.

これが教師あり学習の最も単純な例です. 正解 (体重) の付いたデータから, まだ見ていない入力 (175 cm) に対する出力を予測する規則を得たわけです.

3.3 入力が複数の場合

入力が複数あっても (たとえば身長と年齢から体重を予測する), 考え方は同じです. モデルを

$$\hat{y} = b + w_1x_1 + w_2x_2 + \cdots + w_dx_d$$

($x_1, \dots, x_d$ は d 個の入力) とし, 残差の2乗の和が最小になるように $b, w_1, \dots, w_d$ を決めます. これも, 損失をそれぞれのパラメータで偏微分して0とおけば, 連立1次方程式として解けます. 大学では, この連立方程式を行列を使って1つの式 (正規方程式) にまとめて書きます (付録A.2).

線形回帰は, このように解を式で書ける (解析解がある) 特別な例です. 機械学習の多くの手法では解を式で書けないので, 勾配降下法で数値的に求めます. 勾配降下法は, 「各パラメータについて, 損失の傾き (偏微分) と反対の向きに少しずつ動かす」ことを繰り返す方法でした. 次節のロジスティック回帰は, その例です.

4 分類: ロジスティック回帰

4.1 例題: 勉強時間から合否を予測する

8人の勉強時間 x (時間) と試験の合否 y (合格=1, 不合格=0) のデータがあります.

人	1	2	3	4	5	6	7	8
時間 x	1	2	3	4	5	6	7	8
合否 y	0	0	0	1	0	1	1	1

4時間で合格した人と5時間で不合格だった人がいるように, 勉強時間だけで合否がきれいに決まるわけではありません. そこで「合否そのもの」ではなく, **合格する確率**を予測することにします.

4.2 なぜ直線ではだめか

$\hat{y} = wx + b$ をそのまま使うと, x が大きければ $\hat{y} > 1$, 小さければ $\hat{y} < 0$ になり, 確率として解釈できません. そこで, 直線の値 $z = wx + b$ を, 0と1の間に押し込める関数に通します.

4.3 シグモイド関数

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

をシグモイド関数 (sigmoid function, ロジスティック関数) といいます。

z	$-\infty$	-2	0	2	$+\infty$
$\sigma(z)$	0	0.119	0.5	0.881	1

z が大きいと e^{-z} は0に近づくので $\sigma(z)$ は1に、 z が小さいと e^{-z} が非常に大きくなるので $\sigma(z)$ は0に近づきます。値は必ず $0 < \sigma(z) < 1$ で、S字形のグラフになります (図1(a))。これを使って、

$$p = \sigma(wx + b)$$

を「勉強時間 x の人が合格する確率」のモデルとします。これがロジスティック回帰 (logistic regression) です²。

$p = 0.5$ となるのは $w x + b = 0$, つまり $x = -b/w$ のときです。これを決定境界と呼び、 $p \geq 0.5$ なら合格, $p < 0.5$ なら不合格と予測します。

4.4 損失関数：交差エントロピー

分類では、損失に残差の2乗ではなく、次の交差エントロピー (cross entropy) を使います。

$$L(w, b) = - \sum_{i=1}^n \left[y_i \log p_i + (1 - y_i) \log(1 - p_i) \right], \quad p_i = \sigma(wx_i + b)$$

式は複雑に見えますが、中身は単純です。 y_i は0か1なので、

- $y_i = 1$ (合格) の人については、 $[\]$ の中は $\log p_i$ だけが残る
- $y_i = 0$ (不合格) の人については、 $[\]$ の中は $\log(1 - p_i)$ だけが残る

p_i は「合格」に、 $1 - p_i$ は「不合格」に与えた確率です。

つまり各人の損失は、正解に与えた確率の対数にマイナスを付けたものです。正解に確率1を与えれば $-\log 1 = 0$ で損失は0、正解に与えた確率が小さいほど損失は大きくなります ($-\log 0.5 \approx 0.69$, $-\log 0.1 \approx 2.30$)。

交差エントロピーを最小にすることは、「観測された合否の結果が、モデルのもとで最も起こりやすくなるようにパラメータを選ぶ」と同じです。これを最尤法 (さいゆうほう, maximum likelihood method) といいます³。同じように考えると、最小二乗法も「誤差が正規分布に従う」と仮定したときの最尤法になっています。損失関数の選び方は、データについての確率的な仮定と結びついているのです。

²名前に「回帰」とありますが、分類の手法です。なお、 $p = \sigma(z)$ を z について解くと $z = \log \frac{p}{1-p}$ となります。 $\frac{p}{1-p}$ はオッズ (合格する確率と不合格になる確率の比) なので、ロジスティック回帰は「オッズの対数が x の1次式である」というモデルだともいえます (付録B.1)。

³各人の結果が観測どおりになる確率 ($y_i = 1$ なら p_i , $y_i = 0$ なら $1 - p_i$) を全員分掛け合わせたものを尤度 (ゆうど) といいます。積の対数は対数の和なので、尤度の対数にマイナスを付けると交差エントロピーになります。詳しくは付録B.2。

4.5 勾配

勾配降下法を使うには、 L を w , b で偏微分した値（勾配）が必要です。計算すると（導出は付録B.3）,

$$\frac{\partial L}{\partial w} = \sum_{i=1}^n (p_i - y_i) x_i, \quad \frac{\partial L}{\partial b} = \sum_{i=1}^n (p_i - y_i)$$

という簡単な形になります。

$p_i - y_i$ は「予測した確率と正解のずれ」です。勾配は「**ずれ × 入力**」を全員分足し合わせたものになっています。

実は線形回帰の損失の勾配も、「残差 × 入力」を足し合わせた形をしています（付録A.2）⁴。

この勾配を使って、勾配降下法

$$w \leftarrow w - \eta \cdot \frac{1}{n} \frac{\partial L}{\partial w}, \quad b \leftarrow b - \eta \cdot \frac{1}{n} \frac{\partial L}{\partial b}$$

を繰り返します。 η （イータ）は1回に動かす幅を決める**学習率**です。ここでは勾配を n で割って平均にしています。

4.6 例題の結果

$w = b = 0$ から始め、学習率 $\eta = 0.1$ で勾配降下法を繰り返すと（付録F.1のプログラム）、次のように進みます。

反復回数	w	b	損失 L
0	0	0	5.545
10	0.165	-0.124	4.896
100	0.368	-1.196	3.826
1000	0.997	-4.372	2.576
20000	1.282	-5.770	2.503

初期状態（ $w = b = 0$ ）では $z = 0$ なので全員に確率0.5を与え、損失は $8 \times (-\log 0.5) = 8 \log 2 \approx 5.545$ です。学習が進むと損失は減り、約2.503に収束します。得られたモデルによる合格確率は次のとおりです。

時間 x	1	2	3	4	5	6	7	8
確率 p	0.011	0.039	0.127	0.345	0.655	0.873	0.961	0.989
合否 y	0	0	0	1	0	1	1	1

決定境界は $x = -b/w = 5.770/1.282 \approx 4.5$ 時間です。

⁴ロジスティック回帰の損失 L は、2次関数 $y = x^2$ のように下に凸な形をしているので、勾配降下法で必ず最小値にたどり着けます。一方、シグモイド関数の出力に残差の2乗を使うと、凸にならず勾配も小さくなりやすいため、分類では交差エントロピーを使います（付録B.4）。

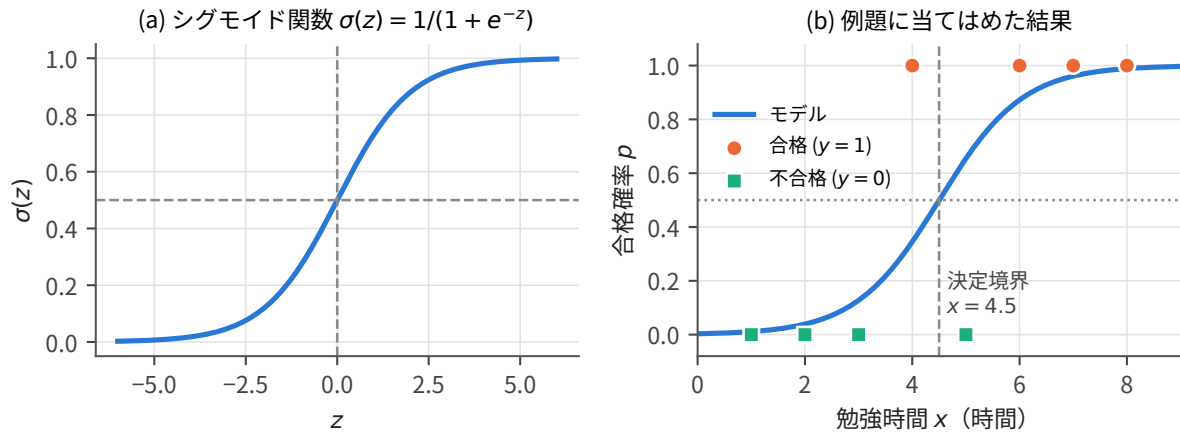


図1 (a) シグモイド関数. $z = 0$ で0.5となり, $z \rightarrow \pm\infty$ で1と0に近づく. (b) 例題のデータ (点) と, 学習で得られたモデル $p = \sigma(1.282x - 5.770)$ (曲線). 破線は決定境界 $x = 4.5$.

4時間の人 (合格) と5時間の人 (不合格) は境界の「反対側」にいますが, モデルはこれを無理に合わせようとはせず, 確率0.345と0.655という「あいまいさ」として表現しています. 損失が0まで下がらないのはこのためです.

4.7 3種類以上への拡張

3種類以上に分類する場合 (例: 犬/猫/鳥) は, 種類ごとに点数 $z_k = w_k x + b_k$ ($k = 1, \dots, K$) を計算し, シグモイド関数の代わりに**ソフトマックス関数** (softmax function)

$$P(\text{種類 } k) = \frac{e^{z_k}}{e^{z_1} + e^{z_2} + \dots + e^{z_K}}$$

で確率に直します. e^z はいつも正なので確率はすべて正になり, 分母がすべての和なので合計はちょうど1になります. 種類が2つ ($K = 2$) のときは, ロジスティック回帰と同じものになります⁵. ChatGPT のような大規模言語モデル (LLM) が「次に来るトークン (単語の断片) の確率」を出すのも, このソフトマックス関数です. トークンの種類の数 K が十数万にもなる, 非常に大きな多クラス分類です⁶.

5 汎化と過学習

5.1 目的は「未知のデータ」をうまく予測すること

教師あり学習で本当に知りたいのは, 学習に使ったデータ (**訓練データ**) での誤差ではなく, **まだ見ていないデータでの誤差**です. 未知のデータに対してもうまく予測できる能力を**汎化能力** (generalization) といいます.

そこで実際には, 手元のデータを**訓練データ**と**テストデータ**に分け, 訓練データだけで学習し, テストデータで性能を評価します.

⁵分子と分母を e^{z_1} で割ると $P(\text{種類}1) = \frac{1}{1 + e^{-(z_1 - z_2)}} = \sigma(z_1 - z_2)$ となり, シグモイド関数に一致します.

⁶たとえば約12.8万種類のトークンを使うモデルがあります. LLM のしくみは, 別の資料 (「microgpt を読む」など) で扱っています.

5.2 例題：多項式の次数を変える

$0 \leq x \leq 1$ で、 $y = \sin 2\pi x$ に小さな誤差（ノイズ）が加わったデータを考えます。訓練データは10点（ $x = 0, \frac{1}{9}, \frac{2}{9}, \dots, 1$ ）、テストデータは訓練データと異なる10点（ $x = 0.05, 0.15, \dots, 0.95$ ）です。

これに M 次の多項式

$$\hat{y} = w_0 + w_1x + w_2x^2 + \dots + w_Mx^M$$

を最小二乗法で当てはめます（図2）。 x の多項式でも、係数 w_j について見れば1次式なので、3.3節の「入力が複数の場合」と同じ方法で解けます（ $x, x^2, \dots, x^M$ を別々の入力とみなせばよい）。

誤差は、残差の2乗の平均の平方根（RMSE, root mean square error）

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

で比べます。標準偏差と同じ形の式で、「予測が平均してどのくらい外れているか」を表します。結果は次のとおりです。

次数 M	訓練データの RMSE	テストデータの RMSE	係数の絶対値の最大
0（定数）	0.711	0.732	0.002
1（直線）	0.572	0.502	1.3
3	0.216	0.168	32.5
9	0.000	0.926	約 6×10^5

- $M = 0, 1$ では、モデルが単純すぎて $\sin$ の形を表せず、訓練・テストともに誤差が大きくなります。これを**未学習**（underfitting）といいます。
- $M = 3$ では、訓練・テストともに誤差が小さくなります。
- $M = 9$ では、10個の係数で10点を通る曲線が作れるので、訓練誤差は0になります。しかし曲線は点と点の間で激しく振動し、テスト誤差は $M = 1$ より悪くなります。これを**過学習**（overfitting）といいます。係数が数十万という極端な値になっているのが過学習の兆候です。

訓練データにぴったり合うことと、よく予測できることは別のことである。

過学習したモデルは、データの背後にある規則（ $\sin 2\pi x$ ）だけでなく、たまたま入ったノイズまで「覚えて」しまっています。

5.3 正則化

過学習を抑える代表的な方法が**正則化**（regularization）です。損失関数に、係数が大きくなることへの罰則を加えます。

$$L = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_j w_j^2$$

第2項は係数の2乗の和に λ （ラムダ）を掛けたもので、係数が大きくなるほど損失が増えます。 $\lambda > 0$ は罰則の強さを決める定数です。これを**リッジ回帰**（ridge regression）と呼びます。リッジ回帰も、最小二乗法とほとんど同じ計算で解を式で求められます（付録A.3）。

$M = 9$ の多項式に正則化を加えた結果は次のとおりです。

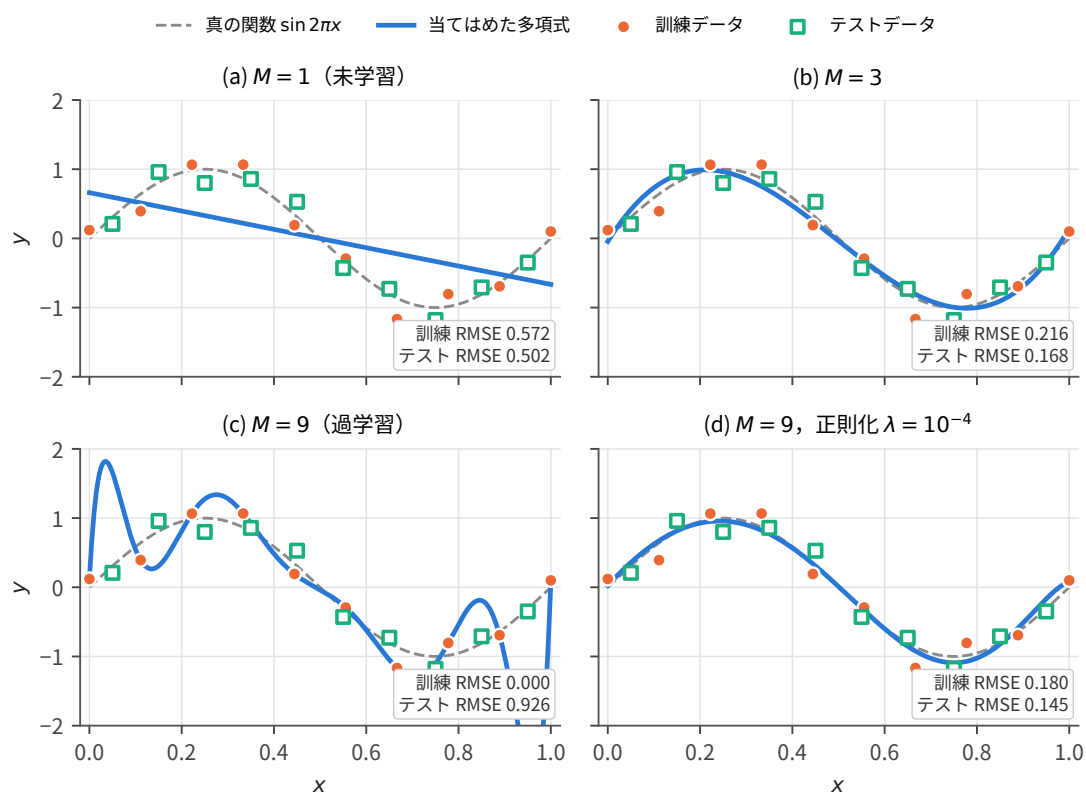


図2 多項式の次数と過学習. 灰色の破線がデータを生成した真の関数, 橙の点が訓練データ, 緑の四角がテストデータ, 青の曲線が訓練データに当てはめた多項式. (c) では曲線が訓練データをすべて通るが, 点の間に大きく振動してテストデータから外れる. (d) は (c) と同じ9次式に正則化を加えたもの (5.3節) .

λ	訓練データの RMSE	テストデータの RMSE	係数の絶対値の最大
0 (正則化なし)	0.000	0.926	約 6×10^5
10^{-6}	0.125	0.153	136
10^{-4}	0.180	0.145	12.2
10^{-3}	0.213	0.171	10.8
10^{-2}	0.331	0.288	3.9

ごく小さな λ を加えるだけで係数の暴走が止まり (図2(d))，テスト誤差が大きく改善します。ただし λ を大きくしすぎると，今度はモデルが単純になりすぎて未学習の側に寄っていきます。

モデルを複雑にする (次数 M を上げる， λ を下げる) と，訓練誤差は減り続けます。一方，テスト誤差は，はじめは減りますが，ある程度以上複雑にすると増え始めます。これは，予測の誤差に「モデルが単純すぎることによる誤差」と「訓練データの偶然のばらつきに振り回されることによる誤差」の2種類があり，モデルを複雑にすると前者は減るが後者は増えるためです⁷。

λ や M のように，学習によってではなく人が前もって決める量をハイパーパラメータ (hyperparameter) といいます。ハイパーパラメータは，テストデータとは別に取り分けたデータ (検証データ) で選びます。テストデータで選ぶと，テストデータに対して過学習してしまうからです。

6 教師なし学習の枠組み

6.1 問題設定

教師なし学習では，正解のないデータ $x_1, x_2, \dots, x_n$ だけが与えられます (各 x_i は，複数の数の組であることが普通です)。予測すべき y がないので，「予測と正解のずれ」という損失は作れません。代わりに，**データ自身の性質に基づく基準**を損失 (目的関数) にします。

代表的な課題は次の2つです。

- ・ **クラスタリング (clustering)**：似たデータ同士をまとめてグループ (クラスタ) に分ける。基準は「各グループがどれだけまとまっているか」。
- ・ **次元削減 (dimensionality reduction)**：多数の変数を，情報をなるべく失わずに少数の変数にまとめる。基準は「まとめた変数がデータのばらつきをどれだけ保っているか」。

6.2 教師なし学習の難しさ

教師なし学習には正解がないので，結果が「正しい」かどうかを直接確かめる方法がありません。たとえばクラスタリングで，データを2つに分けるのと3つに分けるのとどちらが正しいかは，データだけでは決まらないことが多いのです。**結果の解釈には，その分野の知識による判断が欠かせません。**

⁷前者をバイアス，後者をバリエーションと呼び，この関係をバイアス・バリエーションのトレードオフといいます (付録C)。

7 クラスタリング：k-means 法

7.1 考え方

データを K 個のグループに分けます。各グループに**代表点**（中心） $\mu_1, \dots, \mu_K$ （ミュー）を1つずつ置き、各データを最も近い代表点のグループに所属させます。良い分け方とは、各データと、その所属グループの代表点との距離が全体として小さい分け方です。

7.2 目的関数

平面上の点 (x, y) と (a, b) の距離の2乗は $(x - a)^2 + (y - b)^2$ です。これを使って、

$$J = \sum_{i=1}^n (\text{点 } i \text{ から、点 } i \text{ が所属するグループの代表点までの距離})^2$$

を目的関数とし、 J が最小になるような「グループの分け方」と「代表点の位置」を求めます⁸。

7.3 アルゴリズム

分け方と代表点の位置を同時に最適化するのは難しいので、片方を固定してもう片方を最適化する、という操作を交互に繰り返します。

1. **初期化**：代表点 $\mu_1, \dots, \mu_K$ を適当に決める。
2. **割り当て**：各データを、最も近い代表点のグループに入れる。
3. **更新**：各グループの代表点を、そのグループに属するデータの**平均**（重心）に移す。
4. 割り当てが変わらなくなるまで2と3を繰り返す。

なぜうまくいくのか 割り当てのステップでは、代表点を動かさずに、各データを一番近い代表点に付け替えます。どのデータの距離も小さくなるかわからないので、 J は減るかかわらないかです。更新のステップでは、グループを固定したまま代表点を重心に移します。「いくつかの点からの距離の2乗の和」が最も小さくなる位置は、それらの点の平均（重心）なので⁹、このステップでも J は減るかかわらないかです。

J は0以上で、しかも割り当て方は有限通りしかないので、このアルゴリズムは必ず有限回で止まります。ただし、止まった所が J の全体での最小値である保証はありません（**局所的な最小値**に止まることがある）。そのため実際には、初期値を変えて何度か実行し、 J が最も小さい結果を採用します（証明は付録D）。

7.4 例題

次の8個の点を $K = 2$ のグループに分けます。

⁸大学の教科書では、点 i がグループ k に属するとき1、属さないとき0となる変数 r_{ik} を使って、 $J = \sum_{i=1}^n \sum_{k=1}^K r_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$ と書きます。 $\|\cdot\|$ はベクトルの長さです（付録D）。

⁹1次元で確かめてみましょう。 $x_1, \dots, x_m$ からの距離の2乗の和 $f(a) = \sum (x_i - a)^2$ は a の2次関数で、 $f'(a) = -2 \sum (x_i - a) = 0$ より $a = \frac{1}{m} \sum x_i$ （平均）で最小になります。平面では x 座標と y 座標を別々に考えれば同じです。

点	1	2	3	4	5	6	7	8
座標	(1, 1)	(1.5, 2)	(2, 1.5)	(1, 2.5)	(5, 5)	(6, 5.5)	(5.5, 6.5)	(6.5, 6)

人間が見れば、左下の4点と右上の4点に分かれることはすぐ分かります。 k -means 法がこれを見つけられるか確かめるために、初期の代表点をわざと悪く、左下の点1と点2に置きます（付録F.2のプログラム、図3）。

反復	代表点 μ_1	代表点 μ_2	グループ1に入った点	J
0	(1, 1)	(1.5, 2)	点1のみ	132.0
1	(1, 1)	(3.929, 4.143)	点1～4	30.85
2	(1.375, 1.75)	(5.75, 5.75)	点1～4	4.44

- ・反復0：点2～8はすべて代表点 (1.5, 2) の方が近いので、グループ2は7点になります。
- ・反復1：グループ2の重心 (3.929, 4.143) は左下と右上の中間に来ます。これで点2～4は代表点 (1, 1) の方が近くなり、グループ1に移ります。
- ・反復2：それぞれの重心が左下と右上の中心に移動し、割り当てはもう変わりません（収束）。

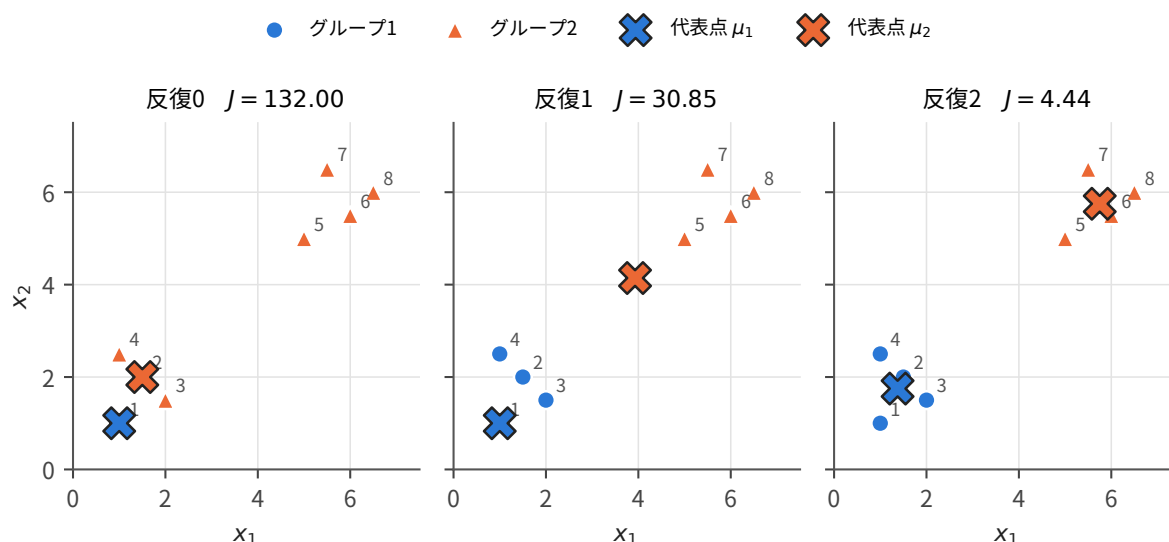


図3 k -means 法の反復の様子。丸（青）がグループ1，三角（橙）がグループ2に割り当てられた点，×印がそれぞれの代表点。数字は点の番号。

誰も「左下と右上に分けよ」とは教えていないのに、データの配置だけから2つのまとまりが見つかりました。これが教師なし学習です。

7.5 注意点

- ・グループの数 K は人が決めます。 K を増やせば J は必ず減る（ $K = n$ なら1点ずつのグループになり $J = 0$ ）ので、 J の大小だけで K を決めることはできません。
- ・「距離」を使うので、変数の単位に結果が左右されます。たとえば身長を cm で測るか m で測るかで結果が変わるため、通常は各変数を標準化（平均0，標準偏差1にそろえる）してから使います。

8 次元削減：主成分分析（PCA）

8.1 考え方

変数が多いデータは、全体像をつかみにくいものです。主成分分析（Principal Component Analysis, PCA）は、データが最も大きくばらついている方向を見つけ、その方向の座標で元のデータを要約する方法です。

8.2 例題：数学と理科の得点

5人の数学と理科の得点です。

人	A	B	C	D	E
数学 x_1	80	60	90	50	70
理科 x_2	75	65	85	60	65

数学が高い人は理科も高い傾向があり、点は右上がりの細長い範囲に分布しています。この「右上がりの方向」を1本の軸として取り出せれば、2つの得点を1つの数（いわば「理数系の総合力」）で要約できそうです。

8.3 平均を原点に移す

まず各変数から平均を引いておきます（**中心化**）。例題では平均はどちらも70点です。平均を引いた値（偏差）と、その分散・共分散は次のとおりです。

人	A	B	C	D	E
数学の偏差 u	10	-10	20	-20	0
理科の偏差 v	5	-5	15	-10	-5

$$\text{数学の分散 } s_u^2 = 200, \quad \text{理科の分散 } s_v^2 = 80, \quad \text{共分散 } s_{uv} = 120$$

ここで分散と共分散は、偏差の2乗の平均、偏差の積の平均です（ $n = 5$ で割っています）。たとえば共分散は $(10 \cdot 5 + (-10)(-5) + 20 \cdot 15 + (-20)(-10) + 0 \cdot (-5))/5 = 600/5 = 120$ です。

8.4 いろいろな方向に射影してみる

原点（平均の点）を通り、 u 軸から角度 θ だけ傾いた直線を考えます。この直線の向きの長さ1のベクトルは $(\cos \theta, \sin \theta)$ です。各人の点 (u_i, v_i) からこの直線に垂線を下ろした足の座標（射影した座標）は、内積

$$z_i = u_i \cos \theta + v_i \sin \theta$$

で表されます。 $\theta = 0^\circ$ なら $z_i = u_i$ （数学の偏差そのもの）、 $\theta = 90^\circ$ なら $z_i = v_i$ （理科の偏差そのもの）です。

直線上に並んだ $z_1, \dots, z_5$ の分散 $V(\theta)$ は、分散と共分散を使って

$$V(\theta) = s_u^2 \cos^2 \theta + 2s_{uv} \sin \theta \cos \theta + s_v^2 \sin^2 \theta = 200 \cos^2 \theta + 240 \sin \theta \cos \theta + 80 \sin^2 \theta$$

と表せます¹⁰。いくつかの角度で計算してみると、次のようになります。

角度 θ	0°	15°	30°	45°	60°	75°	90°	135°
分散 $V(\theta)$	200.0	252.0	273.9	260.0	213.9	148.0	80.0	20.0

$\theta = 30^\circ$ 付近でばらつきが最も大きく、それと直角な 135° 付近（右下がりの方向）ではばらつきがほとんどありません。データが細長く右上がりに分布していることが、数値で表れています。

8.5 分散が最大になる方向

最大になる角度は、三角関数の合成で正確に求められます。2倍角の公式から得られる

$$\cos^2 \theta = \frac{1 + \cos 2\theta}{2}, \quad \sin^2 \theta = \frac{1 - \cos 2\theta}{2}, \quad \sin \theta \cos \theta = \frac{\sin 2\theta}{2}$$

を使うと、

$$V(\theta) = 140 + 60 \cos 2\theta + 120 \sin 2\theta = 140 + \sqrt{60^2 + 120^2} \sin(2\theta + \alpha)$$

(α は $\tan \alpha = \frac{60}{120}$ となる角) と書けます。 $\sqrt{60^2 + 120^2} = \sqrt{18000} \approx 134.16$ なので、

- V の最大値は $140 + 134.16 = 274.16$ 。このとき $\tan 2\theta = \frac{120}{60} = 2$ より $\theta \approx 31.7^\circ$ 。
- V の最小値は $140 - 134.16 = 5.84$ 。そのときの向きは、最大の向きと直角 ($\theta \approx 121.7^\circ$)。

となります。

分散が最大になる向き

$$(\cos 31.7^\circ, \sin 31.7^\circ) \approx (0.851, 0.526)$$

を**第1主成分**の方向といい、それと直角な向きを**第2主成分**の方向といいます。

第1主成分は、数学と理科の両方に正の重みがかかった「右上がり」の方向です（数学の重みが大いのは、数学の得点の方がばらつきが大きいからです）。

8.6 寄与率と主成分得点

上の式から、直角な2つの向き θ と $\theta + 90^\circ$ の分散を足すと、 $\sin$ の項が打ち消し合っているため $140 \times 2 = 280$ になります。これは数学と理科の分散の和 $200 + 80 = 280$ に等しく、データ全体のばらつきの大きさを表します。そのうち第1主成分が占める割合（**寄与率**）は

$$\frac{274.16}{280} \approx 0.979$$

です。

つまり、2つの得点のばらつきの約98%を、第1主成分という1つの数で表せることになります。

各人の第1主成分の座標（**主成分得点**） $z_i = 0.851 u_i + 0.526 v_i$ は次のとおりです（付録F.3のプログラム）。

¹⁰ z_i の平均は0なので、分散は $\frac{1}{n} \sum z_i^2$ です。 $z_i^2 = u_i^2 \cos^2 \theta + 2u_i v_i \sin \theta \cos \theta + v_i^2 \sin^2 \theta$ を全員分足して n で割ると、 $\frac{1}{n} \sum u_i^2 = s_u^2$, $\frac{1}{n} \sum u_i v_i = s_{uv}$, $\frac{1}{n} \sum v_i^2 = s_v^2$ より、この式になります。

人	A	B	C	D	E
数学, 理科	80, 75	60, 65	90, 85	50, 60	70, 65
第1主成分得点 z	11.14	-11.14	24.90	-22.27	-2.63

2次元のデータが1次元の数直線上に並べられ、 $C > A > E > B > D$ という順序が得られました。

変数が3つ以上になると、角度を1つ回すだけでは方向を表せません。大学では、分散と共分散を並べた分散共分散行列の固有値と固有ベクトルを求めることで、一般の場合の主成分を計算します。上で求めた最大値274.16と最小値5.84は、この行列の固有値そのものです（付録E）。

8.7 注意点

- 主成分の「意味」（ここでは「理数系の総合力」）は計算からは出てきません。方向の成分（重み）を見て、人が解釈します。
- k -means 法と同じく、変数の単位の影響を受けます。単位の異なる変数（身長 cm と体重 kg など）を扱うときは、標準化してから行うのが普通です。
- 変数が d 個あれば主成分も d 個得られます。寄与率の大きい順に上位いくつかを使うことで、多数の変数を少数の変数に要約できます。

9 まとめ

9.1 4つの手法の比較

	線形回帰	ロジスティック回帰	k -means 法	主成分分析
分類	教師あり（回帰）	教師あり（分類）	教師なし（クラスタリング）	教師なし（次元削減）
データ	(x_i, y_i) , y は数値	(x_i, y_i) , y は0か1	x_i のみ	x_i のみ
モデル	$\hat{y} = wx + b$	$p = \sigma(wx + b)$	代表点 μ_k	方向 $(\cos \theta, \sin \theta)$
最適化する量	残差の2乗の和（最小）	交差エントロピー（最小）	代表点までの距離の2乗の和（最小）	射影した分散（最大）
解き方	式で解ける	勾配降下法	割り当てと更新の交互反復	分散が最大の向きを求める（一般には固有値問題）

どの手法も、「何かを最小（または最大）にする」という最適化問題として表されていることが分かります。違いは、何を最適化するかにあります。

- 教師あり学習では、最適化の基準が外から与えられた正解との比較で決まります。
- 教師なし学習では、最適化の基準がデータ自身の性質（まとめり具合、ばらつき）で決まります。

9.2 教師あり学習と教師なし学習のあいだ

LLM の学習は「文章の次のトークンを予測する」という課題です¹¹。予測と正解のずれ（交差エントロピー）を最小にするという点では教師あり学習と同じ形ですが、正解（次のトークン）は人が付けたものではなく、文章そのものから自動的に得られます。このように、正解のないデータから正解を自動的に作り出して教師あり学習の形で学ぶ方法を**自己教師あり学習**（self-supervised learning）と呼びます。人手でラベルを付ける必要がないので、膨大な量の文章を学習に使えるのです。

9.3 要点

1. 機械学習は「損失関数を最小にするパラメータを求める」最適化問題である。
2. 教師あり学習は正解付きのデータから予測の規則を学び、教師なし学習は正解なしのデータから構造を見つける。
3. 回帰では残差の2乗の和、分類では交差エントロピーを損失に使う。どちらも最尤法として理解できる。
4. 訓練データでの誤差が小さいことと、未知のデータをよく予測できることは別である（過学習）。
5. 教師なし学習の結果の解釈には、人の判断が欠かせない。

付録A 線形回帰の数学

この付録では、偏微分と行列（大学初年次の線形代数）を使います。

記号 ベクトルは太字の小文字（ $\mathbf{x}$ ），行列は大文字（ X ）で書きます。ベクトルは特に断らない限り縦ベクトルとします。右肩の T は転置（transpose）を表し、べき乗ではありません。転置とは行列の行と列を入れ替える操作で、行列 A の転置 A^T は (i, j) 成分が A_{ji} の行列です。 $\mathbf{a}^T \mathbf{b} = a_1 b_1 + \dots + a_d b_d$ は内積（横ベクトル $\times$ 縦ベクトル = 1個の数）， $\|\mathbf{a}\| = \sqrt{\mathbf{a}^T \mathbf{a}}$ はベクトルの長さ（ノルム）です。

A.1 w と b の導出

$L(w, b) = \sum_i (y_i - wx_i - b)^2$ を偏微分して0とおきます。

$$\frac{\partial L}{\partial b} = -2 \sum_i (y_i - wx_i - b) = 0, \quad \frac{\partial L}{\partial w} = -2 \sum_i (y_i - wx_i - b) x_i = 0$$

1つ目の式を n で割ると $\bar{y} - w\bar{x} - b = 0$ ，すなわち $b = \bar{y} - w\bar{x}$ です。これを2つ目の式に代入すると $\sum_i ((y_i - \bar{y}) - w(x_i - \bar{x}))x_i = 0$ となり， $\sum_i (x_i - \bar{x}) = 0$ ， $\sum_i (y_i - \bar{y}) = 0$ を使って x_i を $x_i - \bar{x}$ に置き換えると

$$S_{xy} - wS_{xx} = 0 \quad \Rightarrow \quad w = \frac{S_{xy}}{S_{xx}}$$

が得られます。

¹¹LLM のしくみと学習については、別の資料（「microgpt を読む」など）で扱っています。

A.2 行列で書く：正規方程式

入力がある場合、

$$X = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} b \\ w_1 \\ \vdots \\ w_d \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$$

とおきます。 x_{ij} は i 番目のデータの j 番目の入力、 X の1列目の1は切片 b のためです。すると予測は $\hat{\mathbf{y}} = X\mathbf{w}$ 、損失は

$$L(\mathbf{w}) = \|\mathbf{y} - X\mathbf{w}\|^2 = (\mathbf{y} - X\mathbf{w})^\top (\mathbf{y} - X\mathbf{w}) = \mathbf{y}^\top \mathbf{y} - 2\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{w}^\top X^\top X \mathbf{w}$$

と書けます。これを $\mathbf{w}$ の各成分で偏微分して並べたもの（勾配ベクトル）は

$$\nabla_{\mathbf{w}} L = -2X^\top \mathbf{y} + 2X^\top X \mathbf{w} = -2X^\top (\mathbf{y} - X\mathbf{w})$$

です。これを $\mathbf{0}$ とおくと、正規方程式

$$X^\top X \mathbf{w} = X^\top \mathbf{y} \quad \Rightarrow \quad \mathbf{w} = (X^\top X)^{-1} X^\top \mathbf{y}$$

が得られます。 $d=1$ のとき、これは付録A.1の結果と一致します。

勾配 $-2X^\top (\mathbf{y} - X\mathbf{w})$ は、「残差 $\mathbf{y} - X\mathbf{w}$ に入力を掛けて足し合わせたもの」という形をしています。ロジスティック回帰の勾配（4.5節）と同じ「ずれ $\times$ 入力」の形です。

多項式回帰 5.2節の M 次多項式は、 X の i 行目を $(1, x_i, x_i^2, \dots, x_i^M)$ とすれば、同じ正規方程式で解けます。

A.3 リッジ回帰の解

正則化した損失 $L(\mathbf{w}) = \|\mathbf{y} - X\mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2$ の勾配は $-2X^\top (\mathbf{y} - X\mathbf{w}) + 2\lambda \mathbf{w}$ なので、これを $\mathbf{0}$ とおくと

$$(X^\top X + \lambda I) \mathbf{w} = X^\top \mathbf{y} \quad \Rightarrow \quad \mathbf{w} = (X^\top X + \lambda I)^{-1} X^\top \mathbf{y}$$

です（ I は単位行列）。正規方程式に λI が加わるだけです¹²。

付録B ロジスティック回帰の数学

B.1 シグモイド関数の性質

$\sigma(z) = 1/(1 + e^{-z})$ を微分すると

$$\sigma'(z) = \frac{e^{-z}}{(1 + e^{-z})^2} = \sigma(z)(1 - \sigma(z))$$

と、自分自身で表せます。この性質のおかげで、B.3で求める勾配がきれいな形になります。

また $p = \sigma(z)$ を z について解くと $z = \log \frac{p}{1-p}$ です。 $\frac{p}{1-p}$ はオッズ（合格する確率と不合格になる確率の比）で、ロジスティック回帰は「オッズの対数が x の1次式である」というモデルだといえます。

¹² $X^\top X$ の逆行列が存在しない、あるいは逆行列の計算が不安定な場合でも、 $\lambda > 0$ を加えると $X^\top X + \lambda I$ は必ず逆行列を持ちます。これも正則化の効果の1つです。

B.2 交差エントロピーはどこから来るか（最尤法）

モデルが正しいとすると、 n 人の合否がちょうど観測どおりになる確率（尤度, likelihood）は

$$\mathcal{L}(w, b) = \prod_{i=1}^n p_i^{y_i} (1 - p_i)^{1-y_i}$$

です（ $\prod$ は積の記号. $y_i = 1$ なら p_i , $y_i = 0$ なら $1 - p_i$ を掛ける）. 観測されたデータを最もよく説明するパラメータとして、尤度を最大にする w, b を選ぶのが**最尤法**です.

積のままでは扱いにくいので対数を取り、さらに最大化を最小化に直すためにマイナスを付けると、

$$-\log \mathcal{L} = -\sum_i [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

となり、交差エントロピーに一致します.

交差エントロピーの最小化は、最尤法そのものです.

最小二乗法と正規分布 回帰で、正解が「予測 $\hat{y}_i$ + 平均0・標準偏差 s の正規分布に従う誤差」で生じると仮定すると、 y_i が観測される確率密度は $\frac{1}{\sqrt{2\pi}s} \exp(-\frac{(y_i - \hat{y}_i)^2}{2s^2})$ です. これを全員分掛けて対数を取りマイナスを付けると、定数を除いて $\frac{1}{2s^2} \sum_i (y_i - \hat{y}_i)^2$ となります. つまり尤度の最大化は残差の2乗の和の最小化と同じで、**最小二乗法は誤差が正規分布に従うと仮定したときの最尤法**です.

B.3 勾配の導出

1個のデータの損失を $\ell = -[y \log p + (1 - y) \log(1 - p)]$, $p = \sigma(z)$, $z = wx + b$ とします. 合成関数の微分（連鎖律）により

$$\frac{\partial \ell}{\partial w} = \frac{\partial \ell}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial w}$$

です. それぞれ

$$\frac{\partial \ell}{\partial p} = -\frac{y}{p} + \frac{1-y}{1-p} = \frac{p-y}{p(1-p)}, \quad \frac{\partial p}{\partial z} = p(1-p), \quad \frac{\partial z}{\partial w} = x$$

なので、掛け合わせると $p(1-p)$ が約分されて $\frac{\partial \ell}{\partial w} = (p-y)x$ となります. b についても同様に $(p-y)$ です. 全員分を足すと4.5節の式になります.

B.4 凸性

ロジスティック回帰の損失 L は (w, b の関数として) 下に凸な関数なので、勾配降下法で必ず最小値にたどり着けます. 一方、シグモイド関数の出力に残差の2乗 $(y - p)^2$ を使うと、損失は凸にならず、また勾配に $p(1-p)$ が掛かって残るため、 p が0や1に近い（予測が大きく外れて自信満々な）ところで勾配がほとんど0になってしまいます. 分類で交差エントロピーを使うのはこのためです.

付録C バイアス・バリエーションのトレードオフ

5.3節で述べたように、モデルを複雑にする（次数 M を上げる、 λ を下げる）と訓練誤差は単調に減りますが、テスト誤差は、はじめは減り、ある程度以上複雑にすると増え始めます。

これは、予測の誤差（の2乗の期待値）が大まかに次の3つに分けられるためです。

- ・ **バイアス** (bias) : モデルが単純すぎて真の関数を表せないことによる誤差。
- ・ **バリエーション** (variance) : 訓練データの偶然のばらつきにモデルが振り回されることによる誤差。訓練データを取り直すたびに、当てはめた曲線が大きく変わるモデルほど大きい。
- ・ **ノイズ** : データそのものに含まれる誤差で、どんなモデルでも減らせない。

モデルを複雑にするとバイアスは減りますがバリエーションは増える、という関係があり、これをバイアス・バリエーションのトレードオフといいます。図2の (a) はバイアスが大きい例、(c) はバリエーションが大きい例です。

付録D k -means 法の数学

点 i の座標をベクトル $\mathbf{x}_i$ 、グループ k の代表点を $\boldsymbol{\mu}_k$ とし、点 i がグループ k に属すれば1、そうでなければ0となる変数を r_{ik} とすると、目的関数は

$$J = \sum_{i=1}^n \sum_{k=1}^K r_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$$

です。割り当てと更新は次のように書けます。

$$r_{ik} = \begin{cases} 1 & k = \underset{j}{\operatorname{argmin}} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{それ以外} \end{cases}, \quad \boldsymbol{\mu}_k = \frac{\sum_i r_{ik} \mathbf{x}_i}{\sum_i r_{ik}}$$

(argmin_j は「最小にする j 」の意味です。)

J が必ず減ることの証明 割り当てのステップでは、 $\boldsymbol{\mu}_k$ を固定したまま、各データを最も近い代表点に割り当てます。各データの項 $\|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2$ がそれぞれ最小になるので、 J は減るか変わらないかです。

更新のステップでは、 r_{ik} を固定したまま J を $\boldsymbol{\mu}_k$ で偏微分して0とおくと

$$\frac{\partial J}{\partial \boldsymbol{\mu}_k} = -2 \sum_i r_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k) = \mathbf{0} \quad \Rightarrow \quad \boldsymbol{\mu}_k = \frac{\sum_i r_{ik} \mathbf{x}_i}{\sum_i r_{ik}}$$

となり、重心が J を最小にする位置です (J は $\boldsymbol{\mu}_k$ の2次関数で下に凸)。したがってこのステップでも J は減るか変わらないかです。

$J \geq 0$ で、かつ割り当て方は有限通り (K^n 通り以下) しかないので、同じ割り当てに戻らない限りアルゴリズムは必ず有限回で止まります。ただし、止まった所が J の全体での最小値である保証はありません。

付録E 主成分分析と固有値

E.1 分散共分散行列

d 個の変数をもつデータ $\mathbf{x}_i$ の平均を $\bar{\mathbf{x}}$ とします. 長さ1のベクトル $\mathbf{w}$ ($\|\mathbf{w}\| = 1$) の方向にデータを射影した座標は $z_i = \mathbf{w}^\top(\mathbf{x}_i - \bar{\mathbf{x}})$ で, その分散は

$$\frac{1}{n} \sum_{i=1}^n z_i^2 = \mathbf{w}^\top \Sigma \mathbf{w}, \quad \Sigma = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$$

と書けます. Σ は分散共分散行列で, 対角成分が各変数の分散, 非対角成分が共分散です. 例題では

$$\Sigma = \begin{pmatrix} 200 & 120 \\ 120 & 80 \end{pmatrix}$$

です. $\mathbf{w} = (\cos \theta, \sin \theta)^\top$ とおくと $\mathbf{w}^\top \Sigma \mathbf{w} = 200 \cos^2 \theta + 240 \sin \theta \cos \theta + 80 \sin^2 \theta$ となり, 本文の $V(\theta)$ に一致します.

E.2 ラグランジュの未定乗数法による導出

問題は, $\|\mathbf{w}\| = 1$ の条件のもとで $\mathbf{w}^\top \Sigma \mathbf{w}$ を最大にする $\mathbf{w}$ を求めることです. そのために

$$F(\mathbf{w}, \lambda) = \mathbf{w}^\top \Sigma \mathbf{w} - \lambda(\mathbf{w}^\top \mathbf{w} - 1)$$

を $\mathbf{w}$ で偏微分して 0 とおきます. Σ は対称行列なので

$$\nabla_{\mathbf{w}} F = 2\Sigma \mathbf{w} - 2\lambda \mathbf{w} = \mathbf{0} \quad \Rightarrow \quad \Sigma \mathbf{w} = \lambda \mathbf{w}$$

となり, $\mathbf{w}$ は Σ の固有ベクトルでなければなりません. このとき分散は $\mathbf{w}^\top \Sigma \mathbf{w} = \lambda \mathbf{w}^\top \mathbf{w} = \lambda$ なので,

最大の固有値に対する固有ベクトルが第1主成分の方向で, その方向の分散は固有値そのものです.

Σ は対称行列なので, 異なる固有値に対する固有ベクトルは互いに直交します. 本文で第2主成分が第1主成分と直角になったのはこのためです.

E.3 例題の計算

固有方程式 $\det(\Sigma - \lambda I) = 0$ は

$$(200 - \lambda)(80 - \lambda) - 120^2 = \lambda^2 - 280\lambda + 1600 = 0$$

なので,

$$\lambda = 140 \pm \sqrt{18000} \quad \Rightarrow \quad \lambda_1 \approx 274.16, \quad \lambda_2 \approx 5.84$$

です. 本文で三角関数の合成によって求めた最大値・最小値と一致します. λ_1 に対する固有ベクトルは $\mathbf{w}_1 \approx (0.851, 0.526)^\top$ です. 全体の分散は固有値の和, すなわち Σ のトレース (対角成分の和) $200 + 80 = 280$ で, 寄与率は $\lambda_1 / (\lambda_1 + \lambda_2) \approx 0.979$ です. 付録F.3のプログラムでは, 固有ベクトルをべき乗法で求めています. 任意のベクトルに Σ を繰り返し掛けて長さを1にそろえると, 最大固有値の固有ベクトルの方向に近づいていく, という性質を使う方法です.

E.4 応力テンソルとの対応

地球科学や構造力学で学ぶ**応力テンソル**も対称行列です。2次元の場合、応力テンソルは

$$\begin{pmatrix} \sigma_{xx} & \tau_{xy} \\ \tau_{xy} & \sigma_{yy} \end{pmatrix}$$

(σ_{xx} , σ_{yy} は垂直応力, τ_{xy} はせん断応力) で, その固有値が**主応力**, 固有ベクトルが**主応力軸**です。

主成分分析は, 分散共分散行列について同じ「主軸」を求める計算であり, 応力テンソルの主応力・主応力軸を求めることと**数学的にはまったく同じ問題**です。

この対応は, 本文8.4節・8.5節の式で直接見ることができます。法線が x 軸から角度 θ 傾いた面にはたらく垂直応力は

$$\sigma_n(\theta) = \sigma_{xx} \cos^2 \theta + 2\tau_{xy} \sin \theta \cos \theta + \sigma_{yy} \sin^2 \theta = \frac{\sigma_{xx} + \sigma_{yy}}{2} + \frac{\sigma_{xx} - \sigma_{yy}}{2} \cos 2\theta + \tau_{xy} \sin 2\theta$$

で, 本文の $V(\theta)$ と同じ形です。対応は「数学の分散 $\leftrightarrow \sigma_{xx}$, 理科の分散 $\leftrightarrow \sigma_{yy}$, 共分散 $\leftrightarrow \tau_{xy}$ 」です。**モールの応力円**は, この式の中心 $\frac{\sigma_{xx} + \sigma_{yy}}{2}$ と半径 $\sqrt{\left(\frac{\sigma_{xx} - \sigma_{yy}}{2}\right)^2 + \tau_{xy}^2}$ を図に表したもので, 例題に当てはめると中心140, 半径134.16の円になり, 円が横軸と交わる2点 274.16 と 5.84 が「主応力」, すなわち2つの固有値 (第1・第2主成分の分散) にあたります。主軸の向きが $\tan 2\theta = \frac{2\tau_{xy}}{\sigma_{xx} - \sigma_{yy}}$ で決まることも, 本文の $\tan 2\theta = \frac{240}{120} = 2$ と同じです。

付録F Python コード (標準ライブラリのみ)

本文の例題を計算するプログラムです。外部ライブラリ (NumPy など) は使わず, Python 3 の標準機能だけで書いてあります。

F.1 ロジスティック回帰 (第4節)

```
import math

# 勉強時間 (時間) と合否 (1=合格, 0=不合格)
x = [1, 2, 3, 4, 5, 6, 7, 8]
y = [0, 0, 0, 1, 0, 1, 1, 1]
n = len(x)

def sigmoid(z):
    return 1.0 / (1.0 + math.exp(-z))

w, b = 0.0, 0.0    # 初期値
eta = 0.1          # 学習率

for step in range(1, 20001):
    gw = gb = 0.0
    for xi, yi in zip(x, y):
        p = sigmoid(w * xi + b)
        gw += (p - yi) * xi    # ∂L/∂w
        gb += (p - yi)        # ∂L/∂b
    w -= eta * gw / n
```

```

b -= eta * gb / n
if step in (1, 10, 100, 1000, 20000):
    L = -sum(yi * math.log(sigmoid(w*xi+b))
            + (1-yi) * math.log(1-sigmoid(w*xi+b)))
    for xi, yi in zip(x, y):
        print(f"step {step:5d}: w={w:7.4f} b={b:8.4f} Loss={L:.4f}")

print("境界 x =", round(-b / w, 3))
for xi in x:
    print(xi, "時間 → 合格確率", round(sigmoid(w * xi + b), 3))

```

実行結果：

```

step    1: w= 0.0875 b=  0.0000 Loss=5.1253
step   10: w= 0.1651 b= -0.1242 Loss=4.8956
step  100: w= 0.3675 b= -1.1960 Loss=3.8259
step 1000: w= 0.9971 b= -4.3722 Loss=2.5761
step 20000: w= 1.2823 b= -5.7703 Loss=2.5030
境界 x = 4.5
1 時間 → 合格確率 0.011
2 時間 → 合格確率 0.039
3 時間 → 合格確率 0.127
4 時間 → 合格確率 0.345
5 時間 → 合格確率 0.655
6 時間 → 合格確率 0.873
7 時間 → 合格確率 0.961
8 時間 → 合格確率 0.989

```

F.2 k -means 法（第7節）

```

# 2次元の点を K=2 個のグループに分ける
points = [(1, 1), (1.5, 2), (2, 1.5), (1, 2.5),
          (5, 5), (6, 5.5), (5.5, 6.5), (6.5, 6)]
K = 2
centers = [points[0], points[1]] # 初期の代表点（わざと悪い選び方）

def dist2(p, q):
    return (p[0] - q[0])**2 + (p[1] - q[1])**2

for it in range(10):
    # (1) 割り当て：各点を最も近い代表点のグループへ
    labels = [min(range(K), key=lambda k: dist2(p, centers[k]))
              for p in points]
    J = sum(dist2(p, centers[l]) for p, l in zip(points, labels))
    print(f"反復{it}: 代表点={[round(c[0],3), round(c[1],3)] for c in centers}")
    f" 割り当て={labels} J={J:.3f}"

    # (2) 更新：各グループの重心を新しい代表点に
    new = []
    for k in range(K):
        g = [p for p, l in zip(points, labels) if l == k]
        new.append((sum(p[0] for p in g) / len(g),
                     sum(p[1] for p in g) / len(g)))
    if new == centers:
        print("収束")
        break
    centers = new

```

実行結果（割り当ての0がグループ1, 1がグループ2）：

```
反復0: 代表点=[(1, 1), (1.5, 2)] 割り当て=[0, 1, 1, 1, 1, 1, 1, 1] J=132.000
反復1: 代表点=[(1.0, 1.0), (3.929, 4.143)] 割り当て=[0, 0, 0, 0, 1, 1, 1, 1] J=30.852
反復2: 代表点=[(1.375, 1.75), (5.75, 5.75)] 割り当て=[0, 0, 0, 0, 1, 1, 1, 1] J=4.438
収束
```

F.3 主成分分析（第8節）

固有ベクトルはべき乗法（付録E.3）で求めています。

```
import math

# 5人の数学・理科の得点
data = [(80, 75), (60, 65), (90, 85), (50, 60), (70, 65)]
n = len(data)

# (1) 平均を引く（中心化）
m = [sum(d[j] for d in data) / n for j in range(2)]
c = [(d[0] - m[0], d[1] - m[1]) for d in data]

# (2) 分散共分散行列 Σ (n で割る)
S = [[sum(ci[a] * ci[b] for ci in c) / n for b in range(2)]
      for a in range(2)]
print("平均", m)
print("Σ =", S)

# (3) べき乗法で最大固有値と固有ベクトルを求める
v = [1.0, 0.0]
for _ in range(100):
    u = [S[0][0]*v[0] + S[0][1]*v[1],
          S[1][0]*v[0] + S[1][1]*v[1]]      # u = Σv
    norm = math.sqrt(u[0]**2 + u[1]**2)
    v = [u[0] / norm, u[1] / norm]
    lam = (v[0]*(S[0][0]*v[0] + S[0][1]*v[1])
            + v[1]*(S[1][0]*v[0] + S[1][1]*v[1])) # λ = v^T Σ v
    trace = S[0][0] + S[1][1]
    print("第1主成分の向き w1 =", [round(t, 4) for t in v])
    print("固有値 λ1 =", round(lam, 3), " 寄与率 =", round(lam / trace, 4))

# (4) 各人の第1主成分得点 z = w1 • (x - 平均)
for d, ci in zip(data, c):
    print(d, "→ z =", round(v[0]*ci[0] + v[1]*ci[1], 2))
```

実行結果：

```
平均 [70.0, 70.0]
Σ = [[200.0, 120.0], [120.0, 80.0]]
第1主成分の向き w1 = [0.8507, 0.5257]
固有値 λ1 = 274.164 寄与率 = 0.9792
(80, 75) → z = 11.14
(60, 65) → z = -11.14
(90, 85) → z = 24.9
(50, 60) → z = -22.27
(70, 65) → z = -2.63
```