

1-4 データ分析

東京大学 数理・情報教育研究センター
2021年5月6日

概要

- 回帰分析, 時系列分析, クラスター分析などの典型的なデータ分析手法を学び, 予測, 分類, パターン発見などへの応用例を学びます.
- 高次元のデータをより扱いやすいものとする次元削減の手法を学びます. また, さまざまなデータ分析の手法の基礎となっている最適化の基本的な概念や具体例を学びます.

本教材の目次

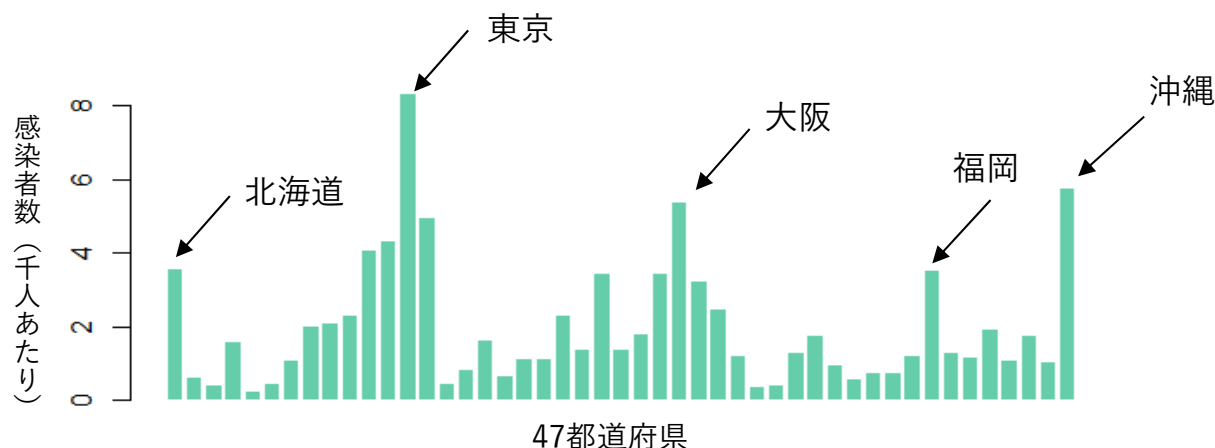
1. 回帰分析	4 – 21
2. ロジスティック分析	22 – 32
3. 時系列分析	33 – 46
4. クラスター分析	47 – 55
5. パターン発見	56 – 64
6. 次元削減	65 – 91
7. 最適化概論	92 – 111

1. 回帰分析

いくつかの量的変数があるときに、特定の目的変数を他の変数によって説明するのが回帰分析で、その結果は予測に用いることもできます。特に、ひとつの説明変数で目的変数を説明する方法は単回帰分析、複数の説明変数で目的変数を説明するのが重回帰分析と呼ばれます。説明変数を入力、目的変数を出力と呼ぶこともあります。回帰分析では、予測誤差とその分散、回帰係数の推定法や、次数や変数の選択が重要です。

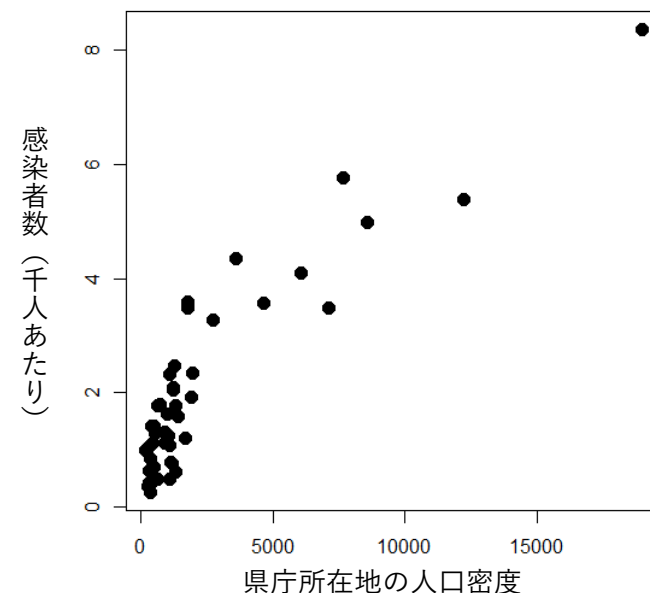
説明変数との関係を見る

下の棒グラフは左から北海道，青森，・・・，沖縄の順に47都道府県の人口千人あたりの新型コロナウイルスの累積感染者数を示しています．都道府県によって感染した人の割合が大きく異なることがわかります．



このような都道府県（以下では単に県と書きます）による感染者の違いを各県の県庁所在地の人口密度と関連づけて説明することを考えてみましょう．

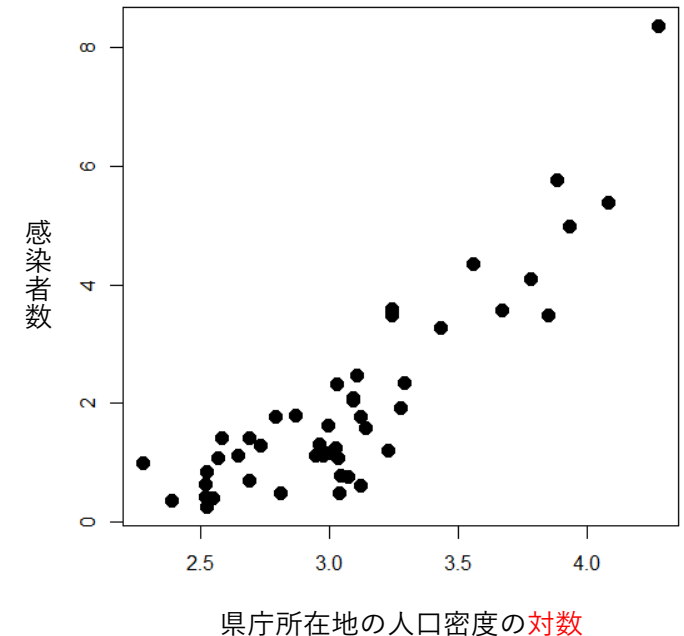
実際に，各県の県庁所在地の人口密度を横軸に，千人当たりの感染者数を縦軸にとって**散布図**を描いてみると，右のように人口密度と感染者数には明確な関係がみえます．



単回帰分析

そこで、人口密度の情報を利用することによって、感染者数の値を説明したり、予測したりすることが考えられます。

右の散布図において、注目する変数を縦軸に表し、**目的変数**あるいは出力と呼びます。一方、横軸には**説明変数**あるいは入力を表します。



【注意】前頁の散布図と違って、右図では人口密度の対数を取って表示しています。このような変換によって二つの変数の関係が直線に近くなり、関係をとらえやすくなることがあることに注意すべきです。

単回帰分析では、目的変数（出力） y の値を

$$y = a + bx$$

のようにひとつの説明変数（入力） x を用いて**回帰直線**で表します。

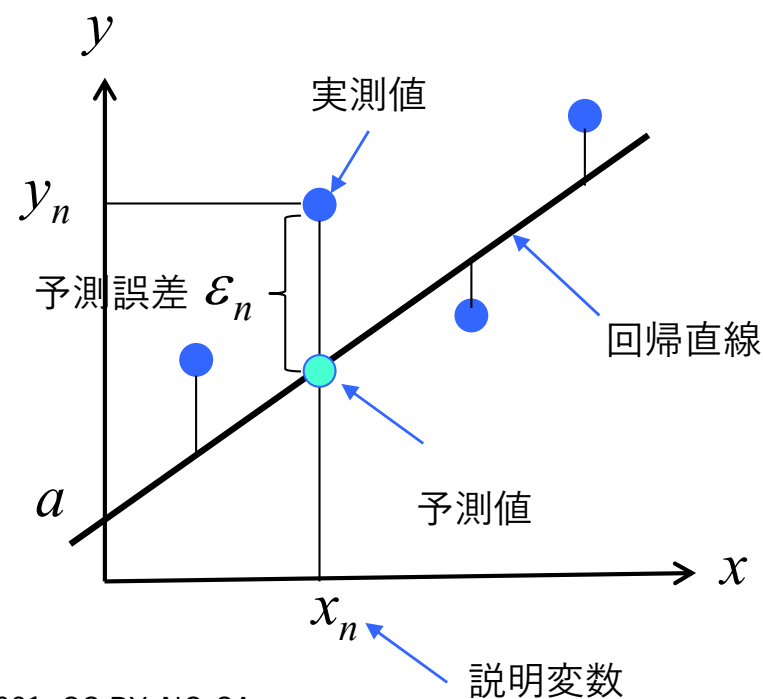
回帰モデル

ただし、実際のデータではすべての点がこの直線上にのるわけではありません。そこで回帰直線と説明変数から定まる予測値と実際の値との差 ε_n を予測誤差とみなし、回帰モデル

$$y_n = a + bx_n + \varepsilon_n$$

を考えることにします。 a と b は回帰係数と呼ばれますが、特に a を切片、 b を直線の傾きと呼ぶこともあります。

以下では回帰係数をデータから推定する方法を考えます。



最小二乗法による回帰直線の推定

回帰モデルの係数 a と b の推定にあたっては、なるべく予測誤差 ε_n を小さくすることを目指します。そのために N 組のデータ $(x_1, y_1), \dots, (x_N, y_N)$ があるとき予測誤差 ε_n の二乗の総和

$$S(a, b) = \sum_{n=1}^N \varepsilon_n^2 = \sum_{n=1}^N (y_n - a - bx_n)^2$$

を最小とするように係数 a と b を求めます。このように誤差の二乗和を最小化することによって未知数を推定する方法を最小二乗法と呼びます。

最小二乗法を適用すると係数の最小二乗推定値が次のように得られます。データから推定された係数などには「 $\hat{}$ 」をつけて表現します。

$$\hat{a} = \bar{y} - b\bar{x}, \quad \hat{b} = \frac{\sum_{n=1}^N (x_n - \bar{x})(y_n - \bar{y})}{\sum_{n=1}^N (x_n - \bar{x})^2}$$

なお、以下のように $S(\hat{a}, \hat{b})$ をデータ数 N で割ると残差分散が得られます。

$$\hat{\sigma}^2 = S(\hat{a}, \hat{b}) / N$$

【参考】 最小二乗法の解を求める

最小二乗法では評価関数 $S(a,b)$ を最小にしますが，そのために $S(a,b)$ を偏微分したものを0とする a と b を求めます．実際に

$$S(a,b) = \sum_{n=1}^N \varepsilon_n^2 = \sum_{n=1}^N (y_n - a - bx_n)^2$$

を a と b について偏微分を計算してみると

$$\frac{\partial S(a,b)}{\partial a} = -2 \sum_{n=1}^N (y_n - a - bx_n) = 0, \quad \frac{\partial S(a,b)}{\partial b} = -2 \sum_{n=1}^N x_n (y_n - a - bx_n) = 0$$

となるので，この連立方程式を解いて，前頁の解が得られます．

回帰係数などの未知数の推定にあたって，残差の二乗和を最小にすることは，残差 ε_n が**正規分布**に従っていると仮定しているものと考えることができます．この想定する分布の形が異なると，推定結果も異なってきます．

例：新型コロナウイルス感染者数データ

右図のCovid-19データの場合に a と b を求めてみると

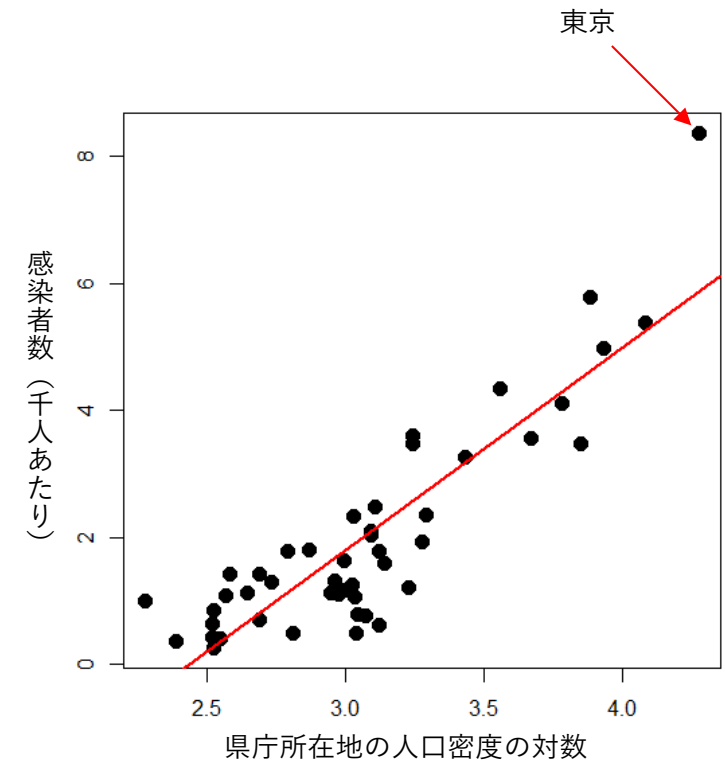
$$a = -7.7792, \quad b = 3.1890$$

となります。したがって、回帰直線は

$$y = -7.7792 + 3.1890x$$

となって、赤線で示した直線が得られます。

最も右上の点(東京)は誤差が大きいです、
それ以外の県についてはよく傾向を表現でき
ているといえます。



多項式回帰モデル

単回帰モデルはデータに直線をあてはめますが，入力と出力の関係が直線では十分に表現できない場合には，**多項式回帰モデル**

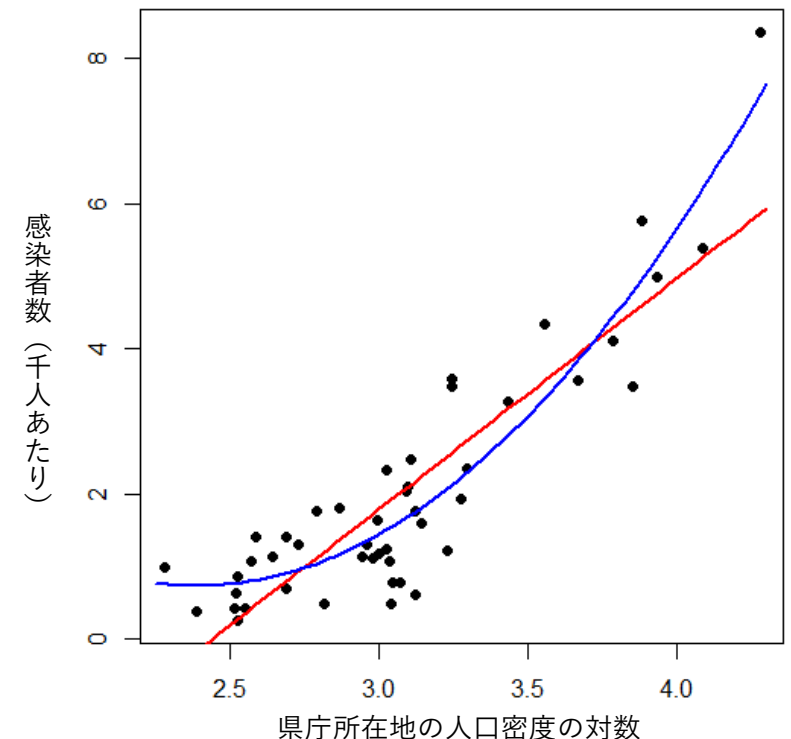
$$y_n = a_0 + a_1 x_n + \cdots + a_m x_n^m + \varepsilon_n$$

を使うとよいことがあります．ここで， m は**多項式の次数**， a_j は回帰係数と呼ばれます． $m=1$ のときは直線， $m=2$ の時には2次式となります．

右図の青色の曲線は $m=2$ として得られた2次の多項式

$$y_n = 11.499 - 9.026x_n + 1.891x_n^2$$

です．赤の直線よりも，データに良く適合しているように見えます．

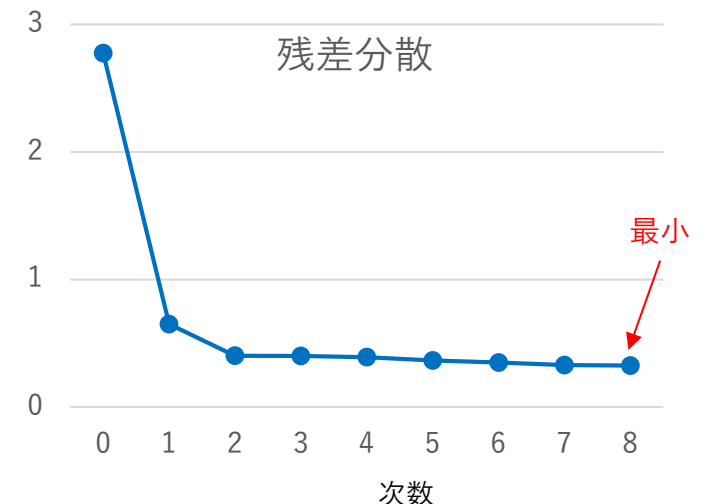
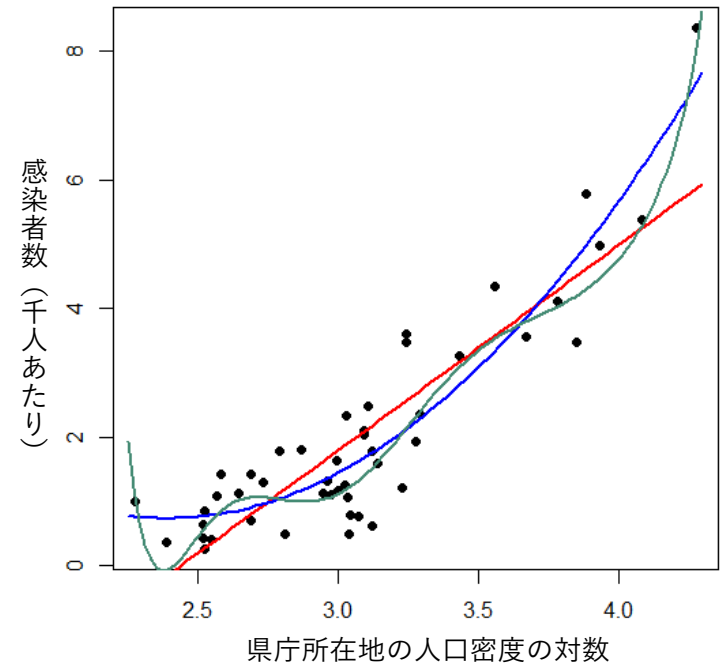


表現力が高いモデルを使う時の注意：

直線を用いた単回帰の代わりに2次式を用いると良い当てはまりが得られたので、もっと高い次数を使ってみたいくなるのは自然です。

実際、**8次の多項式**を使ってみると右上の図の緑色の曲線のように、データの両端ではちょうどデータの上を走っていてデータを忠実に表現しています。

実際、**残差分散**をプロットしてみると、次数が高くなると**残差分散は単調に減少**しており、高次のモデルほどデータへの適合がよいことが分ります。しかし、この8次の回帰曲線は本当に良いモデルでしょうか？

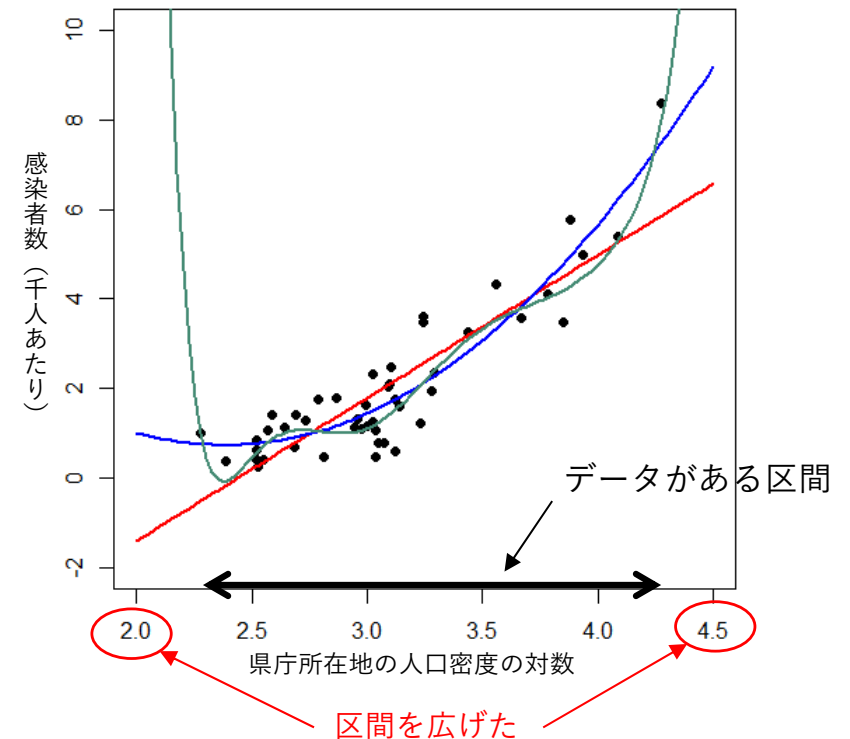


表現力が高いモデル利用時の注意： 過学習

表現力が高すぎるモデルを使うと、データにはよく適合するが観測されたデータ以外の予測、特に**外挿**を行ったときの予測精度が著しく悪くなる恐れがあります。

実際、説明変数の範囲を2.0から4.5に広げて回帰曲線を描いてみると図のようになります。外挿区間を広げても直線や2次式は安定していますが、緑色の8次式は両端で跳ね上がっていて、**外挿能力**が低いことがわかります。

データをよく表現しているが、予測能力が低くなる現象は**過学習**あるいは**過剰適合(オーバーフィッティング)**と呼ばれます。学習に使ったデータだけはよく表現しているが、それ以外のデータにも使える汎用性がないのです。

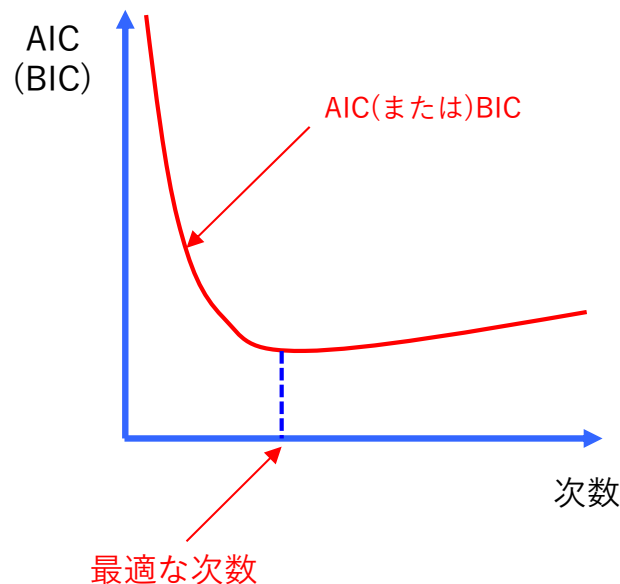


次数選択, モデル選択

多項式の次数は低すぎると現象を十分に表現できませんが、高すぎると過剰適合を起こします。したがって次数を適切に選ぶことが重要で、**次数選択**の問題と呼ばれます。

予測の観点からモデルの良さを評価し、次数を自動的に選択するための評価基準として情報量規準 **AIC**や**BIC**が用いられます。

候補となるすべての次数 m に対して AIC_m (あるいは BIC_m) を計算してその中で最小にする m を探すと、それが**最適な次数**となります。



AICは赤池情報量規準, BICはベイズ型情報量規準と呼ばれます。

情報量規準AIC と BIC

m 次の多項式回帰モデルの場合、 N をデータ数、 σ_m^2 は m 次の多項式回帰モデルの残差分散とすると、AIC と BIC は（共通の定数を見捨てる）次のように定義されます。

$$\text{AIC}_m = N \log 2\pi\sigma_m^2 + 2(m+2)$$

$$\text{BIC}_m = N \log 2\pi\sigma_m^2 + (m+2) \log N$$

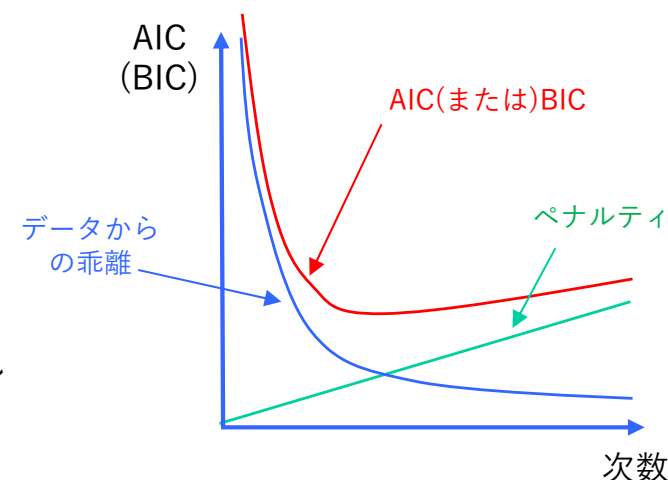
データからの乖離

モデルの複雑さへのペナルティ

ここで、右辺の第1項はデータからの乖離を、第2項はパラメータ数を増やすことによるモデルの複雑さへのペナルティ項と考えることができます。

データからの乖離は次数を増やすと減少する一方、ペナルティ項は増大するので、通常適当な次数で最小となります。

【注意】上記の定義は回帰モデルの場合で、情報量規準はもっと一般のモデルに適用できます。また、AICやBICはモデルの次数選択だけでなく、モデルの選択や変数選択にも利用できます。



次数選択の例

表は新型コロナウイルス感染者数（千人当たり）データに様々な次数の多項式をあてはめた場合の残差分散, AIC, BICを計算した結果です.

次数が増えると, データへの適合度に対応する残差分散は単調に減少しますが, AICやBICは次数2で最小となっており, このデータの場合は **2次曲線が適当**であることがわかります.

13ページの外挿結果を見ると, AIC や BIC が残差分散はある程度大きくても, 低い次数の多項式を選択しているのは妥当な判断であることがわかります.

次数	残差分散	AIC	BIC
0	2.776	185.37	185.22
1	0.650	119.14	120.84
2	0.403	98.62	102.17
3	0.401	100.39	105.79
4	0.391	101.27	108.52
5	0.366	100.08	109.18
6	0.349	99.88	110.83
7	0.329	99.10	111.90
8	0.325	100.49	115.14

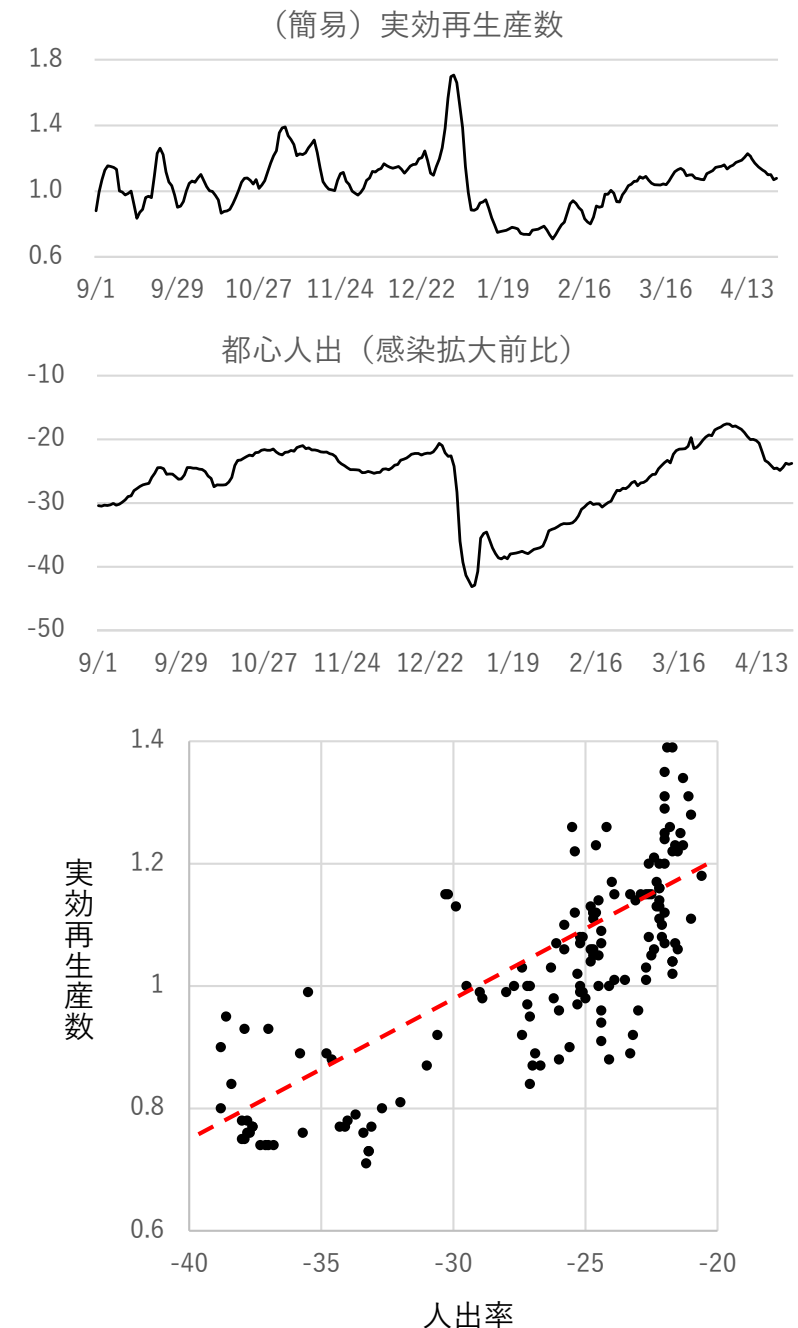
例：実効再生産数について

上の図は東京におけるコロナ感染者の(簡易)実効再生産数の変化を示します。一方、真ん中の図は都心の人出の感染拡大前比です。

下の図は都心の人出率を横軸に、実効再生産数を縦軸とした散布図です。赤の点線で示すように人出が減少すると実効再生産数が小さな値をとる傾向が見えます。

そこで、回帰直線を引いてみると、人出が感染拡大前と比較して30%以上減少しているときには実効再生産数が1以下となっており、新規感染者数が減少していることが示されます。

【注意】 実効再生産数や都心人出は独立な観測値ではなく、時間的な相関がある時系列なので、厳密には後に学ぶ時系列データの分析法を適用すべきです。



重回帰分析

単回帰分析の例では、各県の千人あたりの感染者数を県庁所在地の人口密度で単回帰しましたが、感染者数に関連すると思われる変数は

県全体の人口密度
高齢者率
交通便利度

など、その他にもいろいろ考えられます。したがって、これらの変数を説明変数と考えて、目的変数を

$$y_n = a_0 + \underbrace{a_1 x_{n1} + \cdots + a_m x_{nm}}_{m \text{ 個の説明変数}} + \varepsilon_n$$

1番目の説明変数 m番目の説明変数 残差

のように、定数項 a_0 と m 個の説明変数 x_1, \dots, x_m の影響として表現するモデルが**重回帰モデル**です。単回帰の時と同様に a_1, \dots, a_m は回帰係数で、残差 ε_n は正規分布 $N(0, \sigma^2)$ に従うものと仮定します。

重回帰分析：都道府県別感染者数の例

以下ではコロナウィルスの都道府県別感染者数を例に考えてみます．説明変数の候補として，表のように x_{n1} から x_{n8} までの 8 変数を考えることにします． x_{n1} は定数項に対応します． x_{n8} は県庁所在地の人口密度 x_{n4} の 2 乗です．この変数は多項式回帰で 2 次のモデルが最もよかったことから，採用したものです．

これらの説明変数は，コロナウィルスの感染が人口密度や年齢あるいは人の移動に関連して発生しやすいと言われていることから，候補として採用したのですが，実際にはこのほかにもいろいろな変数を考えることができます．

表：目的変数と説明変数

y_n	コロナウィルスの累積患者数
x_{n1}	定数項
x_{n2}	県全体の人口密度
x_{n3}	高齢化率
x_{n4}	都道府県庁所在地の人口密度
x_{n5}	交通便利度(2時間)
x_{n6}	交通便利度(4時間)
x_{n7}	交通便利度(6時間)
x_{n8}	x_{n4}^2 (人口密度の 2 乗)

重回帰分析：説明変数の選択

m 個の変数について、モデルの変数として利用する場合と利用しない場合を考えると、全部で 2^m 種類のモデルがあります。現在の例では $m=8$ なので、256通りのモデルがありますが、**部分回帰**という方法を用いると、説明変数の数 $1, \dots, 8$ のそれぞれについて、すべての候補の中で誤差分散が最も小さなモデルを自動的に求めることができます。

AICで判断すると、表から x_1 (定数項), x_3 (高齢化率), x_4 (県庁所在地の人口密度), x_8 (人口密度の2乗) の **4変数を用いたモデル**

$$y_n = 16.3230 - 0.1611x_{n3} - 8.5340x_{n4} + 1.7072x_{n8} + \varepsilon_n$$

が最もよいことがわかります。

変数の数	残差分散	AIC	説明変数
1	1.241	147.53	x_8
2	0.460	102.89	x_3, x_8
3	0.403	98.62	x_1, x_4, x_8
4	0.312	88.70	x_1, x_3, x_4, x_8
5	0.309	90.26	x_1, x_3, x_4, x_5, x_8
6	0.303	91.22	$x_1, x_3, x_4, x_5, x_6, x_8$
7	0.287	90.63	$x_1, x_3, x_4, x_5, x_6, x_7, x_8$
8	0.272	90.21	$x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$

説明変数の選択（続）

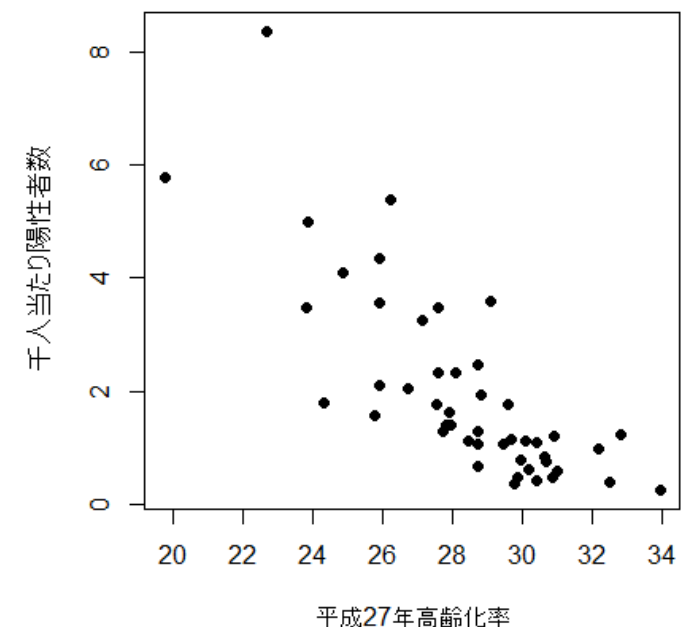
前頁のAICの値をみると、2次の多項式モデルに高齢化率を加えたモデルが最もよいことになります。

また、変数を1個だけ使うとすると人口密度の2乗 (x_{n8}) が最もよく、2個を使うと人口密度の2乗と高齢化率 (x_{n3}) がよいこともわかります。

ちなみに、このときのモデルは

$$y_n = -0.0849x_{n3} + 0.4573x_{n8} + \varepsilon_n$$

となり、高齢化率の回帰係数がマイナスですが、これは高齢化率が上がると患者数が減少するという因果関係を意味しているのではなく、図のように高齢化率が高い県では患者数が少ないという相関関係を表しているにすぎないことに注意しましょう。



2. ロジスティック回帰分析

ロジスティック回帰分析は $\{0, 1\}$ のようなカテゴリー(離散)変数やその観測値を集約した割合などを説明変数で表現する方法です。形式的に普通の回帰モデルをあてはめると不都合なことがおこるので、 $0, 1$ が発生する確率を考え、その非線形変換したものの回帰モデルを考えます。

本稿では最初に、観測値の非線形変換に対して普通の回帰モデルを適用する方法を考え、次に本来のロジスティック回帰モデルを考えます。

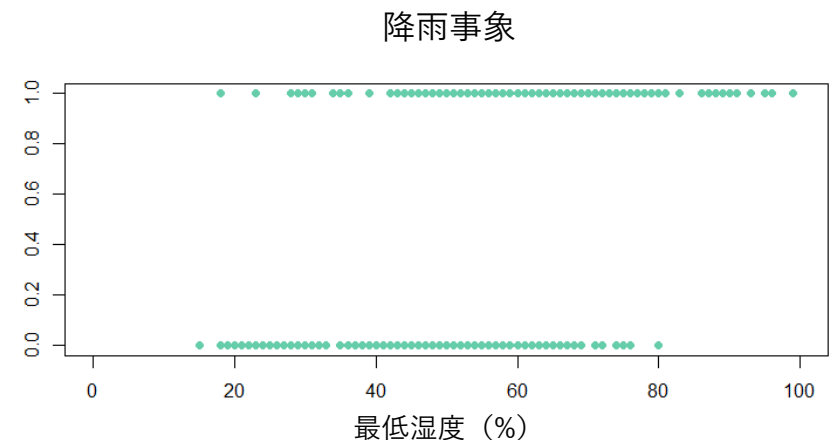
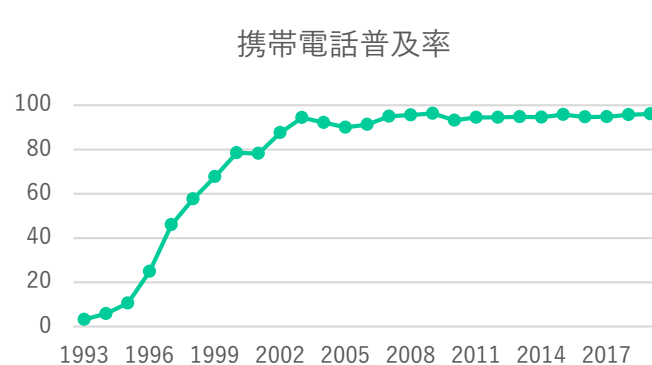
割合やカテゴリ変数のデータ

ロジスティック回帰分析は $\{0, 1\}$ のような離散変数やそれらを集約した割合などを説明変数で表現する方法です。

例として、携帯電話の普及率の推移と湿度と降雨の関係を考えてみましょう。

左図：1993年から2019年までの携帯電話（スマホを含む）の世帯普及率(%)の推移を示します。20世紀に急激に立ち上がってその後は100%近くに張り付いています。西暦を説明変数，普及率を目的変数とします。

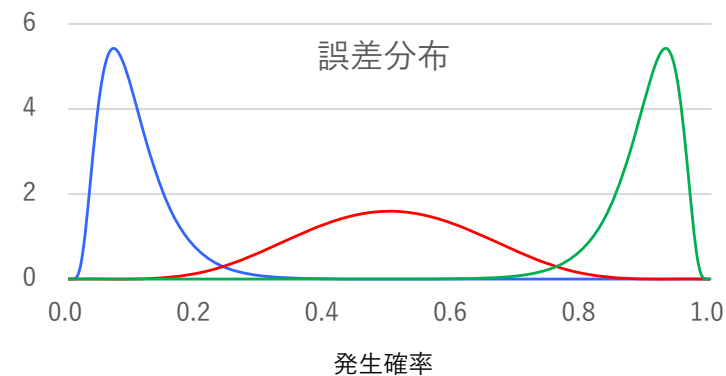
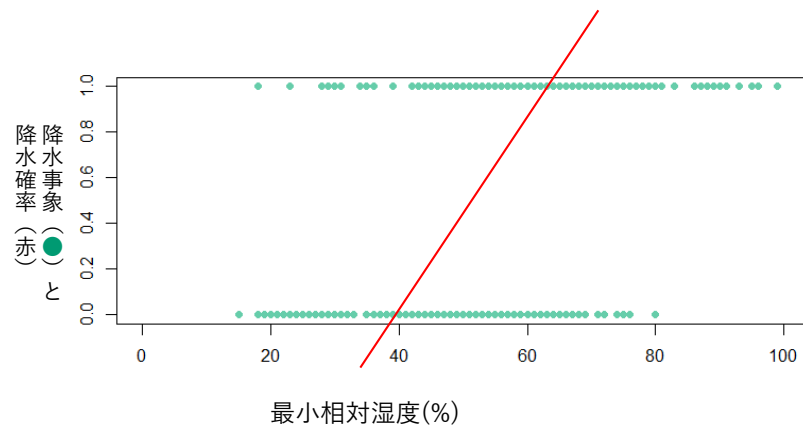
右図：東京の1年間の気象データ。横軸に1日の最低湿度(%), 縦軸は降雨で雨が降った場合は $y=1$, 降らなかった場合は $y=0$ の2値データ。湿度を説明変数，降水を目的変数とします。



回帰分析における問題点

割合データや予測誤差を定義できない離散変数の場合でもその事象が発生する確率を想定することによって，形式的には通常の回帰モデルを適用することは可能ですが，いくつかの問題があることを理解しておく必要があります．

1. 単純な単回帰モデルではその確率が0以下や1以上になるという不都合なことが起こります（左図）．
2. 割合データでは，特に発生確率が0と1の近辺で誤差分布が正規分布から大きく外れます（右図のように場所によって分布の形が変化します）．
3. 変動の分散が大きく変化します．



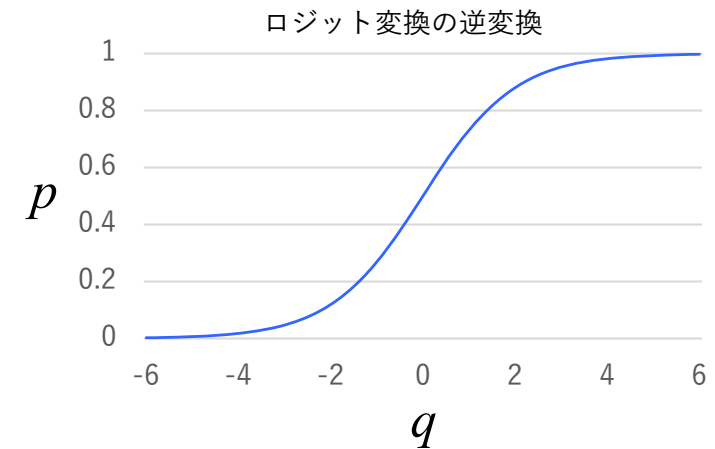
ロジット変換

まず，携帯電話普及率のように，説明変数（年）に対する割合 p_n が大量な観測数に基づいて得られている場合には， p_n をそのまま回帰分析するのではなく， p の **ロジット変換**

$$q = \log\left(\frac{p}{1-p}\right)$$

を定義しそのモデル化を考えます．逆に q から

$$p = \frac{e^q}{1+e^q}$$



で確率 p を求めると， q がどんな値をとっても p は常に 0 と 1 の間の値をとるので q の空間では自由なモデリングが可能となります．

【参考】 $p/(1-p)$ は **オッズ** と呼ばれ，競馬などのギャンブルでも利用されます．

ロジット変換（例）

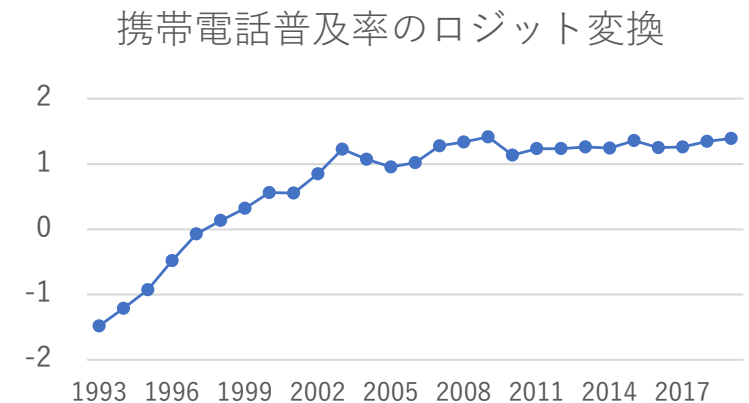
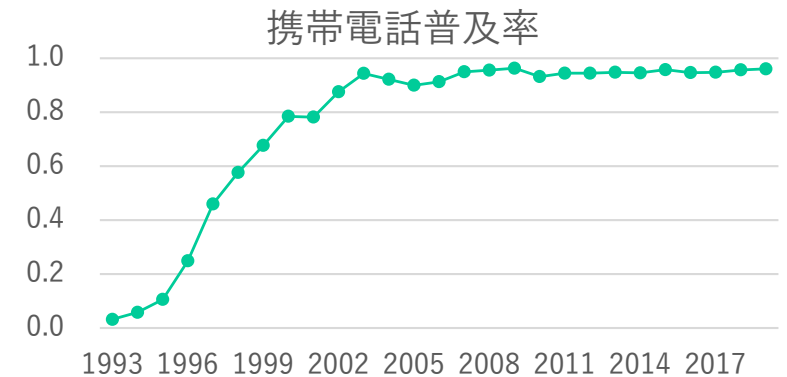
携帯電話普及率（上図）の場合には，ロジット変換によって，下図が得られます．このロジット変換 q は $-\infty$ から ∞ までの値をとれるので，説明変数 x_n を用いて

$$q_n = a_0 + a_1 x_n + \cdots + a_m x_n^m$$

と回帰できます．このとき， q の逆変換によって y が 1 をとる確率が

$$p_n = \frac{\exp(a_0 + a_1 x_n + \cdots + a_m x_n^m)}{1 + \exp(a_0 + a_1 x_n + \cdots + a_m x_n^m)}$$

で求められます．



ロジット変換（例2）

コロナウイルスの致死率（感染者のうち亡くなった人の割合）を考えます。致死率は年齢に大きく依存することが知られているので、横軸に20代以下、30代などの世代をとり、縦軸に致死率をプロットしてみると上図のようになります。

世代が増えるに従って致死率が上昇する傾向があきらかですが、このデータのロジット変換は中図のようになり、直線でかなりよく近似できます。

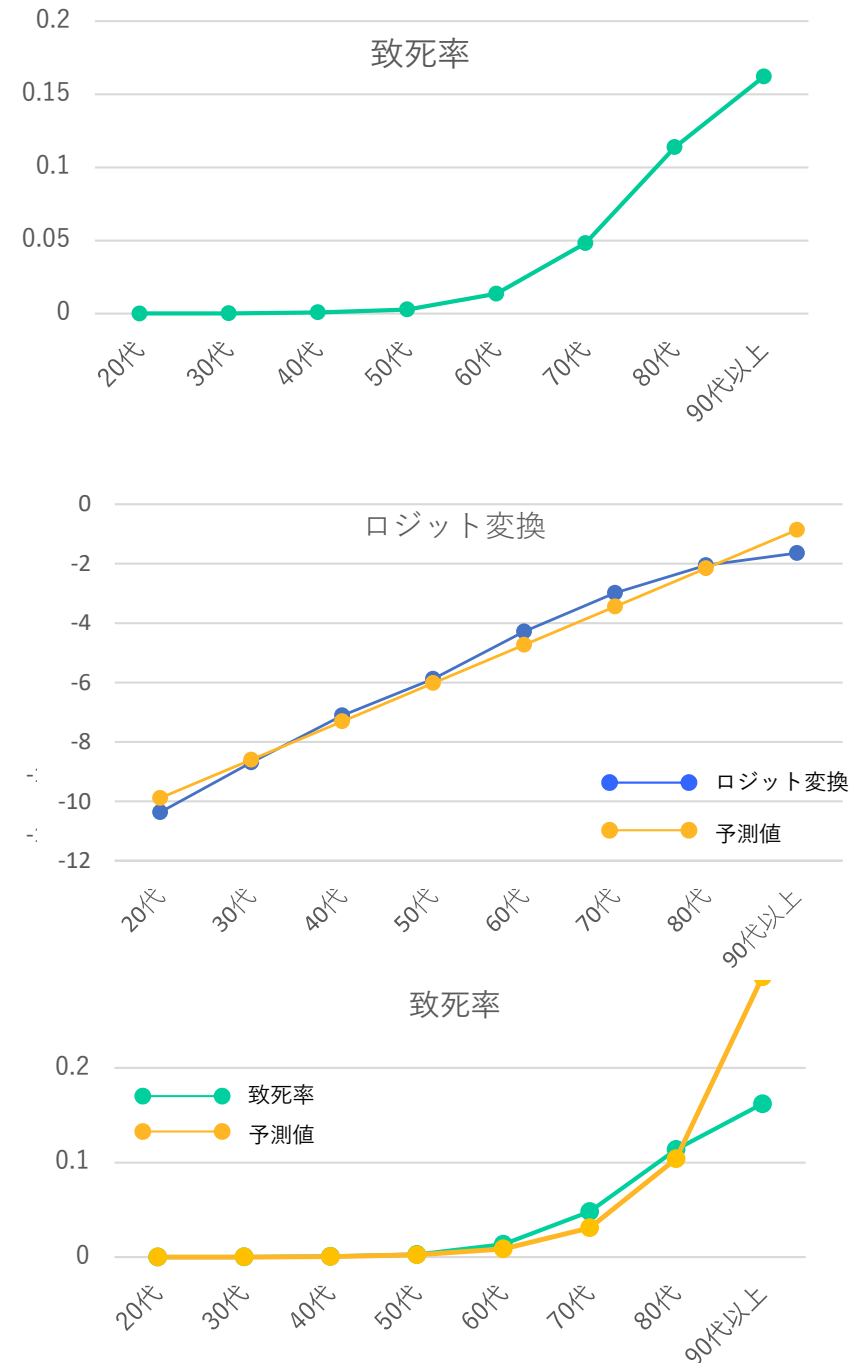
各世代を真ん中の年齢で代表させて単回帰を行うと

$$q_n = -13.108 + 0.129x_n$$

という直線(黄色) が得られます。したがって、

$$p_n = \frac{\exp(-13.108 + 0.129x_n)}{1 + \exp(-13.108 + 0.129x_n)}$$

で確率 p_n の予測式（黄色）が得られます。90歳以上ではデータよりもかなり大きな値が予測値になっています。



ロジスティック回帰モデル

N 組のデータ $(x_1, y_1), \dots, (x_N, y_N)$ があり, x_n は説明変数, y_n は病気に罹患している場合は0, いない場合は1となる離散変数とします.

0 か1の2値をとる変数に対して予測誤差は考えにくいので, 1という事象がおきる確率 p を考え, そのロジット変換 q を説明変数 x_n で表現するモデル

$$q_n = a + bx_n$$

を考えることにします. このとき

$$\log\left(\frac{p_n}{1-p_n}\right) = a + bx_n$$

なので, 事象1および0が起きる確率はそれぞれ次のように表されます.

$$p_n = \frac{\exp(a + bx_n)}{1 + \exp(a + bx_n)}, \quad 1 - p_n = \frac{1}{1 + \exp(a + bx_n)}$$

このように事象1と0が起きる確率を表現するモデルを **ロジスティック回帰モデル** と呼びます.

ロジスティック回帰モデルの尤度

ロジスティック回帰モデルを想定すると、説明変数 x_n が与えられているとき、 y_n の確率は

$$P(y_n | x_n, a, b) = \begin{cases} \frac{1}{1 + \exp(a + bx_n)} & y_n = 0 \text{ のとき} \\ \frac{\exp(a + bx_n)}{1 + \exp(a + bx_n)} & y_n = 1 \text{ のとき} \end{cases}$$

$$= \frac{\exp\{y_n(a + bx_n)\}}{1 + \exp(a + bx_n)}$$

【注意】 y_n は 0 か 1 なので上の式は
このように一つの式で書ける。

で与えられます。したがって、 N 組の独立なデータ $(x_1, y_1), \dots, (x_N, y_N)$ が与えられているとき**尤度**は以下のように定義されます。

$$L(a, b) = \prod_{n=1}^N p(y_n | x_n, a, b) = \prod_{n=1}^N \frac{\exp\{y_n(a + bx_n)\}}{1 + \exp(a + bx_n)}$$

次頁ではこの尤度を用いてパラメータ a, b を推定する方法を考えます。

最尤法

尤度の最大化によって、未知数 $\theta=(a,b)$ を推定する方法は**最尤法**と呼ばれます。ただし、通常は尤度の対数をとった以下の**対数尤度**を最大化します。対数関数は単調増加関数なので、どちらでも同じ推定値が得られます。

ロジスティック回帰モデルの場合、対数尤度は次のようになります。

$$\begin{aligned}\ell(a,b) &= \log L(a,b) \\ &= \sum_{n=1}^N \left\{ y_n(a + bx_n) - \log(1 + \exp\{(a + bx_n)\}) \right\}\end{aligned}$$

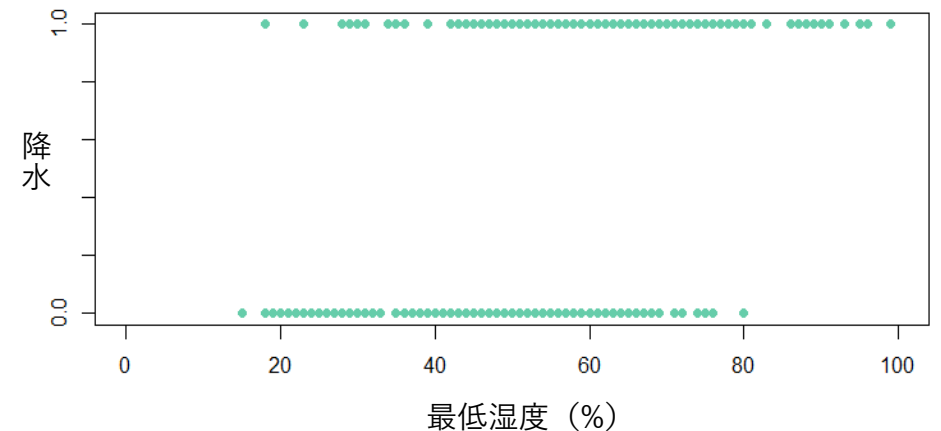
したがって、この対数尤度を最大化、すなわち

$$\max_{a,b} \ell(a,b) = \ell(\hat{a}, \hat{b})$$

となる \hat{a} と \hat{b} を求めることによって、係数の推定値が得られます。このように、最尤法で推定された係数を**最尤推定値**と呼びます。

例：降水確率

東京の1年間（2020年）の気象データを考えます．横軸に1日の最低湿度(%), 縦軸は降雨で雨が降った場合は $y=1$, 降らなかった場合は $y=0$ の2値データ．湿度を説明変数, 降水事象を目的変数として, 湿度からどの程度降雨を予測できるかを見てみます．



ロジスティック回帰モデルを推定してみると

$$\hat{a} = -5.2652$$

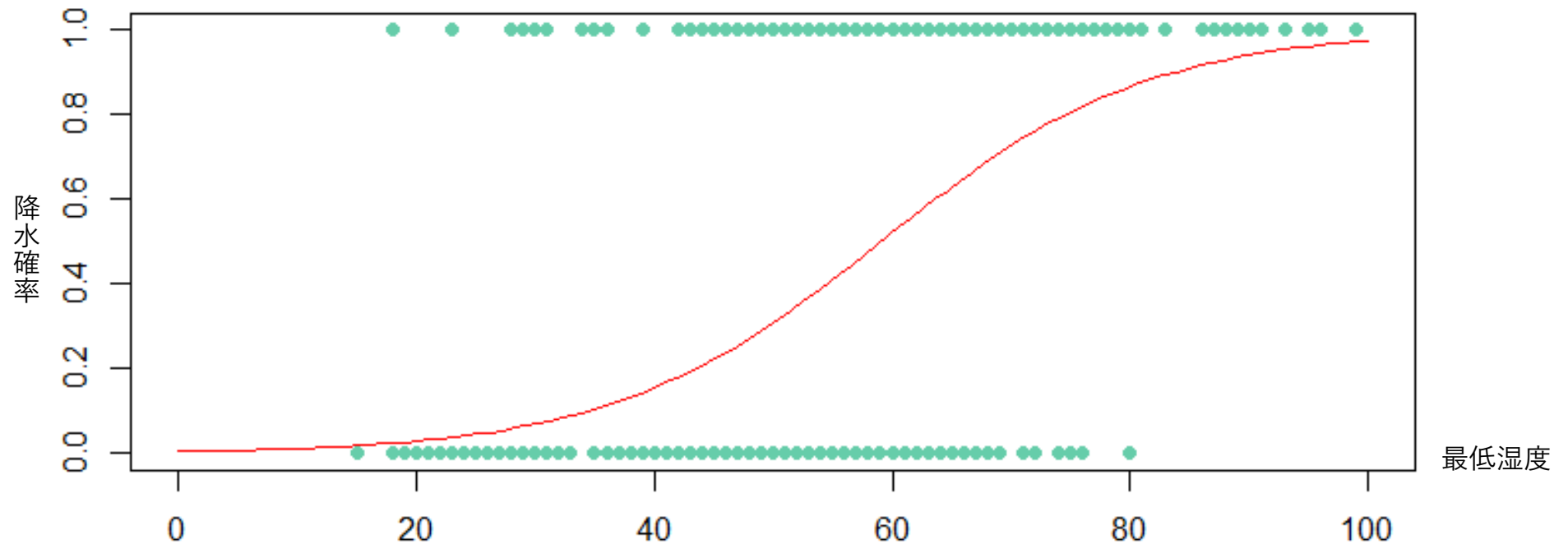
$$\hat{b} = 0.0892$$

という推定値が得られます．したがって, 最低湿度が x の時の降水確率は

$$p(x) = \frac{\exp(-5.2652 + 0.0892x)}{1 + \exp(-5.2652 + 0.0892x)}$$

となります．

例：降雨確率



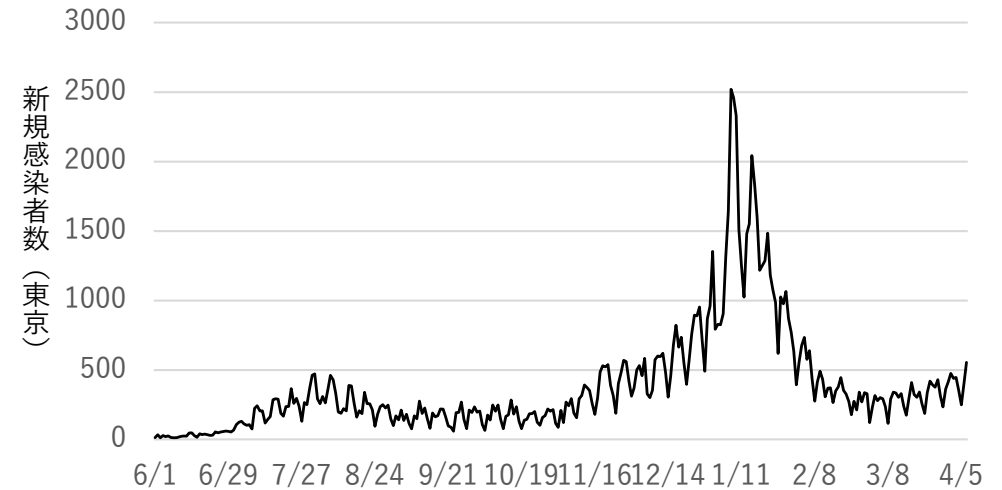
赤い曲線はロジスティック回帰モデルで最低湿度に基づき推定した東京の降水確率です。湿度が40%, 60%, 80%の時, 降水確率がそれぞれおよそ15%, 50%, 85%となることがわかります。この赤い曲線が縦に立っているほど予測精度が高いことになります。

3. 時系列分析

時間と共に変動する現象の記録が時系列です．ここでは，時系列データの簡単な分析の仕方を学びます．より詳しい説明がリテラシーレベル教材の4-4にあります．

時系列とは

図は横軸に月日，縦軸に東京の新型コロナウイルス新規感染者数を示します。このように時間と共に変化する現象を記録したものが**時系列データ**で，それを図示したものは**時系列グラフ**と呼ばれます。

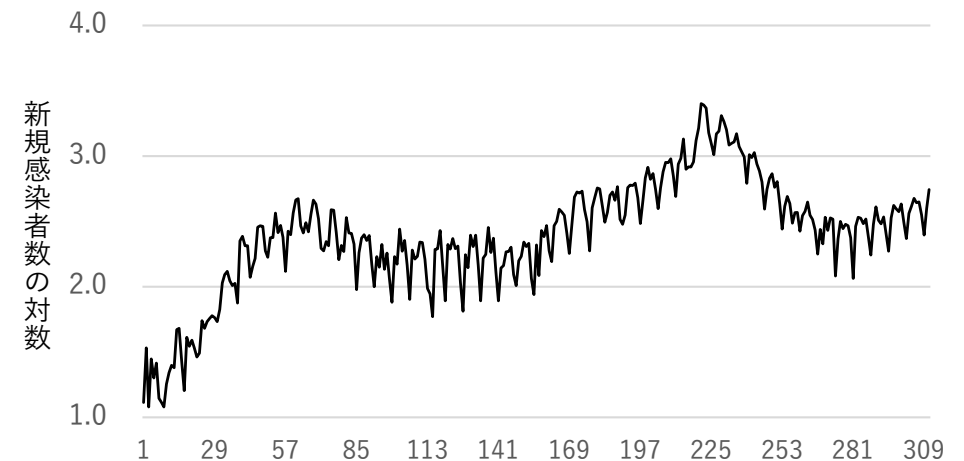


データが何番目に観測されたかを n で表し，そのときの観測値を y_n とします。 N 個の観測値 y_1, \dots, y_N が得られたとき， N は**時系列の長さ**あるいはデータ数と呼ばれます。

時系列分析の目的は，時系列の特徴を捉えたり，モデルを推定することによって，今後の変化を予測したり，意思決定に必要な情報を得ることです。そのための基本的なツールとして，以下では時系列の周期を捉える方法と時系列の傾向を捉える方法を考えます。

時系列を変換してみる

人数や金額などを数えたデータの場合は対数をとった方が特徴を捉えやすいこともあります。右図は新規感染者数の常用対数をとって表示したものです。

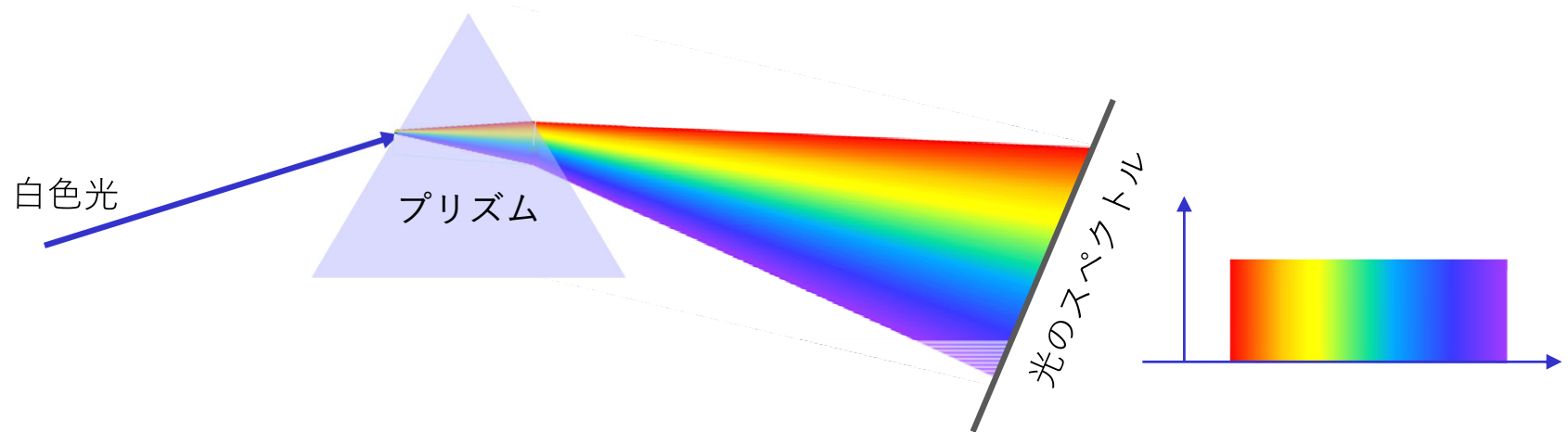


前ページの原データでは、時刻によって変動の幅が大きく変化していましたが、対数変換の結果ほぼ同じような変動幅となり、同じようなパターンで周期的に上下を繰り返していることが顕著になります。この周期性は、検査体制の影響で日曜日や月曜日の新規感染者数が少なくなるからです。

ただし、時系列の場合は完全な周期的変動を示すことは少なく、多くの場合周期的な変動パターン自体が徐々に変化します。このような時系列の周期的変動を可視化するツールとして、ピリオドグラムとパワースペクトルがあります。

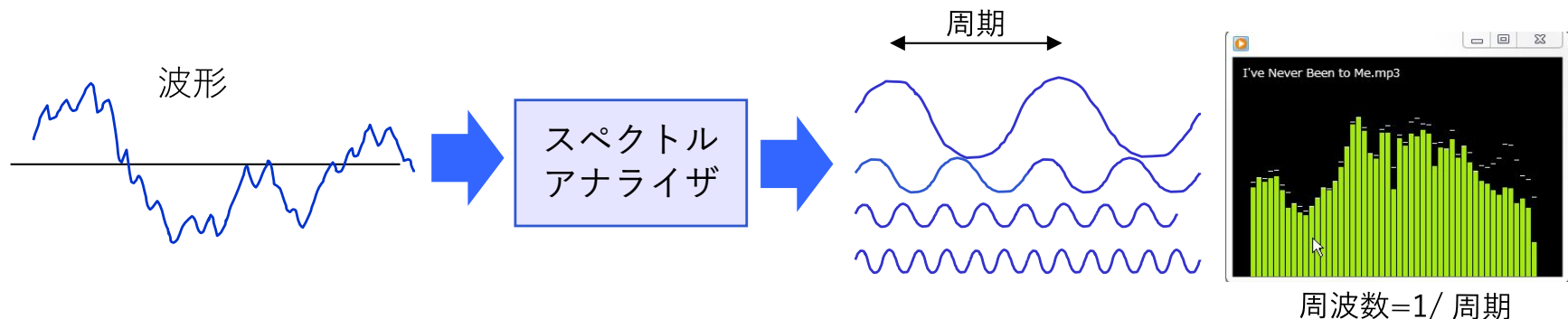
パワースペクトル

光をプリズムで沢山の色に分解できるように、時系列はサイン・コサインの和に分解できます。その強さを表示するとどの周期が強いかが分ります。



同じようにランダムな波形も周期関数の和で表現できます。

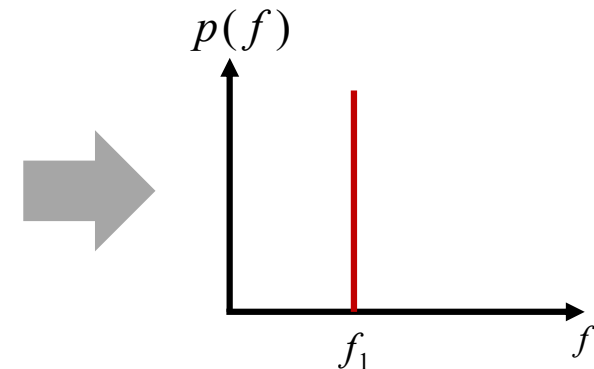
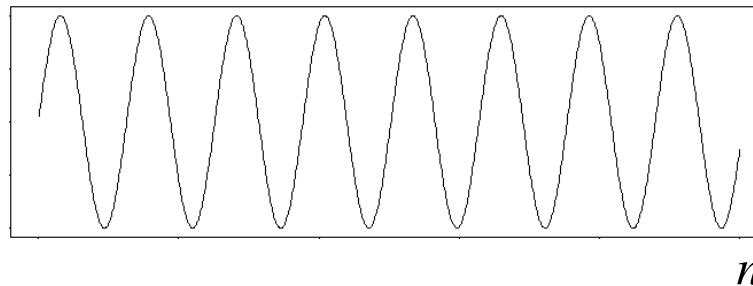
- スペクトルアナライザや音声のグラフィック・イコライザが例です



周期関数の場合

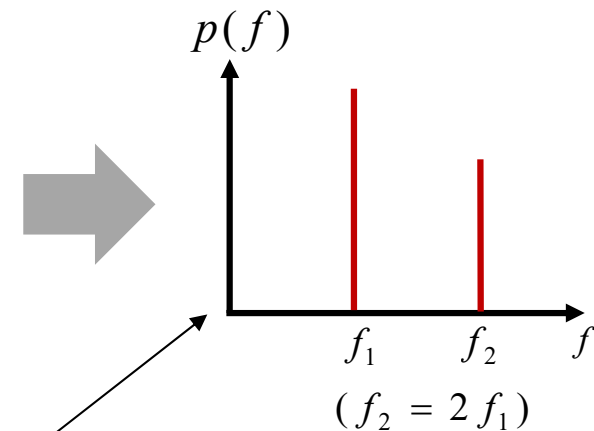
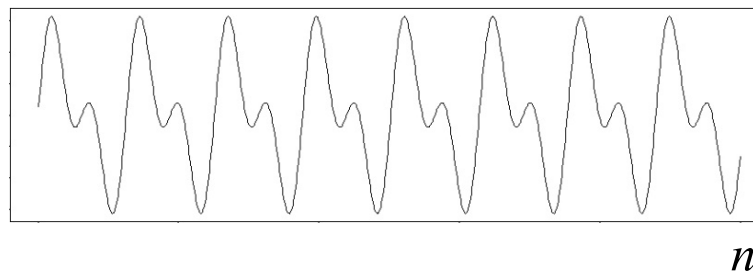
- サイン・コサインのようなひとつの周期関数で表現できる場合にはスペクトルは1点に集中します。

$$y_n = \sin(2\pi f_1 n)$$



- 2つの周期関数の和の場合はスペクトルは2つになります。

$$y_n = \sin(2\pi f_1 n) + 0.8 \sin(4\pi f_1 n)$$



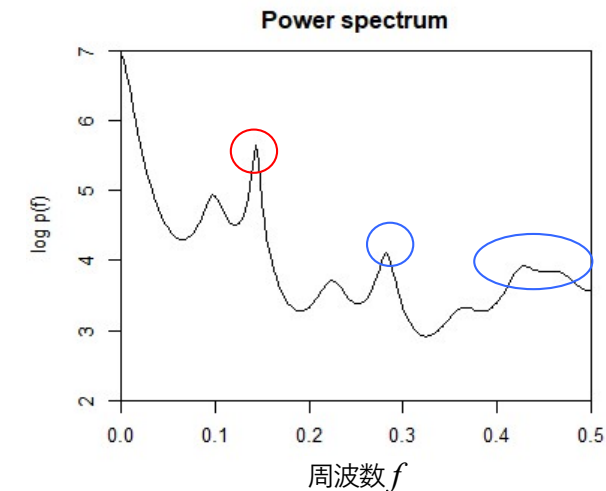
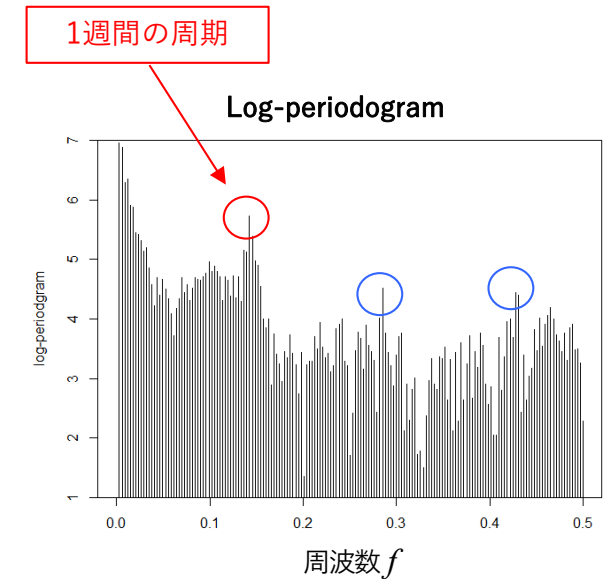
縦棒の高さは振幅の2乗に比例します。

例： Covid-19データの周期性

Covid-19 時系列のピリオドグラムの例です。横軸 f は周期の逆数で **周波数** と呼ばれます。また、縦軸は各周波数におけるピリオドグラムの対数を表します。 $0 \leq f \leq 0.5$ の範囲が図示されていますが、 $f = 0.5$ は 2日に 1回の周期成分、 $f = 0$ は周期無限大の成分です。ピリオドグラムが極大値をとる $f = 1/7 = 0.1429$ は1週間の周期成分に対応します。

ピリオドグラムからこのデータには顕著な1週間周期が存在することがわかります。また、 $f = 2/7$ と $3/7$ にもピークが見られますが、これは **高調波** と呼ばれるもので1週間の周期成分が単純なサイン・コサインではなく、複雑な波形であることを示しています。

ピリオドグラムに似たものとして **パワースペクトル** があります。パワースペクトルはピリオドグラムと違って連続関数になりますが、時系列モデルを推定することによって求められます。パワースペクトルでも同様な周波数にピークが見られます。



周期性の除去

- 周期性を除去する理由

Covid-19の場合について考えてみると，検査体制の事情から毎週日曜日や月曜日は小さな値になりますが，この事情を考慮しないと，患者数が減少したと誤った判断をしかねません．このような場合には，時系列の周期成分を除去してしまう方が簡単です．

- 周期性を除去する方法

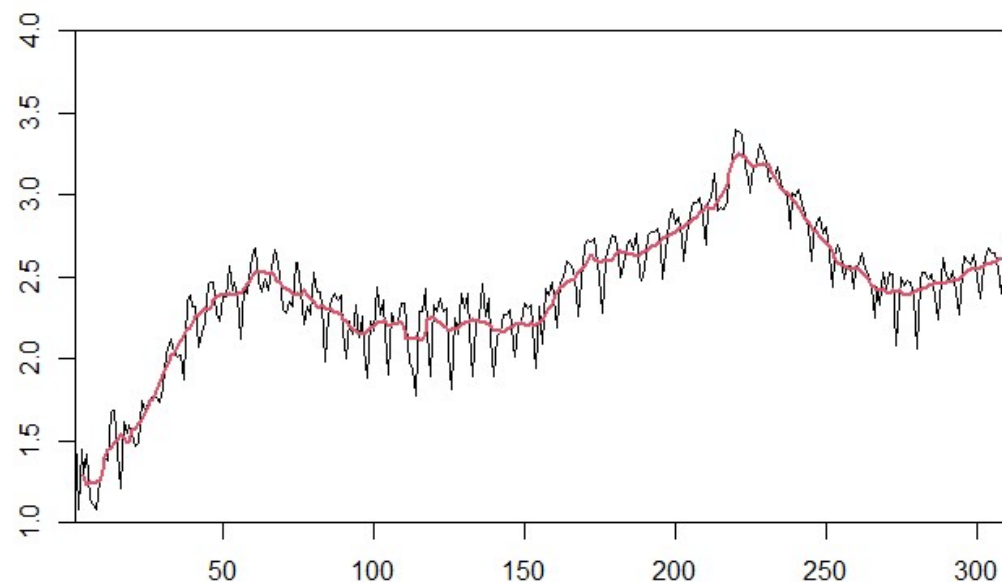
周期の長さが分っている周期成分を除去する簡単な方法として，移動平均があります．一般に，時系列 y_1, \dots, y_n が与えられるとき **(2k+1)項の移動平均**は

$$t_n = \frac{1}{2k+1} (y_{n-k} + \dots + y_n + \dots + y_{n+k})$$

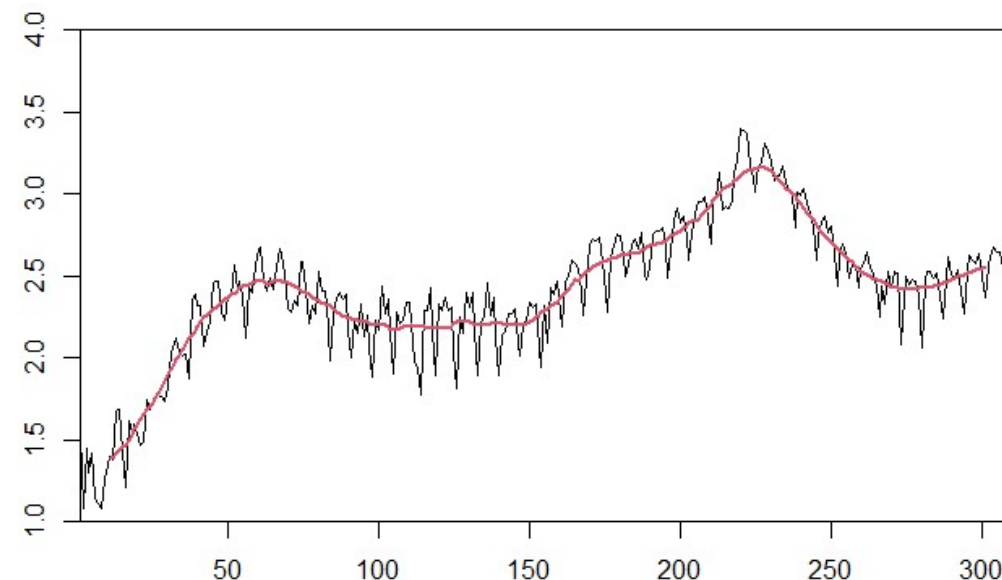
で定義されます．ここで， $k=3$ とすると項数は $2k+1=7$ となり，カッコ内には日曜から土曜までの各曜日が1回ずつ含まれるので，**曜日の影響を除去**できます．この方法では直近の t_{n-2} , t_{n-1} , t_n は計算できないので，上記の式で求めたものを現時点の推定値とみなして \tilde{t}_{n+k} と定義することもあります．ただし，この**片側移動平均**の場合は， k 時点だけ波形が遅れることに注意する必要があります．

例：Covid-19データの場合

上図の赤線は $k=3$ として求めた7項移動平均です。1週間の変動パターンがほとんど除去されて滑らかになり，感染者の増減の傾向がよく捉えられているようにみえます。



【参考】移動平均において $k=10$ とすると，前後の1週間を加え，合計3週間の平均をとることになり，さらに滑らかな移動平均が得られます(下図)．ただし，その反面 急激な構造変化を捉えにくくなるという副作用があり， $n=230$ 付近の急激な変化も滑らかになっている事には注意する必要があります．



季節調整法：より高度な方法

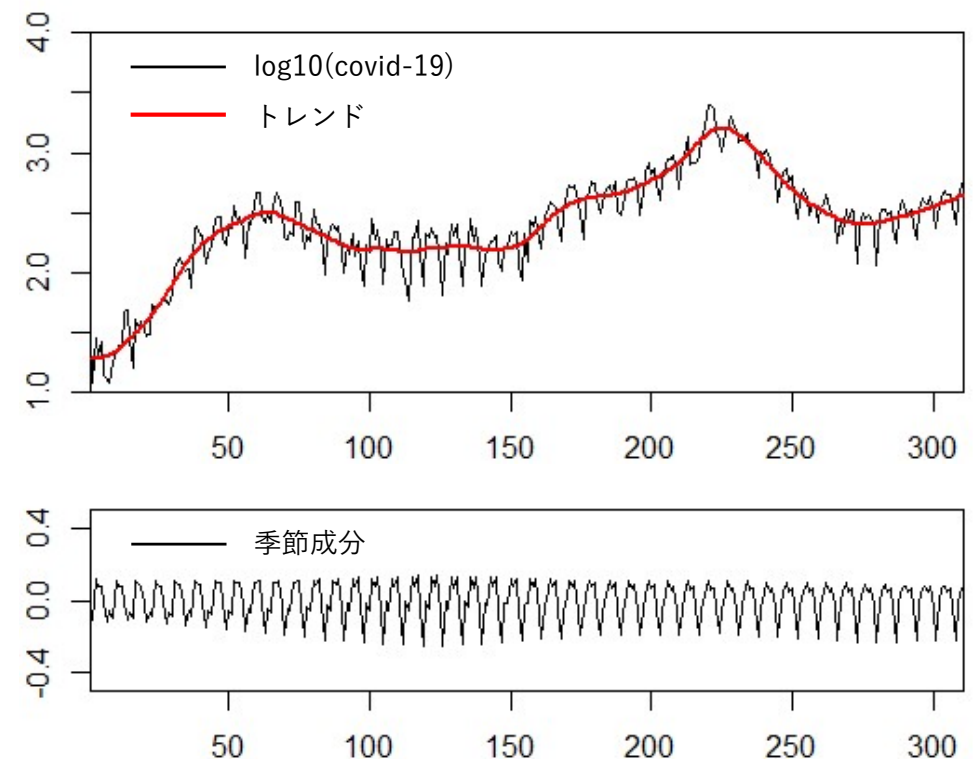
擬似周期的な変動を含む時系列の分析法として季節調整法があります。この方法では

$$y_n = T_n + S_n + w_n$$

と時系列をトレンド T_n 、季節成分 S_n 、ノイズ w_n の3成分に分解します。（季節調整法は年周期の除去を主要な目的とするため、1週間の周期の場合でも季節成分と呼ばれています。詳細はリテラシー教材の4.4節）

周期を $p=7$ 、トレンド次数を2として、この方法を適用すると、右図のような結果が得られます。

非常に滑らかなトレンドが得られ、1週間周期の変動が適切に除去されています。また、季節成分が時間の推移にともなって少しずつ変化していることもわかります。



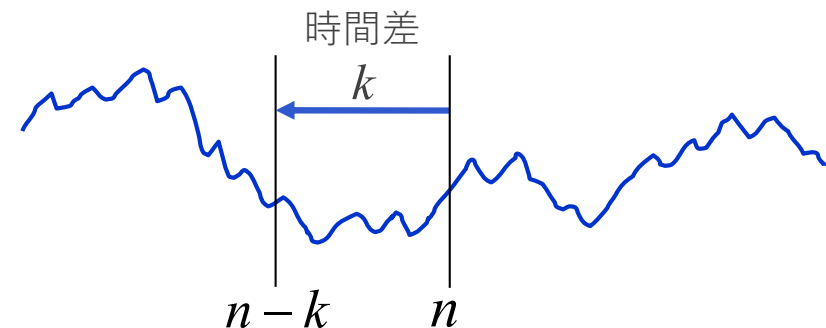
自己相関関数

周期的変動に限らず時系列の変動の特徴は自己相関関数で表現することができます。ある時刻と k だけ離れた時刻との相関係数を計算し

$$R(k) = \text{Cor}(y_n, y_{n-k})$$

と表します。 $R(k)$ は y_n と y_{n-k} の相関係数で k はラグ、 $R(k)$ は **自己相関関数** と呼ばれます。

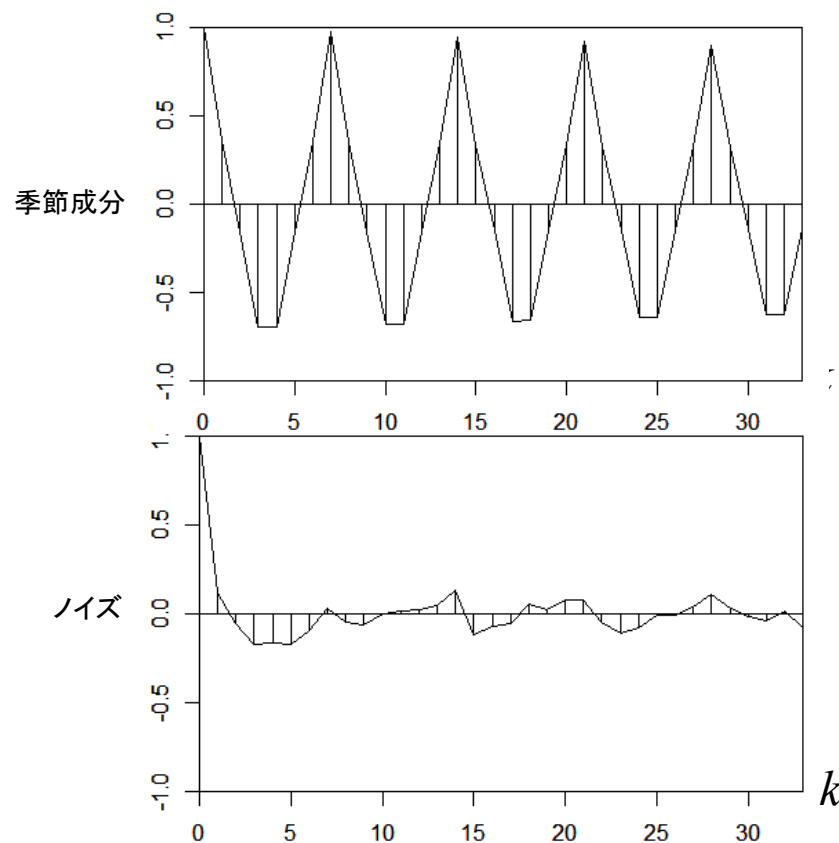
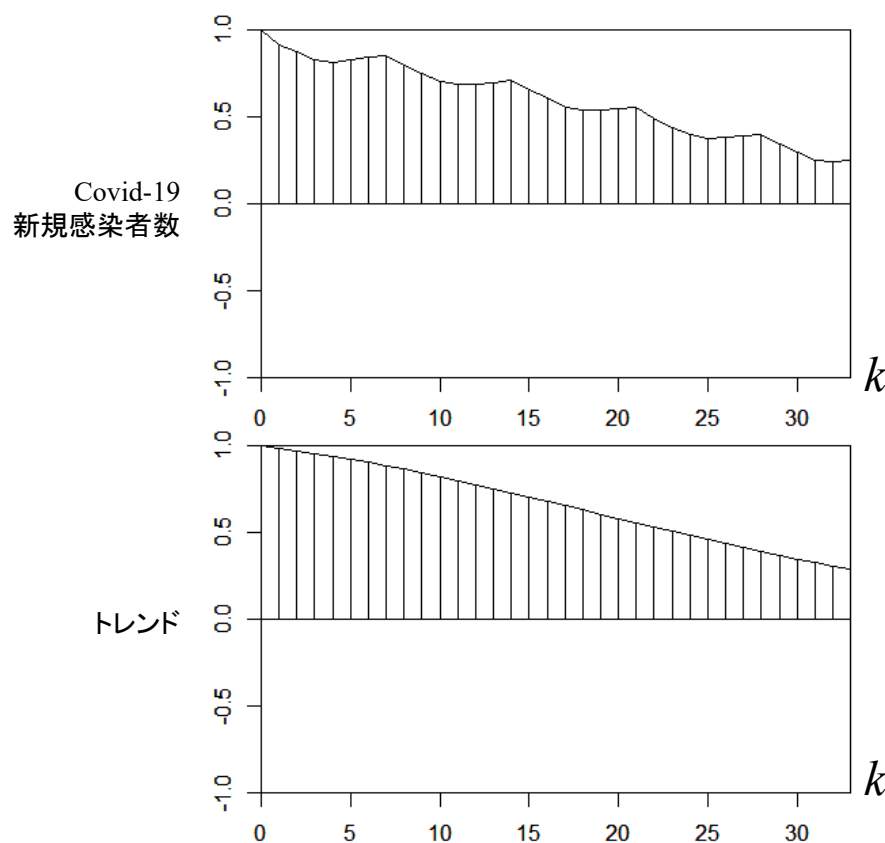
- $R(k)$ は k だけ離れた地点との相関の大きさを表します。
- y_n と y_{n-k} に正の相関があるとき $R(k) > 0$ となります。
- y_n と y_{n-k} に負の相関があるとき $R(k) < 0$ となります。



例：Covid-19 データの自己相関関数

Covid-19データのラグ32までの自己相関関数を示します。

- 左上は原データの自己相関関数で、7日周期で変動しながらゆっくり減衰しています。
- 左下は季節調整で得られたトレンドの場合で、非常にゆっくり減衰しています。また原データの変動の主要部分がトレンドであることもわかります。
- 右上は季節成分で7日周期でほぼ規則的に変動しています。
- 右下はノイズ成分で、ほとんど自己相関がない系列であることがわかります。



時系列の予測

時系列分析の重要な応用が**予測**です。時刻 n までの時系列 y_1, \dots, y_n が与えられたとき、将来の値 y_{n+1}, \dots, y_{n+k} を推定する問題が予測です。とくに1時点先の y_{n+1} を推定する問題を**1期先予測**、 y_{n+j} (j は2以上)を推定するときは**長期予測**と呼びます。

● ARモデルによる予測

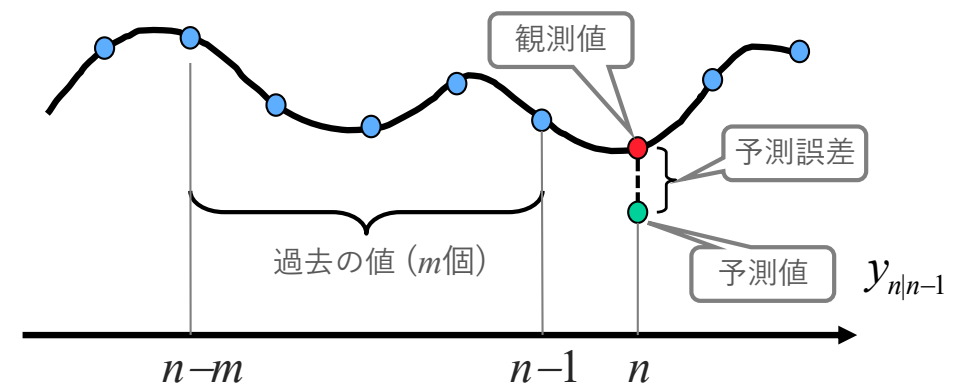
代表的な予測方法はARモデルを用いる方法で、時刻 n までのデータを使って**ARモデル**

$$y_n = a_1 y_{n-1} + \dots + a_m y_{n-m} + \varepsilon_n$$

が推定できると、 y_{n+1} の予測値は

$$y_{n+1|n} = a_1 y_n + \dots + a_m y_{n-m+1}$$

となります。以下、逐次代入により2期先以上の予測値も求められます。



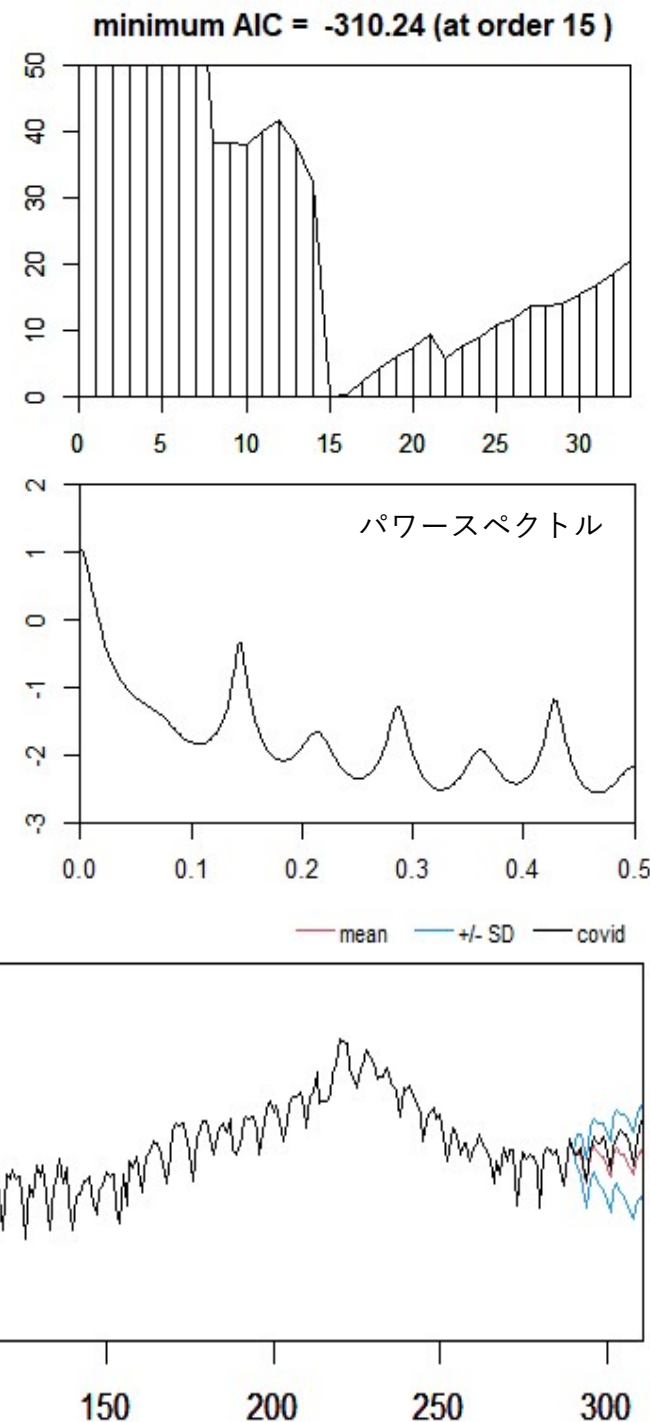
例：Covid-19データの予測

まず $n=260$ までのCovid-19データからARモデルを推定し、そのモデルを使って予測した場合を示します。

上図のAICで選択した結果15次のARモデルが得られます。真ん中の図は、そのモデルで推定したパワースペクトルです。7日とその1/2および1/3の周期が目立っています。

下図はこのモデルによる長期予測結果で、予測値及び標準誤差を示します。黒色で示す実際の値を比較的よく予測していますが、誤差幅が広がっています。

ARモデルによる予測は便利ですが、このデータのように明らかなトレンドがある場合には、トレンドを分離する方がよい結果が得られます。



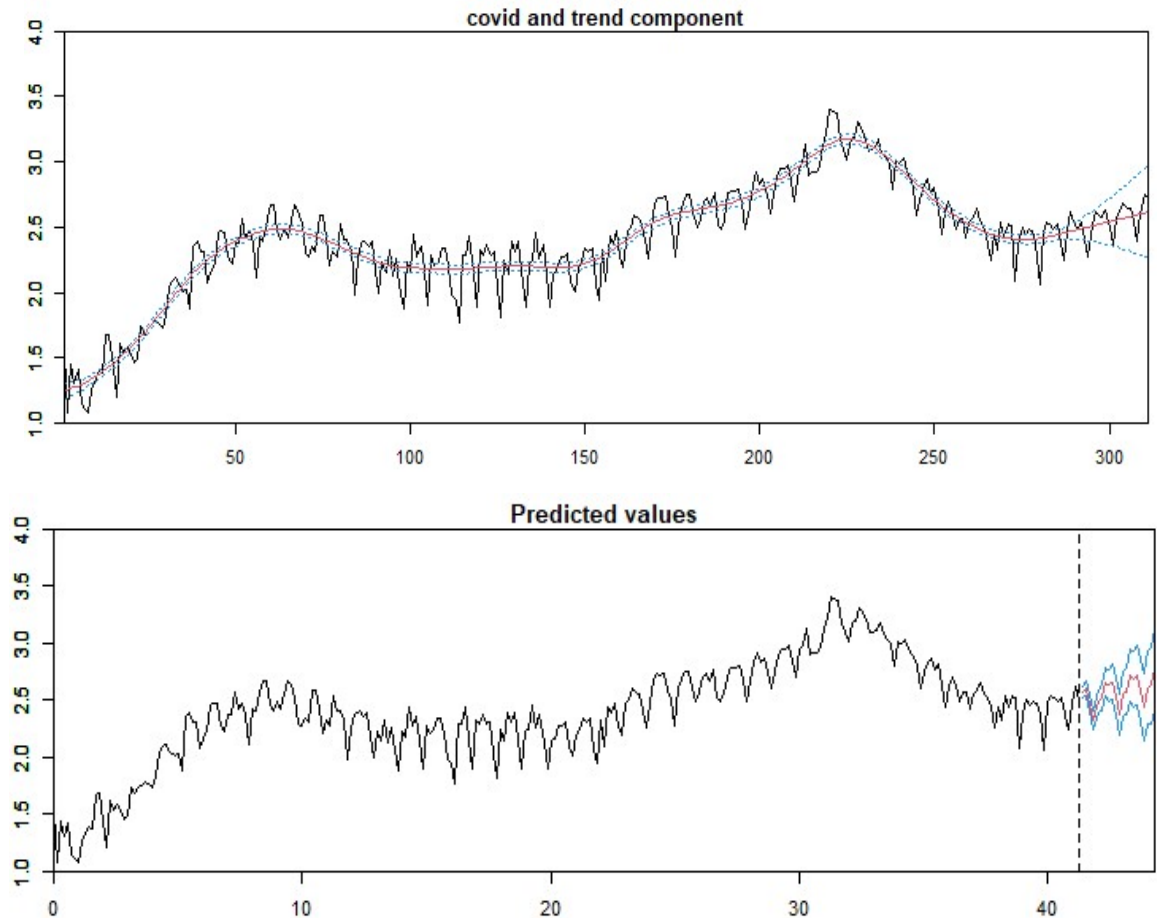
季節調整モデルによる予測の例

季節調整法によるトレンド，季節成分，ノイズへの分解を利用した長期予測の結果を示します。

トレンドは非常に滑らかなので長期予測も安定して実施でき，予測誤差も比較的小さくなります。また，季節成分も安定しているので，その予測誤差も比較的小さくなります。

結果として，この二つを加算した時系列の予測値の予測誤差もARモデルの場合よりも小さく，また周期パターンもよく再現されています。

コロナウイルス感染者数の予測のためにはこの予測値 p_n から 10^{p_n} を計算すればよく，1週間周期の変動も考慮した予測値が得られます。



4. クラスター分析

説明変数の情報を利用せず、データを似たもの同士に分類するのが**クラスター分析**です。どんなデータを似ていると考えるかは、2点間の距離やクラスター間の距離（類似度）の定義に反映されます。距離の定義によって結果がかなり異なることがあることは注意する必要があります。

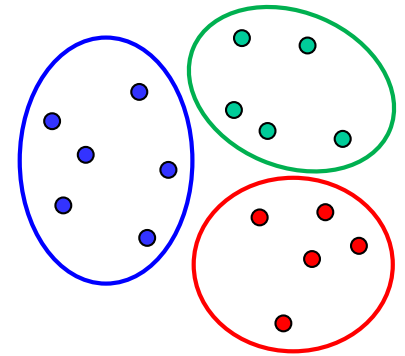
二つの代表的なクラスター分析の方法として、階層的クラスター分析と非階層的クラスター分析があります。

クラスターとは

k 次元のベクトルで表されるデータを考えます.

$$x_n = (x_{n1}, \dots, x_{nk})$$

例えば, n 番目の学生の数学, 物理, 生物, ..., 英語の成績などです. 学生が N 人いると N 個のベクトル x_1, \dots, x_N が得られます.



k 次元空間の似た点の集まりを**クラスター**と呼びます. クラスター分析では, データから図の様に近いものを一つのクラスターにして, いくつかのグループに分けます. ただし, いくつかのクラスターが存在するか前もってわかっているわけではないので, 2点間の近さやクラスター間の近さをどのように定義するかによって結果が変わってきます.

クラスター間の距離

二点間の距離としては通常、次を示すユークリッド距離を使います。

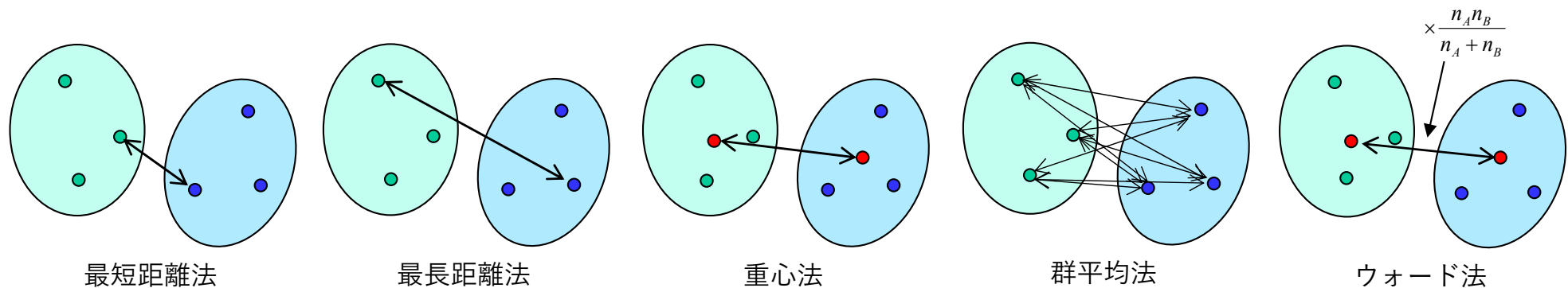
$$x_n = (x_{n1}, \dots, x_{nk}), \quad y_n = (y_{n1}, \dots, y_{nk})$$

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + \dots + (x_k - y_k)^2}$$

- ただし、距離の定義としては、このほかマンハッタン距離、ミンコフスキー距離、チェビシェフ距離などがあります。

クラスター間の距離（類似度）

クラスター間の距離の定義としては、最短距離法、最長距離法、重心法、群平均法、ワード法などいろいろ提案されており、どれを選ぶかによってクラスター分析の結果が異なります。



階層的クラスタリング

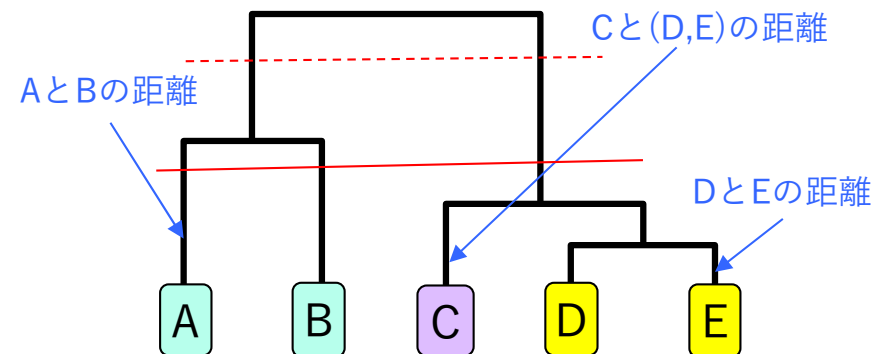
階層的クラスタリングでは、最初は全てのデータが異なるクラスターに属するものとし、
(クラスターの数) = (データ数) とする状態を考えます。

次にすべてのクラスター間で最も距離が小さな2つを探し、それらを一つのクラスターに統合し、クラスター間の距離を更新します。以下、このステップを繰り返すことによってボトムアップにひとつずつクラスター数を減らしていきます。

このクラスター統合のときにU型にふたつのクラスターを繋ぎ、縦の高さをクラスター間の距離とします。この操作を続けていくと、最終的にクラスターがひとつになると、
樹形図（デンドログラム）ができます。

既に樹形図が与えられている場合には、横線を引いて上からだんだん下げていくと2クラスター、3クラスターの順に分類ができます。

右図の場合は2クラスターの場合は、{A,B}, {C,D,E}, 3クラスターの場合は、{A}, {B}, {C,D,E}であることがわかります。



移動距離データの例

デンドログラム作成の説明のために東北・北海道の県庁所在地間の移動時間が距離として与えられている場合を想定し、最長距離法で階層的クラスタリングを行ってみます。

- ① 福島と宮城を併合：距離42分
- ② 山形と1の2県を併合：距離76.5分
- ③ 青森と岩手を併合；距離88分
- ④ 秋田と3を併合：距離170分
- ⑤ 2と4を併合：距離251分
- ⑥ 北海道と5を併合：距離373分

	北海道	青森	岩手	宮城	秋田	山形	福島
北海道	0						
青森	313.5	0					
岩手	290.5	88	0				
宮城	253.5	146.5	61.5	0			
秋田	373	170	97	153.5	0		
山形	319	247.5	161	76.5	251	0	
福島	283	193.5	115.5	42	205.5	72.5	0

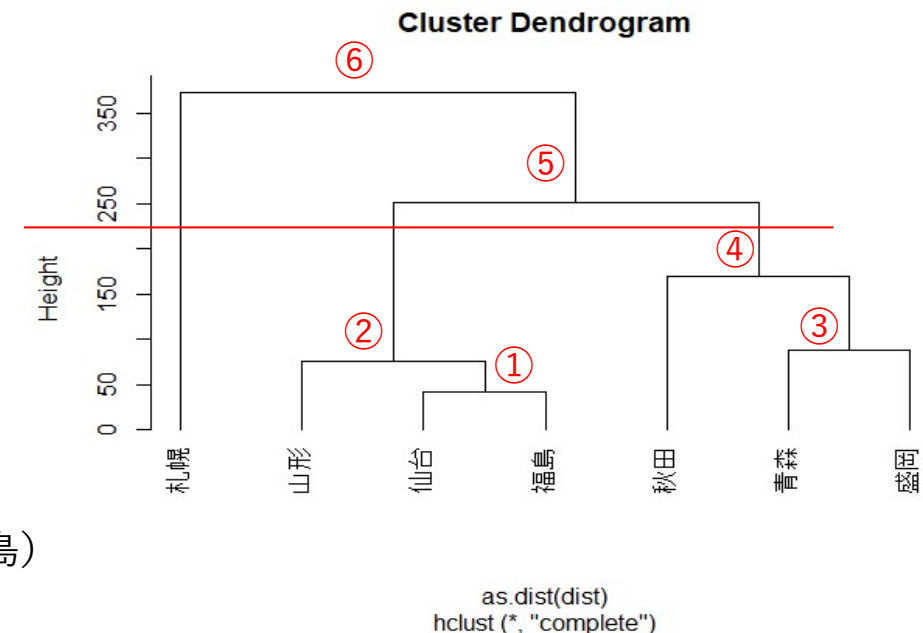
(移動距離：県庁所在地のJR駅を朝8時に出発した時の到達時間)

結果として、右下の樹形図が得られます。

①～⑥は併合していった順番を示します。

逆に樹形図が与えられると、横線を上からおろしていくと、次のように2つ～5つのクラスターが順番に得られます。

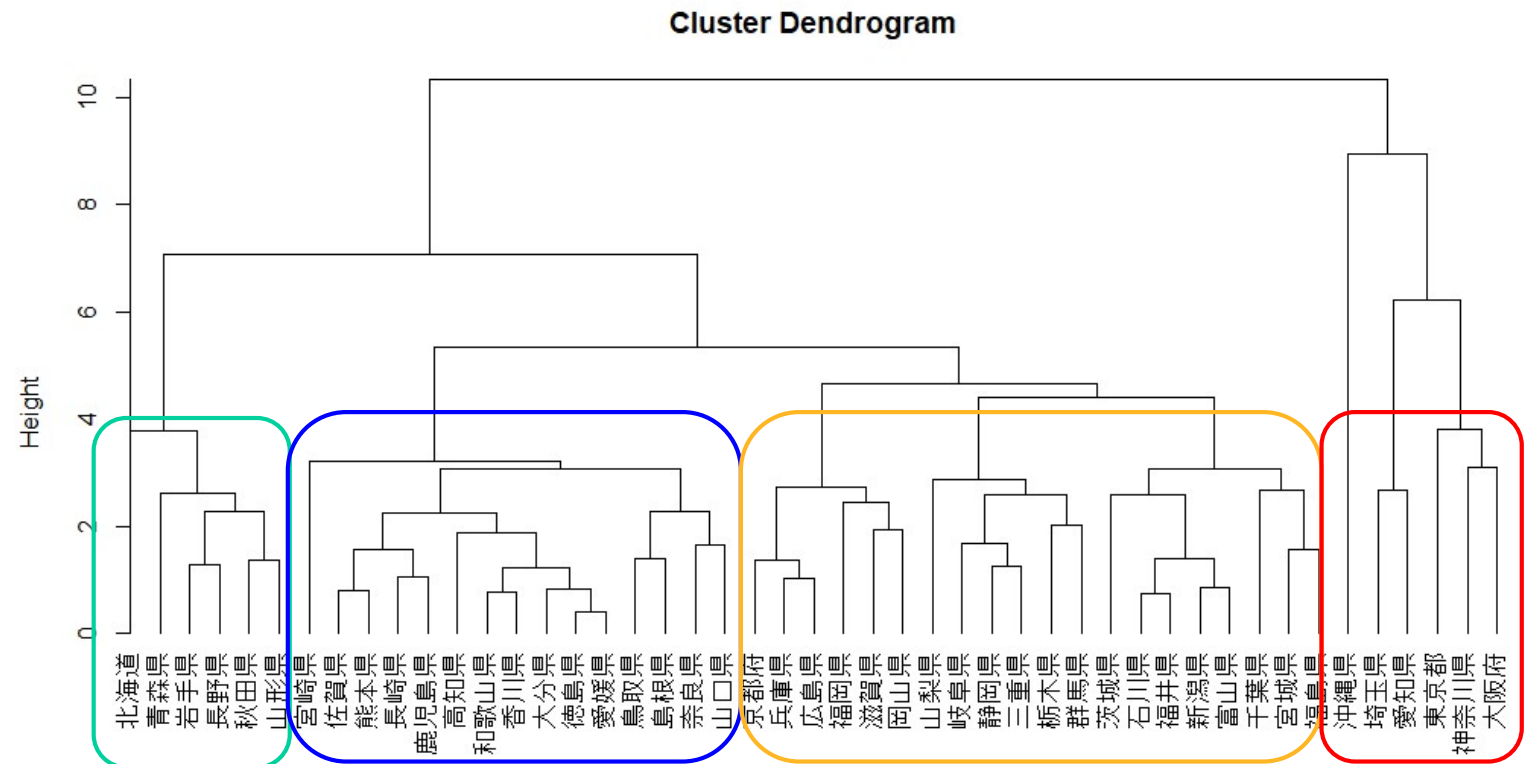
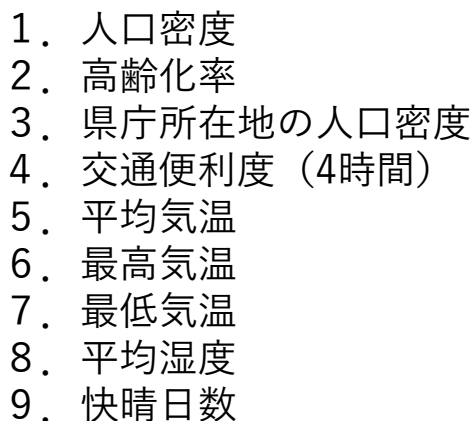
- 2 (北海道) (青森, 岩手, 宮城, 秋田, 山形, 福島)
- 3 (北海道) (青森, 秋田, 岩手) (宮城, 山形, 福島)
- 4 (北海道) (秋田) (青森, 岩手) (宮城, 山形, 福島)
- 5 (北海道) (秋田) (青森, 岩手) (山形) (宮城, 福島)



例：多変量データの場合

重回帰モデルの説明変数として用いたデータを拡大した9変量（下記）のデータに対して階層的クラスタリングを適用した計算手順と結果を示します。

1. 9 変量のデータをそれぞれ、平均0, 分散1に変換して規準化します.
2. 9 つの基準化したデータの距離を計算します.
3. クラスタリングを実施します.



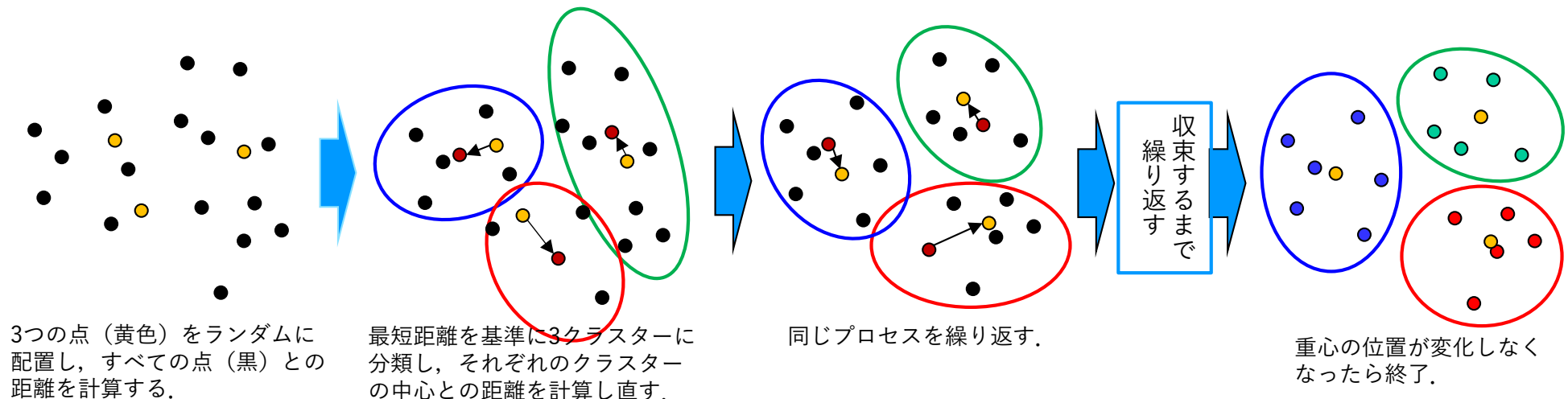
非階層的クラスタリング

階層的クラスタリングは全データバラバラの状態から、一つずつクラスターを形成していき、また最終的に樹形図ができるので、直感的にもわかりやすいです。

しかしながら、この方法ではまず全データ間の距離を計算する必要があり、またクラスター間の距離の定義によっては、毎回距離を更新していく必要があるために、巨大なデータに適用することは困難です。

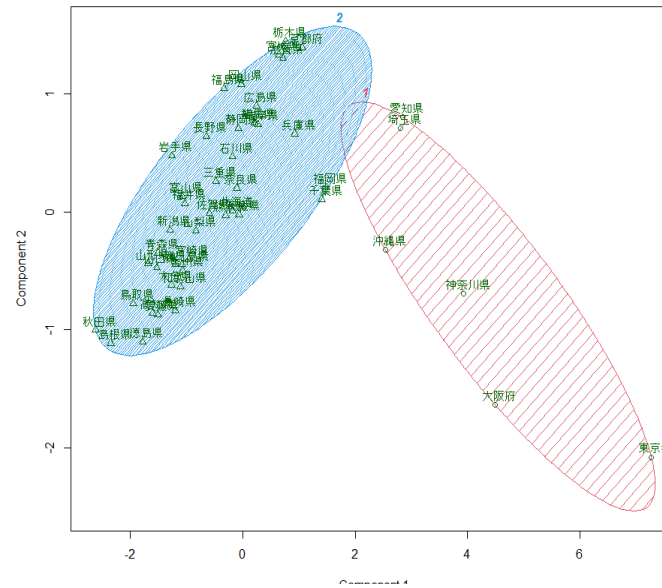
非階層的クラスタリングの代表的な k -means法では、樹形図（階層的な構造）を想定せず、予め定めたクラスター数について、適当に定めた初期クラスターから繰り返し計算によって、与えられた距離を最小にするようにクラスターを求めます。

3 クラスターの場合

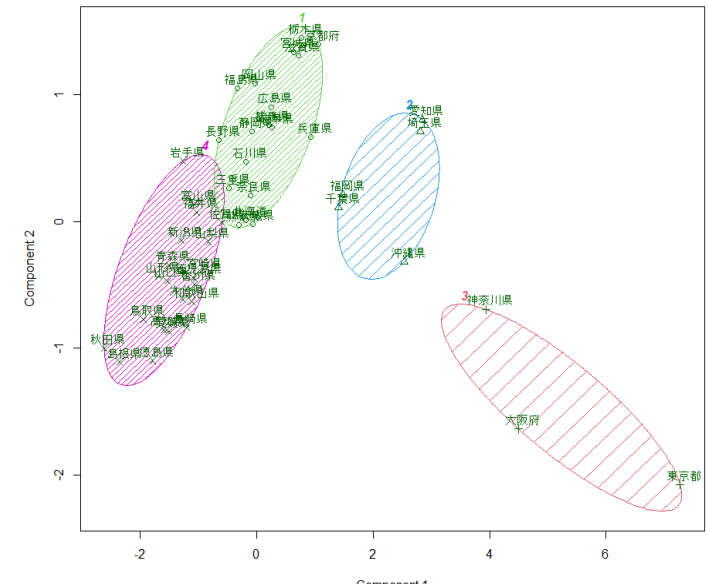


非階層的クラスタリングの例

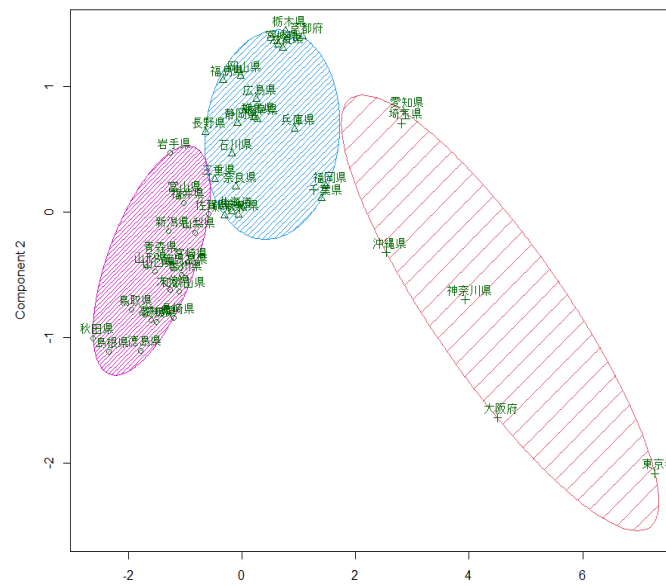
$k=2$ の場合



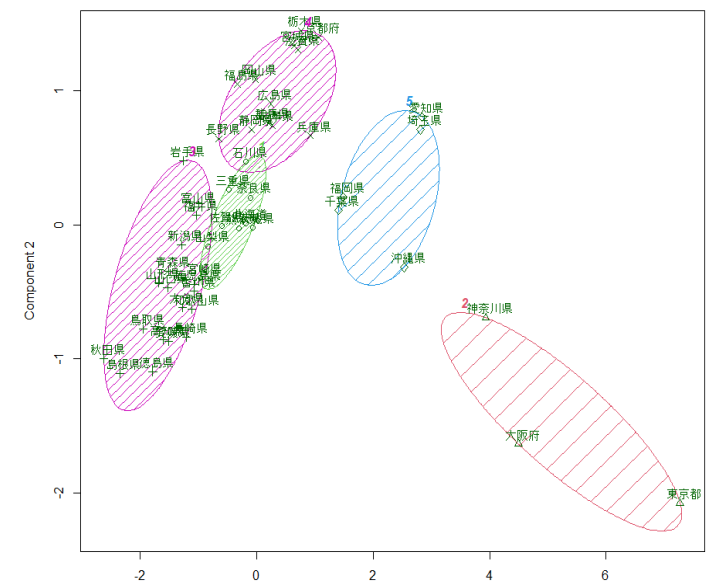
$k=4$ の場合



$k=3$ の場合



$k=5$ の場合



階層的クラスタリングでも用いた多変量データのうち1, 3, 4, 5, 6番目の5変量データに k -means法を適用した結果です. k としては2, 3, 4, 5の4つの場合を示しています. 右図では5次元のうち, 第1主成分と第2主成分だけを表示しています.

大雑把にいうと沖縄を除き, $k=2$ の場合は首都圏・近畿・中部の人口密度が高い県とそれ以外にグループ化されています. $k=3$ ではそれ以外が2つに分けられ, $k=4$ の場合は, さらに大都市圏が2分割されています.

These two components explain 83.42 % of the point variability.

These two components explain 83.42 % of the point variability.

非階層的クラスタリングのまとめ

非階層的クラスタリングのメリット

- 全サンプル間の距離を計算する必要がなく、仮定した重心と全サンプル間の距離だけを計算すればよいので、階層的クラスタリングよりも計算量が少なく、ビッグデータにも適用できます。

非階層的クラスタリングのデメリット

- 最初にランダムに指定する重心の位置によって、最終的結果が変わることがあります（初期値依存性）
- 何個のクラスターに分けるのがよいかの明確な判断基準がありません。

5. パターン発見

POSデータなどの販売・取引に関するデータから意思決定に役に立つ規則や注目すべきパターンを発見する方法としてアソシエーション分析があります。支持度，確信度，リフト値などの概念と相関ルール抽出や頻出パターン発見の方法としてのAprioriアルゴリズムを紹介します。

POSデータからのパターン発見

コンビニなどの店舗で買い物をするとレジで様々な情報が入力されており **POS** (Point of Sale) データが得られます。POSデータはチェーン店などを中心に販売戦略に活かされています。

POSデータのうち、どのような商品と一緒に購入したかの部分だけからも重要な情報が得られることがあります。例えば、下記のような販売データがあったとします。このような販売データが大量にあったとき、その中から販売促進に役に立ちそうな情報を取り出す方法が**アソシエーション分析**です。バスケット分析と呼ばれることもあります。

ID	バスケット
1	{A, B, C, D}
2	{A, B, E}
3	{B, C, F, G}
4	{D, E, F, G, H}
5	{B, C, E, H}

パン、ジュース、牛乳などの商品は**アイテム**と呼ばれます。一人のお客はいくつかのアイテムを購入するので、それを $\{A, B, C, D\}$ のように表して**バスケット**と呼びます。また、 $I = \{A, B\}$ のようにアイテムを集めたものを**アイテム集合**と呼びます。

多くのIDのバスケットの中に同じアイテム集合（例えば I と J ）が現れる場合、 I と J は一緒に買われる可能性が高いことがわかるので、並べて陳列するなどの販売促進策が考えられます。アソシエーション分析によって、紙オムツとビールが頻出アイテムとして見出されたというのが有名な例です。

アソシエーション分析

アソシエーション分析では、膨大な販売データに見られる商品Bを購入すると商品Cも買う傾向があるというような、関連性や規則性、相関ルール (association rule) を見出すことを目的とします。

販売データは以下のような情報を持ちます。

ID 通し番号, 何番目の購入(バスケット)かを n で表す

$j(n)$ n 番目の客が購入した商品数

I_n $\{I_1, \dots, I_{j(n)}\}$ n 番目のバスケット内のアイテム

アソシエーション分析では、全バスケットの中でアイテム集合 I が購入されたときに、アイテム集合 J が購入される傾向があるときに $I \Rightarrow J$ と表現します。

アイテムとアイテム集合

相関ルールではアイテム集合 I と J について、 $I \Rightarrow J$ という関係を考えます。ただし、 I と J は $\{A\}$ のような単独のアイテムの場合も $\{A, B, E\}$ のような複数のアイテムの集合の場合もあります。

相関ルールは I ならば必ず J が起こるというルールではないので、次頁のように**支持度**、**確信度**、**リフト値**などの指標が考慮されます。

これらの定義には以下の記号が使われます。

N	全データ数 (バスケットの数)
I, J	アイテム集合
$k(I)$	全データ中にアイテム集合 I が現れる回数

ID	バスケット
1	{A, B, C, D}
2	{A, B, E}
3	{B, C, F, G}
\vdots	\dots
N	{B, C, E, H}

支持度, 確信度, リフト値

支持度 : 全データ中に I と J が同時に現れる割合

$$S(I \Rightarrow J) = \frac{k(I \cap J)}{N}$$

確信度 : I が出現した時に J も同時に現れる割合

$$C(I \Rightarrow J) = \frac{k(I \cap J)}{k(I)}$$

リフト値 : 支持度と全データの中でアイテム 集合 J が出現する割合の比

$$L(I \Rightarrow J) = \frac{C(I \Rightarrow J)}{k(J)/N} = \frac{k(I \cap J)N}{k(I)k(J)}$$

- 支持度は全体の中で I と J が同時に現れる割合で, 因果関係を意味しているわけではないが, シンプルで分かりやすい指標です.
- 確信度は I と J の関係以上に J の出現割合に依存する指標なので注意が必要です.
- リフト値が 1 より大きければアイテム集合 J の販売に I の関連性があると考えられます.

例：簡単な仮想データの場合

右図の販売データの場合,

$$N(\text{データ数}) = 5, k(B) = 4, k(C) = 3, k(B \cap C) = 3$$

なので, 相関ルール $B \Rightarrow C$ について

$$S(B \Rightarrow C) = \frac{k(B \cap C)}{N} = \frac{3}{5} = 0.6$$

$$C(B \Rightarrow C) = \frac{k(B \cap C)}{k(B)} = \frac{3}{4} = 0.75$$

$$L(B \Rightarrow C) = \frac{k(B \cap C)N}{k(B)k(C)} = \frac{3 \times 5}{4 \times 3} = 1.25$$

ID	アイテム集合
1	{A, B, C, D}
2	{A, B, E}
3	{B, C, F, G}
4	{D, E, F, G, H}
5	{B, C, E, H}

となり, 支持度, 確信度, リフト値はそれぞれ, 0.6, 0.75, 1.25 となります.

Aprioriアルゴリズム

販売データの中に頻繁に表れる頻出パターンや相関ルールを求めるためには、すべてのアイテム集合がデータの中に何回現れるかを調べることになります。これは一見すると簡単そうに見えますが、ビッグデータの場合には組み合わせ爆発によって計算が困難になります。Apriori アルゴリズムはアソシエーション分析において高速に相関ルールや頻出アイテム集合を検出するために1990年代前半にIBM研究所で開発されたアルゴリズムで、その後開発されたアルゴリズムの基礎ともなっています。

Aprioriアルゴリズムでは、支持度(support)と確信度(confidence) の閾値を指定するとそれ以上の値の相関ルールを列挙します。支持度を小さく設定しすぎると大量のルールを列挙するので注意が必要です。

Aprioriアルゴリズムの適用例（１）

Rのライブラリーに標準データとして付属している Groceries（食料品データ $N=9835$ ）に対してAprioriアルゴリズムを適用し、リフト値が大きな順に並べた結果を示します。ただし、support=0.005, confidence=0.6と設定した場合の結果です。

ID	lhs（条件） \Rightarrow rhs（結論）	支持度	確信度	リフト値
1	{柑橘類, 根菜, 牛乳} \Rightarrow {他の野菜}	0.00579	0.633	3.273
2	{果物, 根菜, 牛乳} \Rightarrow {他の野菜}	0.00549	0.614	3.171
3	{果物, クリーム} \Rightarrow {他の野菜}	0.00559	0.604	3.124
4	{根菜, 玉葱} \Rightarrow {他の野菜}	0.00569	0.602	3.112
5	{南国果物, 根菜, ヨーグルト} \Rightarrow {牛乳}	0.00569	0.700	2.740
6	{バター, クリーム} \Rightarrow {牛乳}	0.00549	0.675	2.642

この結果は、食料品だけの間の相関分析で目新しいものではありませんが、アメリカにおける相関分析の有名な例として紙オムツとビールの関連性の発見がよく知られています。このように人間では気づきにくい関連性の発見が、ビッグデータ分析の醍醐味で、それらを並べて陳列することによって販売促進に役に立つ可能性があります。

Aprioriアルゴリズムの適用例（2）

同じデータでも支持度と確信度の閾値を変えると全く違った結果が得られることがあります。
下の結果はsupport=0.001, confidence=0.5と設定した場合です。

ID	lhs（条件）	⇒ rhs（結論）	支持度	確信度	リフト値
1	{インスタント食品, ソーダ}	⇒ {ハンバーグ肉}	0.00122	0.632	18.996
2	{ソーダ, ポップコーン}	⇒ {塩味スナック}	0.00122	0.632	16.698
3	{小麦粉, ベーキングパウダー}	⇒ {砂糖}	0.00102	0.556	16.408
4	{ハム, プロセスチーズ}	⇒ {白パン}	0.00193	0.633	15.045
5	{牛乳, インスタント食品}	⇒ {ハンバーグ肉}	0.00153	0.500	15.038
6	{野菜, 凝乳, ヨーグルト, クリーム}	⇒ {クリームチーズ}	0.00102	0.588	14.834

支持度の閾値を下げるとリフト値が15を超えるパターンが検出されるようになります。また前頁の結果と違って、野菜や乳製品以外のアイテムも出現します。ただし、支持度は0.1%とかなり小さな値になります。

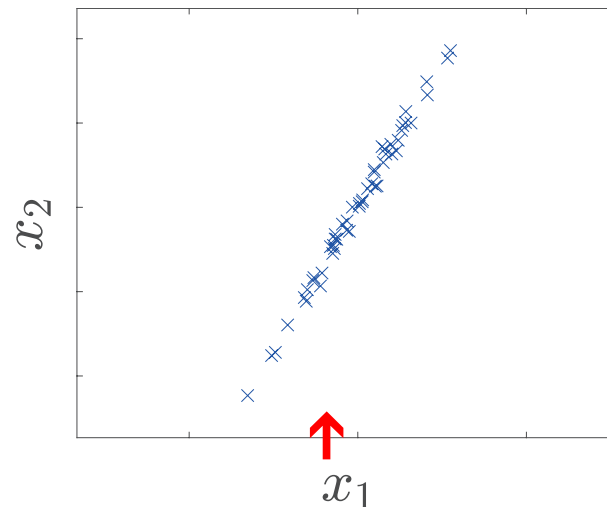
1-4-6 次元削減

動機：多次元のデータの取り扱い

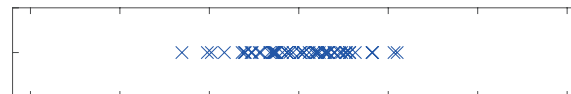
- 多次元のデータは、さまざまな場面で現れます。
 - たとえば、100 人の「身長、体重、体温、血圧、脈拍、酸素濃度、血糖値」のデータは、7 次元空間の中の 100 個の点の集合とみなせます。
- 高次元のデータは、低次元のデータに（近似的に）変換してから扱うと、さまざまな点で便利です。
 - データの可視化が、容易になります。
 - 少ない数の因子で、データを説明できるようになります。
 - データの容量が小さくなり、計算機で扱いやすくなります。
- このように、与えられたデータをより低次元のデータに近似的に変換することを、次元削減とよびます。
 - 元のデータがもつ情報を完全に保ったまま次元削減を行うことは、一般に、不可能です。したがって、元のデータがもつ情報をなるべく保持したまま次元を減らすことが、次元削減の目標です。

2次元のデータを1次元に圧縮する例 (1/4)

- 説明の都合上, まず, 2次元のデータを1次元に変換する状況を考えましょう.
- 次の図のようなデータ集合が与えられたとします:

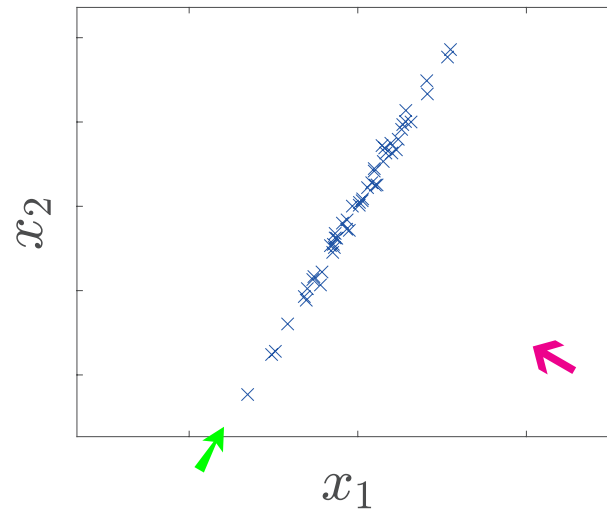


- ある特定の視点 (方向) からこのデータを見ると, 情報をなるべく失わないためにはどの視点 (方向) から見ればよいでしょうか.
- たとえば, **↑** の方向から見た場合は, 次のように見えます:

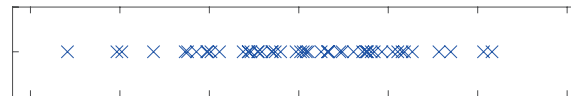


- これは, x_2 の情報を捨てて, x_1 のみの情報を取り出したことに相当します.

2次元のデータを1次元に圧縮する例 (2/4)



- データを  の方向から見ると、次のように見えます：



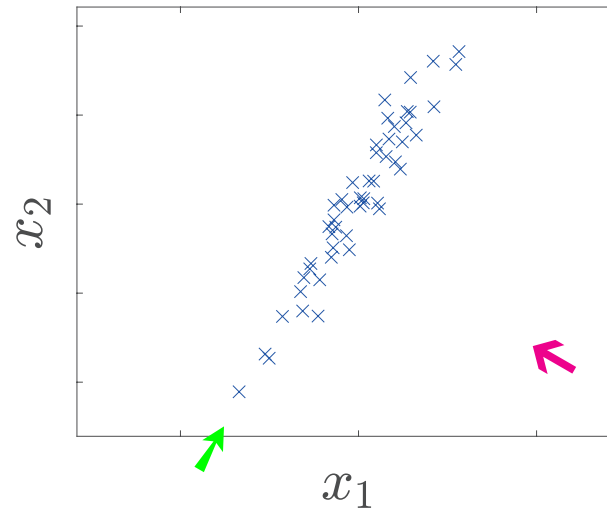
- 一方,  の方向から見ると、次のように見えます：



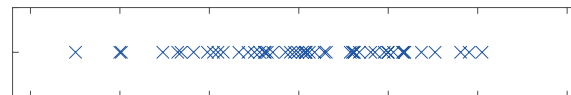
- 後者では, ほとんどの情報が失われています (データ点どうしの区別が, ほとんどつきません).
- 前者のように, ばらつき (つまり, 分散) が大きくなる方向が, 望ましい方向であることがわかります.

2次元のデータを1次元に圧縮する例 (3/4)

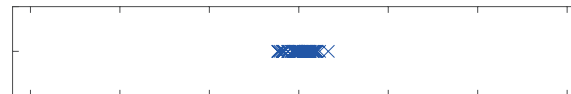
- 先ほどは極端な例でしたが，このデータ集合ではどうでしょうか．



- データを  の方向から見ると，次のように見えます：



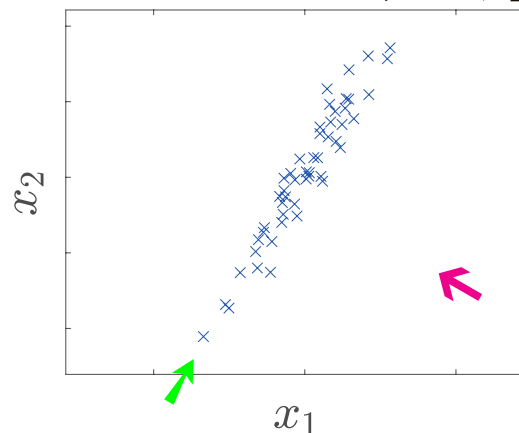
- 一方， の方向から見ると，次のように見えます：





- やはり，分散が大きくなる前者の方向が，望ましい方向であると考えられます．

2次元のデータを1次元に圧縮する例 (4/4)

- 「ある方向からデータを見る」ことは、「データ点の座標の重み和をとる (ただし, 重みベクトルのノルムは1とする)」ことに相当します.



- x_2 軸から $\pi/3$ 回転した方向  から見ることは, $\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \cos(\pi/3) \\ \sin(\pi/3) \end{bmatrix}$ を計算することに相当します.
- x_2 軸から $-\pi/6$ 回転した方向  から見ることは, $\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \cos(-\pi/6) \\ \sin(-\pi/6) \end{bmatrix}$ を計算することに相当します.
- 一般に, $\|w\| = 1$ を満たす $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ を用いて $\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$ と表せます.

主成分分析：分散最大化としての導入

- n 次元のデータを 1 次元に圧縮することを考えましょう.

- m 個のデータ点

$$\mathbf{x}_i = [x_{i1} \ x_{i2} \ \cdots \ x_{in}] \quad (i = 1, \dots, m)$$

が与えられているとします.

- 以降の表記を簡潔にするため、データ点は行ベクトルで表します.
- 1 次元への圧縮は、 $\|\mathbf{w}\| = 1$ を満たす列ベクトル $\mathbf{w} \in \mathbb{R}^n$ を用いて $\mathbf{x}_1\mathbf{w}$, $\mathbf{x}_2\mathbf{w}, \dots, \mathbf{x}_m\mathbf{w}$ と表せます.
- 以下では、データの情報をできるだけ失わないため、 $\mathbf{x}_1\mathbf{w}, \mathbf{x}_2\mathbf{w}, \dots, \mathbf{x}_m\mathbf{w}$ の分散が最大になるように \mathbf{w} を選ぶことを考えます.
- このような次元削減の手法を、主成分分析とよびます.
- 主成分分析の解説の前に、予備知識として、行列の固有値分解と特異値分解についてまとめておきましょう.

予備知識：対称行列の固有値分解 (1/2)

- n 次の対称行列 A に対して, n 次の直交行列 Q をうまく選ぶことで

$$A = Q \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}}_{\text{対角行列}} Q^{\top}$$

と表すことができます. これを, A の固有値分解とよびます.

- ただし, n 次の正方行列 Q が $Q^{\top}Q = I$ (つまり, $Q^{-1} = Q$) を満たすとき, Q を直交行列とよびます.
- また, $\lambda_1, \lambda_2, \dots, \lambda_n$ は A の固有値です (\Rightarrow 1-6-5 線形代数).

予備知識：対称行列の固有値分解 (2/2)

- A の固有値分解

$$A = Q \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} Q^{\top}$$

において, Q を

$$Q = \left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array} \right]$$

と表すと, Q の列ベクトル \mathbf{q}_j ($j = 1, 2, \dots, n$) は固有値 λ_j に対応する A の固有ベクトルです (\Rightarrow 1-6-5 線形代数).

- Q が直交行列であることから, \mathbf{q}_j ($j = 1, 2, \dots, n$) は, $\|\mathbf{q}_j\| = 1$, $\mathbf{q}_j^{\top} \mathbf{q}_l = 0$ ($j \neq l$) を満たします.
- このように, ノルムが 1 で, かつ, 互いに直交するベクトルの組を, 正規直交系とよびます.

予備知識：行列のランク

- ベクトル x_1, x_2, \dots, x_n が与えられたとき、定数 a_1, a_2, \dots, a_n に対して
$$a_1x_1 + a_2x_2 + \dots + a_nx_n = \mathbf{0}$$
が成り立つのが $a_1 = a_2 = \dots = a_n = 0$ である場合に限られるとき、 x_1, x_2, \dots, x_n は線形独立（または、1 次独立）であるといいます.
- $m \times n$ 型の実行列 A を考えます.
 - A の n 本の列ベクトルのうち線形独立なものの最大個数を、 A のランク（または、階数）とよびます.
 - A のランクは、また、 A の m 本の行ベクトルのうち線形独立なものの最大個数でもあります.

予備知識：特異値分解 (1/2)

- $m \times n$ 型の実行列 A のランクを r で表します. m 次の直交行列 U と n 次の直交行列 V をうまく選ぶことで, A を

$$A = U\Sigma V^{\top}$$

と表すことができます. ただし, Σ は $m \times n$ 型の実行列で,

$$\Sigma = \left[\begin{array}{cccc|ccc} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \end{array} \right], \quad \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$$

の形です. これを, A の特異値分解とよびます.

- ここで, $\sigma_1, \sigma_2, \dots, \sigma_r$ を A の特異値とよびます.

予備知識：特異値分解 (2/2)

- A の特異値分解を考えます： $A = U\Sigma V^\top$.
- A の特異値は、 $A^\top A$ の（ゼロでない）固有値の平方根に等しいです.
 - $V = [\boldsymbol{v}_1 \mid \boldsymbol{v}_2 \mid \cdots \mid \boldsymbol{v}_n]$ と表すと、 V の列ベクトル \boldsymbol{v}_j は $A^\top A$ の固有ベクトルです（この \boldsymbol{v}_j を、 A の右特異ベクトルとよびます）.
- A の特異値は、 AA^\top の（ゼロでない）固有値の平方根に等しいです.
 - $U = [\boldsymbol{u}_1 \mid \boldsymbol{u}_2 \mid \cdots \mid \boldsymbol{u}_m]$ と表すと、 U の列ベクトル \boldsymbol{u}_i は AA^\top の固有ベクトルです（この \boldsymbol{u}_i を、 A の左特異ベクトルとよびます）.
- 行列 A のノルム（フロベニウスノルム）は、
$$\|A\|_F = \sqrt{\text{tr}(A^\top A)} = \sqrt{\text{tr}(AA^\top)}$$
で定義されます（正方行列 B に対して、対角成分の和をトレースとよび、 $\text{tr } B$ で表します）.
 - A の特異値を用いると、
$$\|A\|_F = \sqrt{\sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2}$$
と表せます.

主成分分析：1次元への圧縮 (1/2)

- データ行列 $X \in \mathbb{R}^{m \times n}$ を次式で定義します：

$$X = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_m \end{bmatrix}.$$

- 各 $j = 1, 2, \dots, n$ に対して, $x_{1j}, x_{2j}, \dots, x_{mj}$ の平均を \bar{x}_j で表します：

$$\bar{x}_j = \frac{1}{m} \sum_{i=1}^m x_{ij}.$$

- 平均 \bar{x}_j を並べた行ベクトルを $\bar{\mathbf{x}}$ で表します：

$$\bar{\mathbf{x}} = [\bar{x}_1 \quad \bar{x}_2 \quad \cdots \quad \bar{x}_n] = \frac{1}{m} \mathbf{1}_m^\top X, \quad \mathbf{1}_m = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

- X の中心化 $X_0 \in \mathbb{R}^{m \times n}$ を次式で定義します：

$$X_0 = X - \mathbf{1}_m \bar{\mathbf{x}}.$$

X_0 は, データ行列の平均を 0 としたものです： $\frac{1}{m} \mathbf{1}_m^\top X_0 = \mathbf{0}_{1,n}$.

主成分分析：1次元への圧縮 (2/2)

- $x_1 w, \dots, x_m w$ の平均は,

$$\frac{1}{m} \sum_{i=1}^m x_i w = \left(\frac{1}{m} \sum_{i=1}^m x_i \right) w = \bar{x} w .$$

- $x_1 w, \dots, x_m w$ の分散は,

$$\frac{1}{m} \sum_{i=1}^m (x_i w - \bar{x} w)^2 = \frac{1}{m} \sum_{i=1}^m ((x_i - \bar{x}) w)^2 = w^\top \left(\frac{1}{m} X_0^\top X_0 \right) w .$$

- したがって, $S = \frac{1}{m} X_0^\top X_0$ とおくと, 求めたいものは, 条件 $\|w\| = 1$ を満たす w のうち $w^\top S w$ を最大にするものです.
 - S は, 標本分散共分散行列とよばれる n 次の対称行列です.
- そのような w は, S の最大固有値に対応する単位固有ベクトルであることがわかります (スライド80で説明します) .

主成分分析： p 次元への圧縮 (1/3)

- 以上では、 n 次元のデータの 1 次元への圧縮 $x_1 w, \dots, x_m w$ について、圧縮後の分散が最大になるように単位ベクトル w を選ぶことを考えました。
- 以下では、これを一般化して、データの p 次元 ($p < n$) への圧縮を考えます。これは、分散ができるだけ大きくなるように正規直交系 w_1, \dots, w_p を選ぶことに相当します。
 - 正規直交系は、スライド 73 で定義しました。

主成分分析： p 次元への圧縮 (2/3)

- 標本分散共分散行列 S は n 次の対称行列ですから、その固有ベクトル $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ を正規直交系としてとると

$$S = \left[\begin{array}{c|c|c|c} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_n \end{array} \right] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \left[\begin{array}{c} \hline \boldsymbol{v}_1^\top \\ \boldsymbol{v}_2^\top \\ \hline \vdots \\ \hline \boldsymbol{v}_n^\top \end{array} \right]$$

と表せます。ただし、実数 $\lambda_1 \geq \cdots \geq \lambda_n$ は S の固有値です。

- $\boldsymbol{x}_1 \boldsymbol{w}, \dots, \boldsymbol{x}_m \boldsymbol{w}$ の分散は $\boldsymbol{w}^\top S \boldsymbol{w}$ でした。
- $\boldsymbol{w} = c_1 \boldsymbol{v}_1 + \cdots + c_n \boldsymbol{v}_n$ とおけば、 $\boldsymbol{w}^\top S \boldsymbol{w} = \lambda_1 c_1^2 + \cdots + \lambda_n c_n^2$ です。また、条件 $\|\boldsymbol{w}\| = 1$ は $c_1^2 + \cdots + c_n^2 = 1$ と書けます。
- つまり、分散 $\boldsymbol{w}^\top S \boldsymbol{w}$ は、 $\lambda_1, \dots, \lambda_n$ の非負の重み付き和（重みの和をとると 1）とみなせますから、 $\boldsymbol{w} = \boldsymbol{v}_1$ で最大値をとることがわかります。
- 次に、 \boldsymbol{v}_1 に直交するベクトルは $\boldsymbol{w} = c_2 \boldsymbol{v}_2 + \cdots + c_n \boldsymbol{v}_n$ と表せます。このとき、 $\boldsymbol{w}^\top S \boldsymbol{w} = \lambda_2 c_2^2 + \cdots + \lambda_n c_n^2$ は $\boldsymbol{w} = \boldsymbol{v}_2$ において最大値をとります。
- つまり、 p 次元への圧縮には $\boldsymbol{v}_1, \dots, \boldsymbol{v}_p$ を用いればよいことがわかります。

主成分分析： p 次元への圧縮 (3/3)

- このようにして，データ行列 X の 1 次元への圧縮は

$$Xv_1$$

で与えられ，2 次元への圧縮は

$$X \left[\begin{array}{c|c} v_1 & v_2 \end{array} \right]$$

で与えられます．一般に， p 次元への圧縮は

$$X \left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_p \end{array} \right]$$

で与えられます．

- Xv_1 の第 i 行 $x_i v_1$ を x_i の第 1 主成分とよび， Xv_2 の第 i 行 $x_i v_2$ を x_i の第 2 主成分とよびます．一般に， $x_i v_p$ を x_i の第 p 主成分とよびます．

低ランク行列近似としての主成分分析 (1/3)

- 主成分分析で用いるベクトル $\mathbf{v}_1, \dots, \mathbf{v}_n$ は, 行列 $S = \frac{1}{m} X_0^\top X_0$ の正規直交な固有ベクトルでした.
- したがって, $\mathbf{v}_1, \dots, \mathbf{v}_n$ は中心化されたデータ行列 $X_0 \in \mathbb{R}^{m \times n}$ の右特異ベクトルでもあります. 特異値分解により X_0 は

$$X_0 = \left[\begin{array}{c|c|c|c} \sigma_1 \mathbf{u}_1 & & & \\ & \cdots & & \\ & & \sigma_r \mathbf{u}_r & \\ & & & O_{m, n-r} \end{array} \right] \left[\begin{array}{c} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_n^\top \end{array} \right]$$

と表せます. ただし, r は X_0 のランクであり, $\sigma_1 \geq \dots \geq \sigma_r > 0$ は X_0 の特異値であり, $\mathbf{u}_1, \dots, \mathbf{u}_r \in \mathbb{R}^m$ はこれに対応する (正規直交な) 左特異ベクトルです.

- ここで $X_0 \mathbf{v}_j = \sigma_j \mathbf{u}_j$ が成り立ちますから, $\sigma_j \mathbf{u}_j$ ($j = 1, \dots, r$) は中心化されたデータ $x_1 - \bar{x}, \dots, x_m - \bar{x}$ の第 j 主成分を縦に並べたベクトルであることがわかります.

低ランク行列近似としての主成分分析 (2/3)

- スライド82の結果から, $X_0 \in \mathbb{R}^{m \times n}$ は

$$X_0 = \left[\begin{array}{c|c|c} & & \\ \hline \sigma_1 \mathbf{u}_1 & \cdots & \sigma_p \mathbf{u}_p \\ \hline & & \end{array} \right] \left[\begin{array}{c} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_p^\top \end{array} \right] + \underbrace{\left[\begin{array}{c|c|c} & & \\ \hline \sigma_{p+1} \mathbf{u}_{p+1} & \cdots & \sigma_r \mathbf{u}_r \\ \hline & & \end{array} \right] \left[\begin{array}{c} \mathbf{v}_{p+1}^\top \\ \vdots \\ \mathbf{v}_r^\top \end{array} \right]}_{= E \text{ とおきます.}}$$

と表せます. ここで, もし行列 E のノルムが小さければ, n 次元のデータ X_0 はランクが p の行列で近似できることになります:

$m \times n$ 型で, ランクは $p (< n)$

$$X_0 \simeq \underbrace{\left[\begin{array}{c|c|c} & & \\ \hline \sigma_1 \mathbf{u}_1 & \cdots & \sigma_p \mathbf{u}_p \\ \hline & & \end{array} \right]}_{X_0 \text{ の } p \text{ 次元への圧縮}} \left[\begin{array}{c} \mathbf{v}_1^\top \\ \vdots \\ \mathbf{v}_p^\top \end{array} \right].$$

X_0 の p 次元への圧縮

これを, X_0 の低ランク行列近似とよびます.

低ランク行列近似としての主成分分析 (3/3)

- スライド83で, 行列

$$E = \left[\begin{array}{c|c|c} \sigma_{p+1} \mathbf{u}_{p+1} & \cdots & \sigma_r \mathbf{u}_r \end{array} \right] \left[\begin{array}{c} \hline \mathbf{v}_{p+1}^\top \\ \vdots \\ \hline \mathbf{v}_r^\top \end{array} \right]$$

のノルムが小さいほど, 低ランク行列近似の精度はよいと考えられます.

E のフロベニウスノルム $\|E\|_F$ の 2 乗は

$$\|E\|_F^2 = \text{tr}(E^\top E) = \sigma_{p+1}^2 + \cdots + \sigma_r^2$$

です.

- 同様に, $\|X_0\|_F^2 = \sigma_1^2 + \cdots + \sigma_r^2$ が成り立ちます.

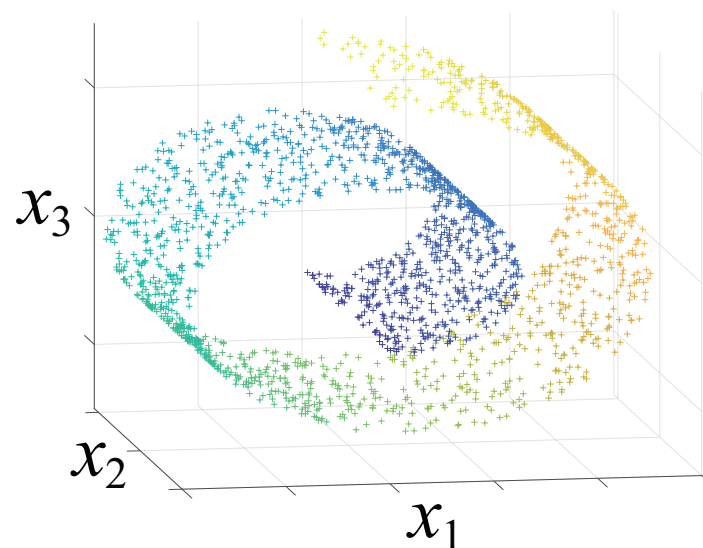
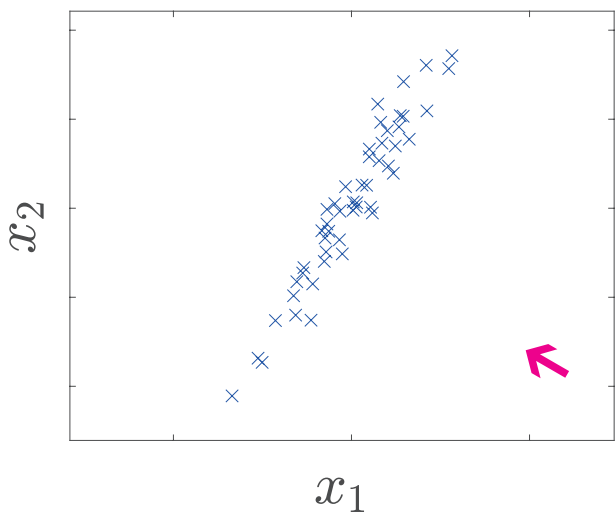
- したがって, 近似の精度の尺度として

$$\frac{\|X_0\|_F^2 - \|E\|_F^2}{\|X_0\|_F^2} = \frac{\sigma_1^2 + \cdots + \sigma_p^2}{\sigma_1^2 + \cdots + \sigma_p^2 + \cdots + \sigma_r^2}$$

が得られますが, この値を累積寄与率とよびます.

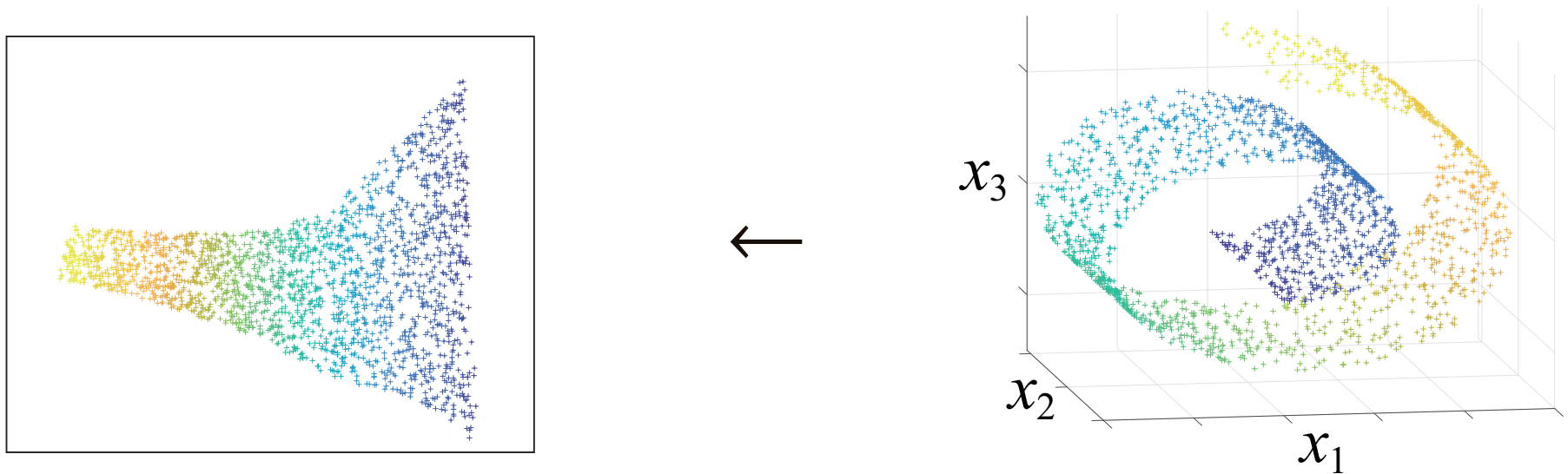
非線形の次元削減：導入 (1/2)

- 主成分分析は、「データを見る方向」を上手に選ぶことで、分散をなるべく保ったまま次元を削減する手法でした。「見る方向」を定めることは、主成分分析が本質的に線形変換を行っていることを意味します。
- このため、主成分分析は、たとえば左図のデータならうまく扱うことができます。
- 右図のデータ（スイスロールとよばれます）は、どうでしょうか。
 - これは3次元のデータですが、データ点は（2次元的な）曲面上に載っているように見えます。



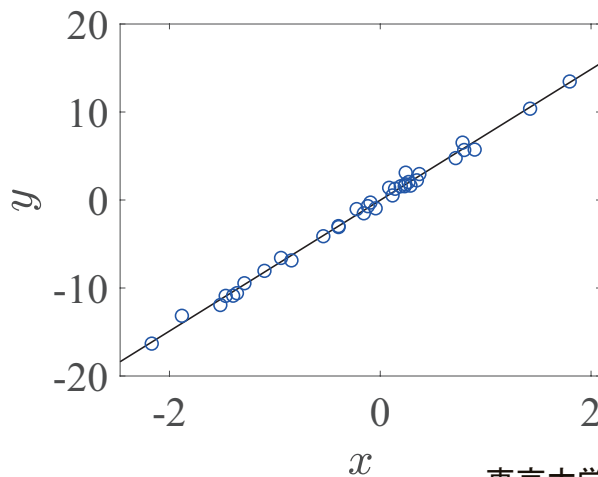
非線形の次元削減：導入 (2/2)

- 右図の「スイスロール」の曲面を（ある方法で）2次元平面上に広げると、左図のようになります。これで、次元が1つ削減できました。
- 曲面を平面に広げるのは非線形な変換ですので、主成分分析では扱うことができません。
- 以下では、非線形な次元削減の手法として、カーネル主成分分析を取りあげます。
- その解説の前に、予備知識として、カーネル法の概要をみておきましょう。



カーネル法：回帰分析への適用例 (1/3)

- 線形回帰（左下の図）では，目的変数 $y \in \mathbb{R}$ を説明変数 $x \in \mathbb{R}^n$ の 1 次関数として近似します.
- 与えられたデータ点を $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ とします.
 - x_1, x_2, \dots, x_m は，これまでと同様に，行ベクトルとします.
- 回帰式を $y = x\alpha + \beta$ とします ($\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n)^\top$ です).
- 最小 2 乗法では，回帰式の予測とデータ y_i の差の 2 乗和
$$\sum_{i=1}^m [(x\alpha + \beta) - y_i]^2 = \|X\alpha + \beta\mathbf{1}_m - y\|^2$$
が最小になるように $\alpha \in \mathbb{R}^n$ と $\beta \in \mathbb{R}$ を選びます.

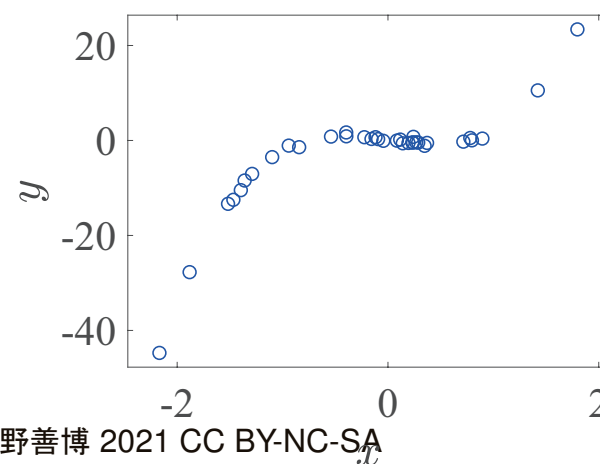
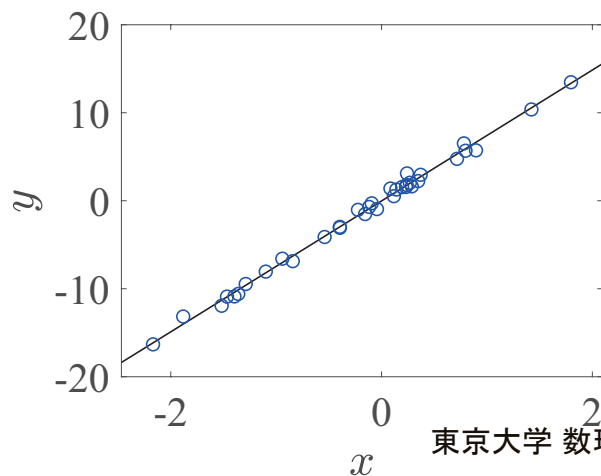


- X と y の定義は次のとおりです：

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}.$$

カーネル法：回帰分析への適用例 (2/3)

- 右下の図のデータは，1 次関数では当てはまりがとても悪そうです．このような場合に用いられる方法の 1 つがカーネル法です．
- カーネル法では，カーネル関数とよばれる関数を用います．
 - 代表例に，ガウスカーネルがあります ($\gamma > 0$ はパラメータです)：
$$k(x, s) = \exp(-\gamma \|x - s\|^2) .$$
- 回帰式は $y = \sum_{l=1}^m \tau_l k(x_l, x)$ とします．そして，回帰式の予測とデータの差の 2 乗和 $\sum_{i=1}^m \left[\sum_{l=1}^m \tau_l k(x_l, x_i) - y_i \right]^2$ が最小になるように $\tau_1, \tau_2, \dots, \tau_m \in \mathbb{R}$ を選びます．



カーネル法：回帰分析への適用例 (3/3)

- 関数 $f(\tau) = \sum_{i=1}^m \left[\sum_{l=1}^m \tau_l k(\mathbf{x}_l, \mathbf{x}_i) - y_i \right]^2$ を最小にする $\tau = (\tau_1, \tau_2, \dots, \tau_m)^\top$ は、次のようにして求められます.
- まず、 m 次の対称行列 K を
$$K = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \cdots & k(\mathbf{x}_1, \mathbf{x}_m) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \cdots & k(\mathbf{x}_2, \mathbf{x}_m) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_m, \mathbf{x}_1) & k(\mathbf{x}_m, \mathbf{x}_2) & \cdots & k(\mathbf{x}_m, \mathbf{x}_m) \end{bmatrix}$$
で定めます. この K を, グラム行列とよびます.
 - カーネル関数 k は, グラム行列が常に正則であるように定めます (たとえばガウスカーネルはこの条件を満たしています).
 - この K を用いると, $f(\tau) = \|K\tau - \mathbf{y}\|^2 (\geq 0)$ と書けます.
 - したがって, $f(\tau)$ は $\tau = K^{-1}\mathbf{y}$ において最小値 0 をとることがわかります.
 - 実際には過学習を防ぐために何らかの正則化を施しますが, 以上がカーネル法を回帰分析へ適用する際の概要です.

カーネル主成分分析 (1/2)

- それでは、カーネル主成分分析を説明します。主成分分析では、中心化されたデータ行列 X_0 の特異値分解を考えました（スライド82）：

$$X_0 = U\Sigma V = \left[\begin{array}{c|c|c|c} \sigma_1 \mathbf{u}_1 & \cdots & \sigma_r \mathbf{u}_r & O_{m,n-r} \end{array} \right] \left[\begin{array}{c} \hline \mathbf{v}_1^\top \\ \vdots \\ \hline \mathbf{v}_n^\top \end{array} \right]. \quad (*1)$$

ここで、 $\sigma_j \mathbf{u}_j$ ($j = 1, \dots, r$) は $\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_m - \bar{\mathbf{x}}$ の第 j 主成分を縦に並べたベクトルでした。

- また、 X_0 の定義（スライド77）は

$$X_0 = X - \mathbf{1}_m \bar{\mathbf{x}}, \quad \bar{\mathbf{x}} = \frac{1}{m} \mathbf{1}_m^\top X$$

でしたから、 $H_m = I_m - \frac{1}{m} \mathbf{1}_m \mathbf{1}_m^\top$ とおくと $X_0 = H_m X$ と書けます。

- 次に、式 (*1) より $X_0 X_0^\top = U \Sigma^2 U^\top$ が成り立ちますから、行列 $X_0 X_0^\top = H_m (X X^\top) H_m^\top$ の（ゼロでない）固有値は $\sigma_1^2, \dots, \sigma_r^2$ であり、対応する固有ベクトルは $\mathbf{u}_1, \dots, \mathbf{u}_r$ です。

カーネル主成分分析 (2/2)

- ここで行列 $H_m(XX^\top)H_m^\top$ を書き下すと

$$H_m(XX^\top)H_m^\top = H_m \begin{bmatrix} \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \langle \mathbf{x}_1, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_1, \mathbf{x}_m \rangle \\ \langle \mathbf{x}_2, \mathbf{x}_1 \rangle & \langle \mathbf{x}_2, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_2, \mathbf{x}_m \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle \mathbf{x}_m, \mathbf{x}_1 \rangle & \langle \mathbf{x}_m, \mathbf{x}_2 \rangle & \cdots & \langle \mathbf{x}_m, \mathbf{x}_m \rangle \end{bmatrix} H_m^\top$$

です.

- カーネル主成分分析では, データ点どうしの内積 $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ をカーネル関数の値 $k(\mathbf{x}_i, \mathbf{x}_j)$ で置き換えます. すると, $H_m(XX^\top)H_m^\top$ は $H_mKH_m^\top$ に置き換わります (K は, グラム行列です).
- したがって, $H_mKH_m^\top$ の (ゼロでない) 固有値を $\tilde{\lambda}_1, \dots, \tilde{\lambda}_r$ で表し, 対応する固有ベクトルを $\tilde{\mathbf{u}}_1, \dots, \tilde{\mathbf{u}}_r$ で表すと, カーネル主成分分析による X の p 次元への圧縮は

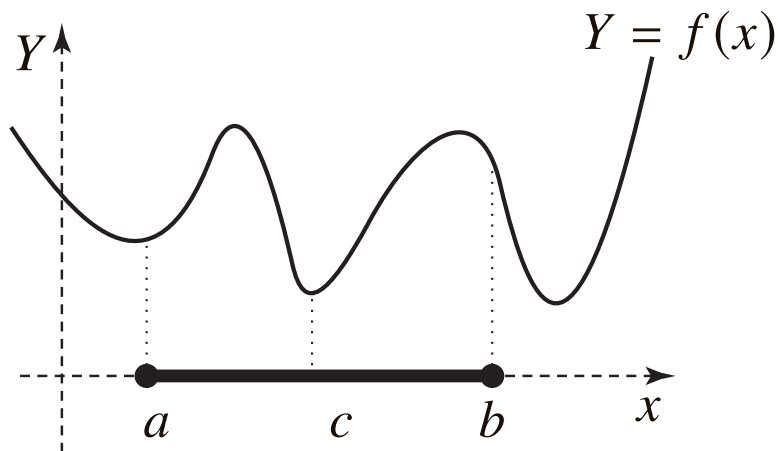
$$\left[\begin{array}{c|c|c|c} \sqrt{\tilde{\lambda}_1} \tilde{\mathbf{u}}_1 & \sqrt{\tilde{\lambda}_2} \tilde{\mathbf{u}}_2 & \cdots & \sqrt{\tilde{\lambda}_p} \tilde{\mathbf{u}}_p \end{array} \right]$$

であることがわかります.

1-4-7 最適化概論

最適化とは：最適化問題の定義

- 最適化問題とは、与えられた条件の下である関数の値を最小にする解（または最大にする解）を求める問題のこと（その解を最適解とよびます）。
- 最適化とは、何らかの意思決定を支援するために、最適化問題を定式化し、それを解く方法論のこと。
- 最適化問題の構成要素
 - （決定）変数：意思決定において値を決めるべき量
 - 目的関数：最小（または最大）にしたい評価尺度となる関数
 - 制約条件：決定変数が満たすべき条件

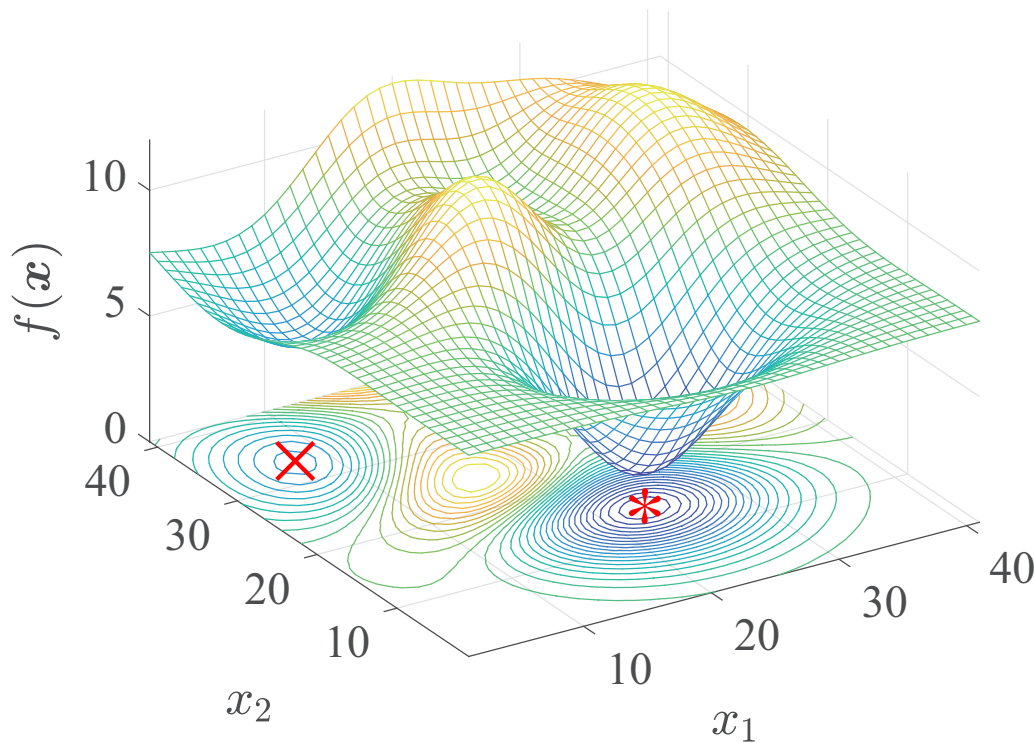


1 変数の最適化問題の例

- 左の最小化問題の例では、決定変数は $x \in \mathbb{R}$ ，目的関数は $f : \mathbb{R} \rightarrow \mathbb{R}$ ，制約条件は $a \leq x \leq b$ ，最適解は $x = c$ です。
- 最適解における目的関数の値を、最適値とよびます。

最適化とは：局所最適解と大域的最適解

- $(f(x)$ を最大化する問題の最適解) = $(-f(x)$ を最小化する問題の最適解).
- したがって、最小化問題に限って議論してもよいことがわかります.
- 制約条件をもたない最適化問題は、無制約最適化問題とよべれます.



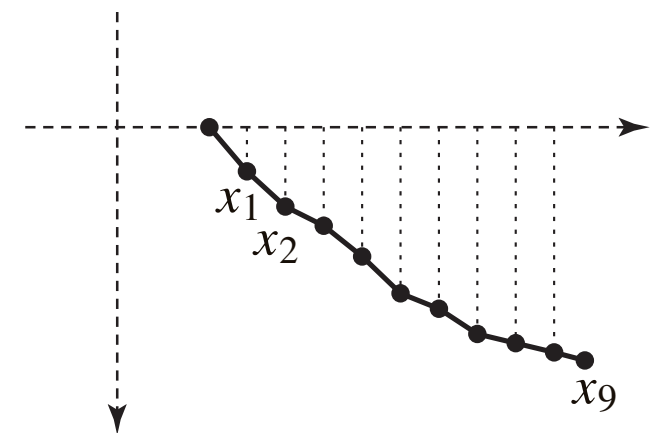
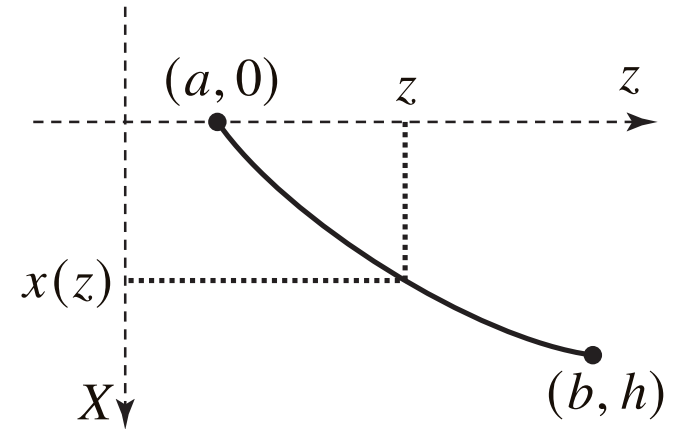
2変数の無制約最適化問題の例

- 左の最小化問題の例で,
 - 点 **X** のように、その点の近傍には目的関数値がより小さい解が存在しない点を、局所最適解とよびます.
 - 最適解 ***** のことを他の局所最適解と特に区別したい場合は、大域的最適解とよびます.

最適化とは：変分問題と最適化問題

- 最速降下線問題：

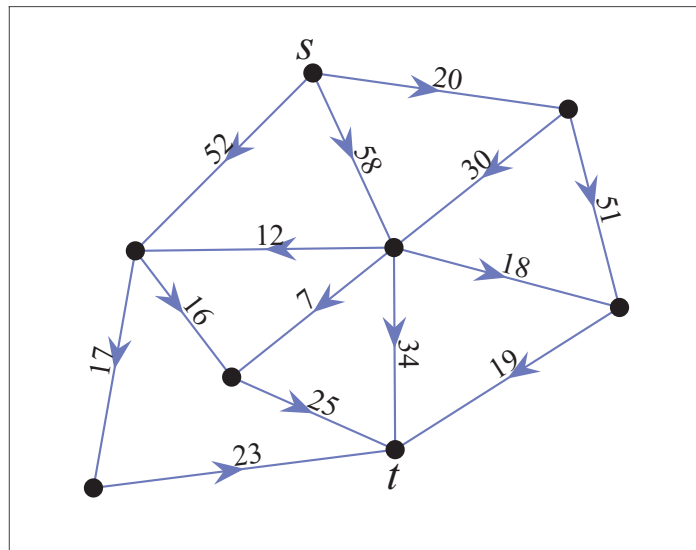
- 鉛直面内に曲線 $X = x(z)$ があり，端点 $(a, 0)$, (b, h) の位置は指定されています．質点を，重力に従って曲線上を降下させます．
- 質点が降下に要する時間が最小の曲線は？
- これは，変分問題とよばれる問題の例です．
- 下図のように，折れ線で考えてみましょう．
 - 各折れ点の高さを決めればよいので，決定変数の数は折れ点の数（図では9個）です．
 - 逆に言うと，上図の曲線の問題は，無限個の点の高さを決める最適化問題とみなせます．



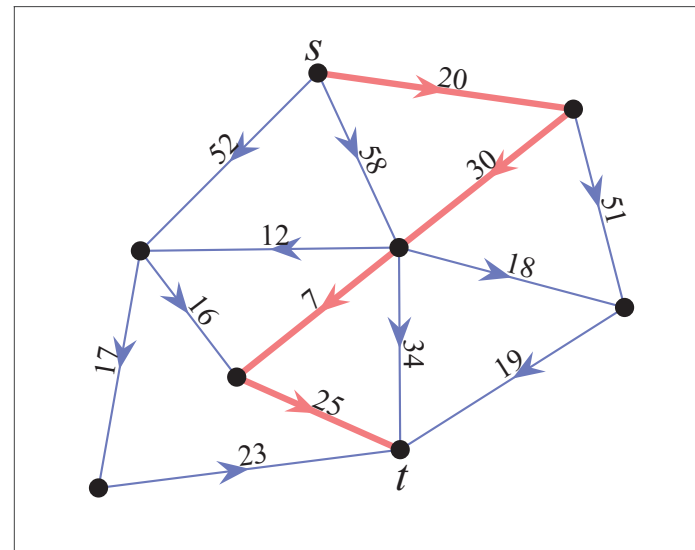
- このように，変分問題は，無限個の決定変数をもつ最適化問題とみなせます．理工学におけるさまざまな支配式が，変分問題として記述できます．

離散最適化問題：最短路問題を例として

- 左図において、点 s から点 t に移動する経路のうち、経路上の数字（重みとよびます）の和が最小のものを求める問題を考えます。



辺が 14 本ある例題



最適解

- この最適化問題は、2つの点を結ぶ矢印（辺とよびます）それぞれを経路として採用するかしないかを選択する問題とみることができます。この選択は離散的ですので、このような問題を離散最適化問題とよびます。
- 点 s から点 t に移動する経路は、有限個しかありません。このように、制約条件を満たす解の個数が有限個の問題を組合せ最適化問題とよびます。
- 離散最適化問題と組合せ最適化問題は、厳密に区別しないことが多いです。

連続最適化問題：線形計画問題を例として

- 制約条件

$$5x_1 + 4x_2 \leq 80, \quad 2x_1 + 4x_2 \leq 40, \quad 2x_1 + 8x_2 \leq 64$$

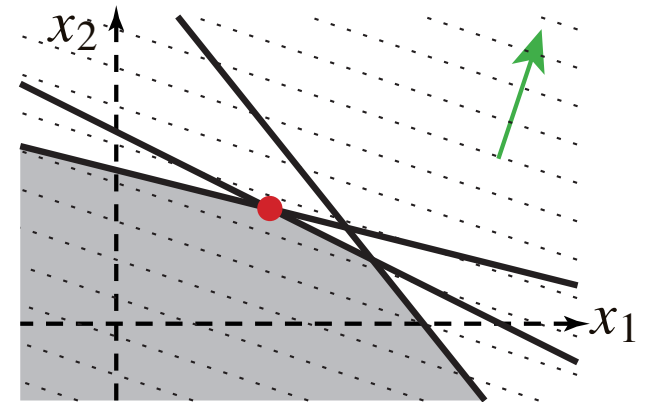
の下で，目的関数

$$20x_1 + 60x_2$$

を最大化する最適化問題を考えます．

- この問題は，数式で次のように記述します：

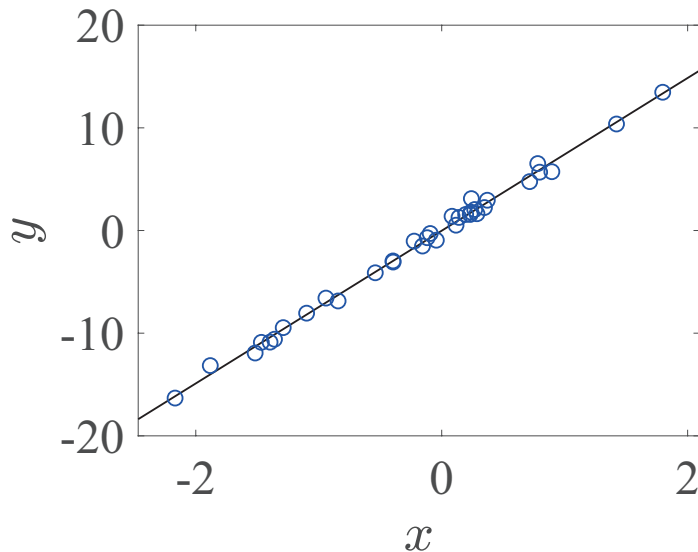
Maximize	$20x_1 + 60x_2$
subject to	$5x_1 + 4x_2 \leq 80,$
	$2x_1 + 4x_2 \leq 40,$
	$2x_1 + 8x_2 \leq 64.$



- 決定変数 x_1 および x_2 は連続的に変化する量ですので，このような問題を連続最適化問題とよびます．
- 特に，この問題は，目的関数も制約条件もすべて 1 次式で表されています．このような問題を線形計画問題とよびます．

最適化の具体例：回帰分析 (1/3)

- 回帰分析の1つである線形回帰（左下の図）では，目的変数 $y \in \mathbb{R}$ を説明変数 $x \in \mathbb{R}$ の1次関数として近似します.
- 与えられたデータ点を $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ とします.
- 回帰式を $y = \beta_0 + \beta_1 x$ とします.
- 最小2乗法では，回帰式による予測とデータの差の2乗和
$$\sum_{i=1}^m [(\beta_0 + \beta_1 x_i) - y_i]^2$$
 を最小にする β_0 と β_1 の値を求めます.



- X, β, y を

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

で定義すると，目的関数は $\|X\beta - y\|^2$ と書けます（ β が，最適化の決定変数です）.

最適化の具体例：回帰分析 (2/3)

- 重回帰分析では、説明変数の数を p とすると、各 $i = 1, 2, \dots, m$ に対して、データ点は $(x_{1i}, x_{2i}, \dots, x_{pi})$ と y_i の組として与えられます。
- 回帰式を $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$ とします。
- 最小 2 乗法では、回帰式による予測とデータとの差の 2 乗和

$$\sum_{i=1}^m [(\beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \dots + \beta_p x_{pi}) - y_i]^2$$

を最小にする $\beta_0, \beta_1, \beta_2, \dots, \beta_p$ の値を求めます。

- X, β, y を

$$X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{p1} \\ 1 & x_{12} & x_{22} & \cdots & x_{p2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{1m} & x_{2m} & \cdots & x_{pm} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

で定義すると、目的関数は $\|X\beta - y\|^2$ と書けます。さらにこれを展開すると、2 次関数 $\beta^\top (X^\top X) \beta - 2(X^\top y)^\top \beta$ を最小化する β を求めればよいことがわかります（定数項 $\|y\|^2$ は最適化に影響しないことに注意します）。

最適化の具体例：回帰分析 (3/3)

- データへの過学習（過度なあてはまり）を防ぐための種々の方法が知られています.
- ティコノフ正則化付き最小 2 乗法（リッジ回帰）とよばれる方法では，誤算の 2 乗和に $\gamma\|\beta\|^2$ を加えます（ $\gamma > 0$ はパラメータです）. 目的関数を整理すると

$$\beta^\top (X^\top X + \gamma I) \beta - 2(X^\top y)^\top \beta$$

となりますので，やはり 2 次関数の最小化問題であることがわかります.

- 一方， ℓ_1 ノルム正則化付き最小 2 乗法とよばれる方法では， $\gamma(|\beta_0| + |\beta_1| + |\beta_2| + \cdots + |\beta_p|)$ を加えます. この問題は，補助的な変数として s_j ($j = 0, 1, 2, \dots, p$) を導入すると，1 次不等式で表される制約条件

$$s_j \geq \beta_j, \quad s_j \geq -\beta_j \quad (j = 0, 1, 2, \dots, p)$$

の下で 2 次関数

$$\beta^\top (X^\top X) \beta - 2(X^\top y)^\top \beta + \sum_{j=0}^p s_j$$

を最小化する問題に帰着できます（最適解では各 $j = 0, 1, 2, \dots, p$ に対して $s_j = |\beta_j|$ が成り立つことに注意します）.

ナップサック問題とは

- 品物が n 個あり，そのうちのいくつかを選んで1つのナップサックに詰め込むことを考えます．
- それぞれの品物には，重量と価格が与えられています．ナップサックには重量制限が与えられており，詰め込んだ品物の重量の和がその制限を超えてはいけません．
- このとき，ナップサックに詰め込んだ品物の価格の和が最大になるように品物を選ぶ問題を，ナップサック問題とよびます．
- ナップサック問題は，それぞれの品物をナップサックに入れるか入れないかを選択する問題とみることができます．この選択は離散的ですので，ナップサック問題は離散最適化問題です．

整数計画問題としての定式化

- 品物の数が $n = 5$ 個の場合の具体例を考えます：

	品物 1	品物 2	品物 3	品物 4	品物 5
重量	3	4	1	2	5
価格	33	32	10	28	45

ナップサックの重量制限：7

- 品物 j ($j = 1, 2, \dots, 5$) に対して，決定変数 x_j を

$$x_j = \begin{cases} 1 & (\text{品物 } j \text{ をナップサックに入れるとき}), \\ 0 & (\text{品物 } j \text{ をナップサックに入れないとき}) \end{cases}$$

と対応させます．すると，ナップサック問題は次のように定式化できます：

$$\begin{aligned} &\text{Maximize} && 33x_1 + 32x_2 + 10x_3 + 28x_4 + 45x_5 \\ &\text{subject to} && 3x_1 + 4x_2 + x_3 + 2x_4 + 5x_5 \leq 7, \\ &&& x_j = 0 \text{ または } 1, \quad j = 1, 2, \dots, 5. \end{aligned}$$

- 決定変数が整数値のみをとるため，このような最適化問題は整数計画問題とよばれます．

貪欲算法

- 重量が軽くて価格が高い品物と、重量が重くて価格が安い品物とが、あったとします。ナップサック問題では、前者を詰め込む方が、後者を詰め込むよりも、得策のように思えます。
- そこで、単位重量あたりの価格が大きい順に、できるだけ多くの品物をナップサックに詰め込むことにすると、

$$(x_1, x_2, x_3, x_4, x_5) = (1, 0, 1, 1, 0)$$

が得られます。

	品物 1	品物 2	品物 3	品物 4	品物 5
重量	3	4	1	2	5
価格	33	32	10	28	45
価格/重量	11	8	10	14	9

ナップサックの重量制限：7

- このような解法を、貪欲算法とよびます。
- 貪欲算法では、必ずしも最適解は得られません。実際、この問題では

$$(x_1, x_2, x_3, x_4, x_5) = (0, 0, 0, 1, 1)$$

が最適解です。

緩和問題

- ナップサック問題の「 x_j は 0 または 1 をとる」という制約条件を，不等式「 $0 \leq x_j \leq 1$ 」に置き換えてみます：

$$\begin{array}{ll} \text{Maximize} & 33x_1 + 32x_2 + 10x_3 + 28x_4 + 45x_5 \\ \text{subject to} & 3x_1 + 4x_2 + x_3 + 2x_4 + 5x_5 \leq 7, \\ & 0 \leq x_j \leq 1, \quad j = 1, 2, \dots, 5. \end{array} \quad (*1)$$

- このように制約条件を緩めることで得られる問題のことを，緩和問題とよびます。
- 問題 (*1) の最適解は，貪欲算法で得られることを示すことができます：
 $(x_1, x_2, x_3, x_4, x_5) = (1, 0, 1, 1, 1/5)$

	品物 1	品物 2	品物 3	品物 4	品物 5
重量	3	4	1	2	5
価格	33	32	10	28	45
価格/重量	11	8	10	14	9

ナップサックの重量制限：7

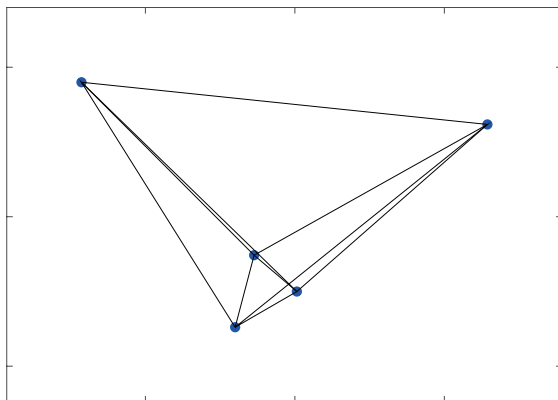
ナップサック問題の $1/2$ -近似解法

- 緩和問題はもとの問題の制約条件を緩めて得られた問題ですから,
(ナップサック問題の最適値) \leq (緩和問題の最適値)
が成り立ちます.
- ナップサック問題に貪欲算法を適用したときの目的関数の値を f^g で表し,
品物 $1, \dots, n$ の価格の最大値を c^{\max} で表すと,
(緩和問題の最適値) $\leq f^g + c^{\max}$
が成り立ちます.
- したがって,
$$\frac{\max\{f^g, c^{\max}\}}{(\text{ナップサック問題の最適値})} \geq \frac{\max\{f^g, c^{\max}\}}{f^g + c^{\max}} \geq \frac{1}{2}$$

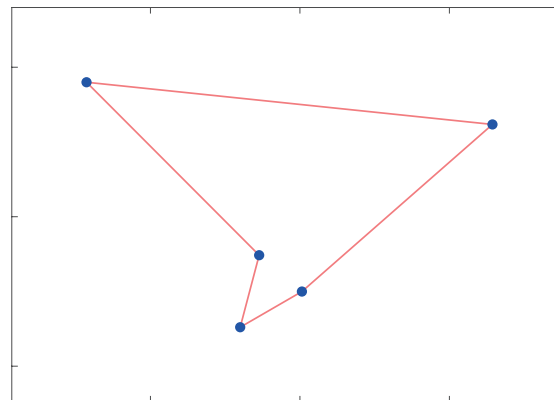
が成り立つことがわかります. つまり, 「貪欲算法の解」と「価格が c^{\max} の品物のみを詰め込む解」のうち目的関数の値が大きい方を出力する解法を考えると, この解法で得られる解の目的関数の値は最適値の $1/2$ 倍以上であることが保証されます.
- このときの比 $1/2$ のことを, 近似比とよびます. また, 近似比が α 以上である解法を, α -近似解法とよびます.

巡回セールスマン問題とは

- いくつかの点と、任意の2点間の距離（正の実数）が与えられているとします。2点間を結ぶ線分を辺とよび、距離をその辺の重みとよびます。
- ある点から出発し、それぞれの点をちょうど1回ずつ訪れてから、もとの点に戻る経路を考えます（このような経路を、巡回路とよびます）。
- 巡回路のうち重みの和が最小のものを求める問題を、巡回セールスマン問題とよびます。
- 下の例では、辺の重みは2点間のユークリッド距離としています：



点が5個ある例題

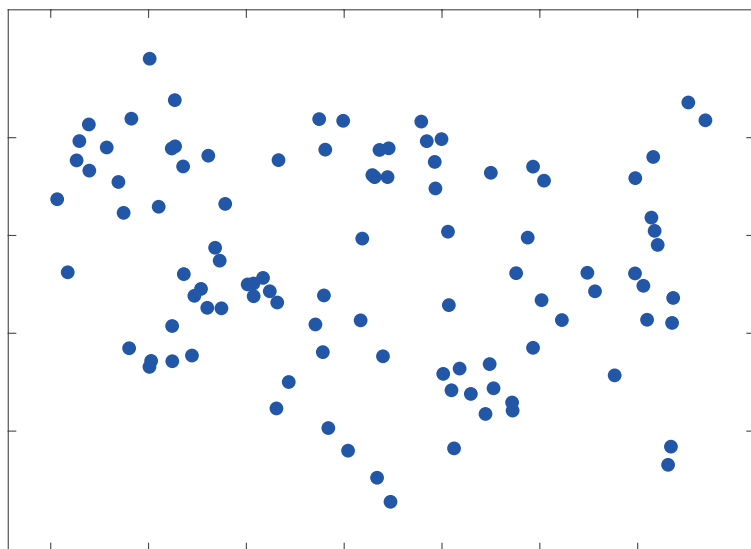


最適解

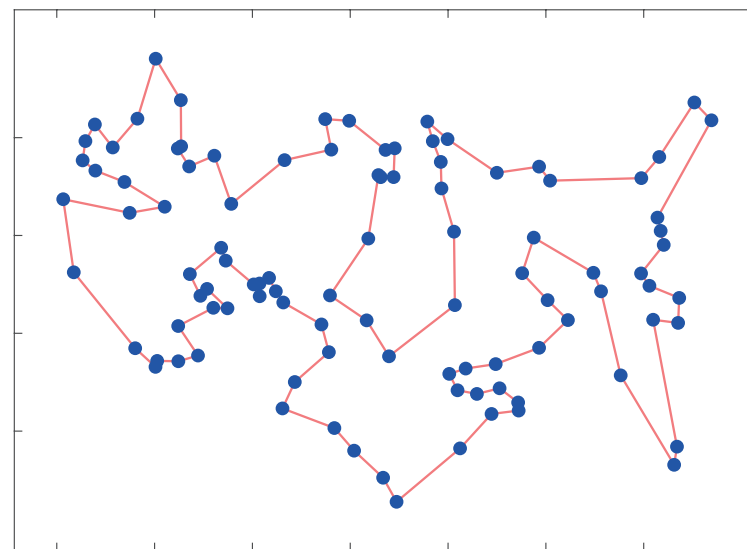
- 巡回セールスマン問題は、それぞれの辺を巡回路に用いるか用いないかという離散的な選択をする問題とみることができますので、離散最適化問題です。

巡回路の個数

- 点が n 個の巡回セールスマン問題を考えます.
- 出発する点を1つ決めると, 巡回路は「残りの $n-1$ 個の点をどの順番に訪れるか」で決まります. したがって, 巡回路は $(n-1)!$ 個だけあります.
- 例として $n = 100$ の場合, 巡回路は $99! \simeq 10^{156}$ 個あります.



$n = 100$ の例題

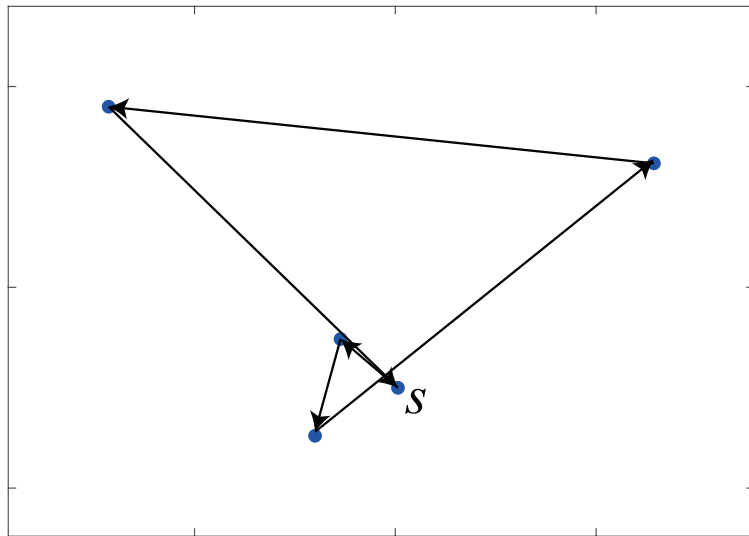


最適解

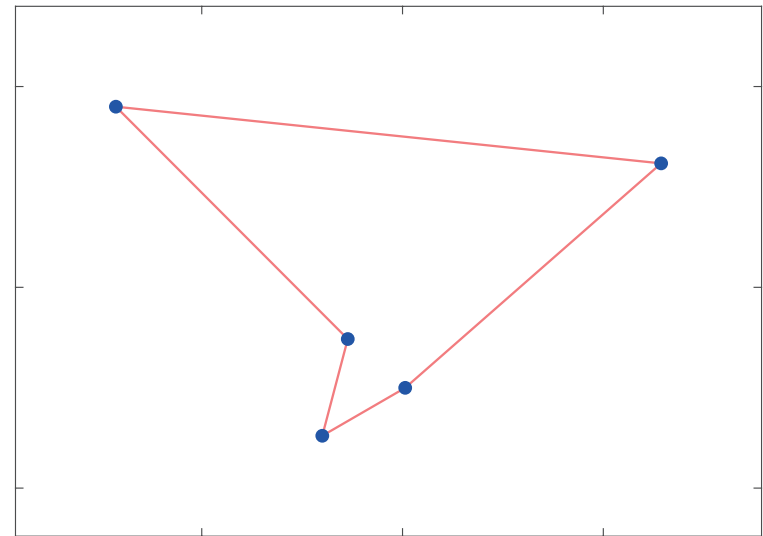
- 制約条件を満たす解の個数は有限個ですが, 非常に多いので, それらを単純に列挙することで最適解を見つけることは事実上不可能です.
- 巡回セールスマン問題は, 解くことが難しい最適化問題の代表例として知られています,

貪欲算法の例

- 巡回セールスマン問題に対して、たとえば次のような貪欲算法が考えられます：
 - ある点 s を出発点とし、まず、 s から最も近い点（対応する辺の重みが最小の点）に移動します。
 - 「まだ訪れていない点のうち、今いる点から最も近い点（対応する辺の重みが最小の点）に移動する」ことを繰り返します。
- 貪欲算法では、必ずしも巡回セールスマン問題の最適解は得られません：



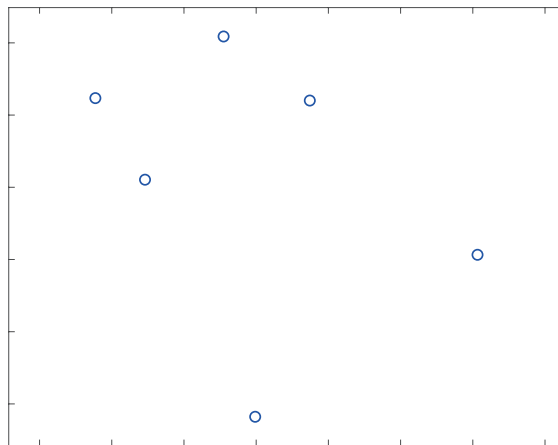
貪欲算法により得られる解



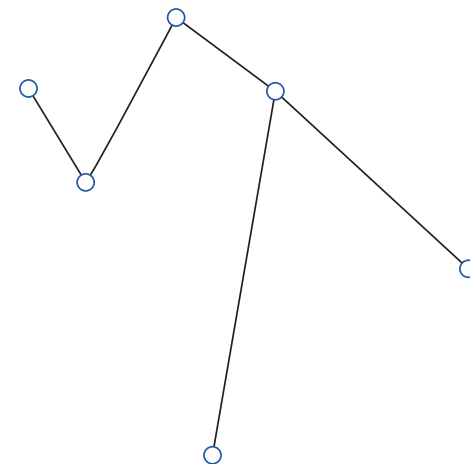
最適解

最小木問題とその貪欲算法

- ある点から出発していくつかの辺をたどって同じ点に戻ることができるとき、それらの辺の集合を閉路とよびます。
- 巡回セールスマン問題の問題設定で、閉路ができないように辺をいくつか選び、すべての点が辺でつながるようにします。点と選んだ辺との組を、全域木とよびます。辺の重みの和が最小の全域木を、最小木とよびます。
- 最小木は、次のような貪欲算法で得られることが知られています（クラスカルのアルゴリズムとよばれています）：
 - 重みが小さい順に、辺を採用します。ただし、その辺を選ぶことで閉路ができる場合には、その辺は採用しません。



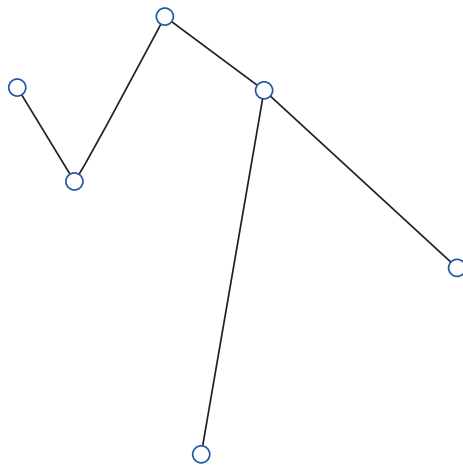
$n = 6$ の例題



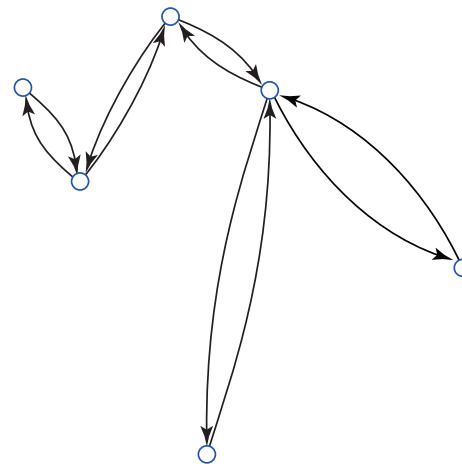
最小木

巡回セールスマン問題の $1/2$ -近似解法 (1/2)

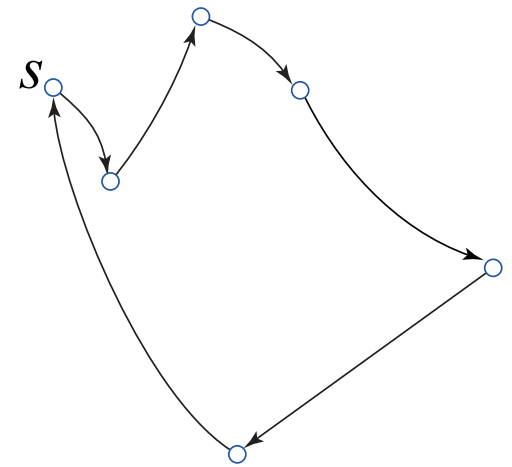
- ここでは，辺の重みは三角不等式を満たしていることを仮定します.
- 下記は，巡回セールスマン問題の $1/2$ -近似解法です：
 - 最小木を求めます.
 - その辺を2本ずつにし，相異なる向きを付けます.
 - ある点 s を出発点とし，向き付きの辺を順にたどります. その際に，既に訪れた点はスキップします.



最小木



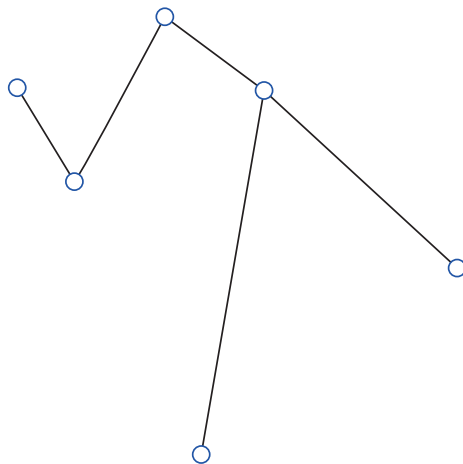
各辺を2本ずつに



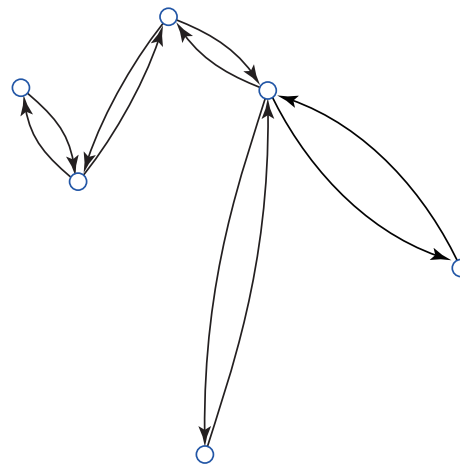
得られた巡回路

巡回セールスマン問題の 1/2-近似解法 (2/2)

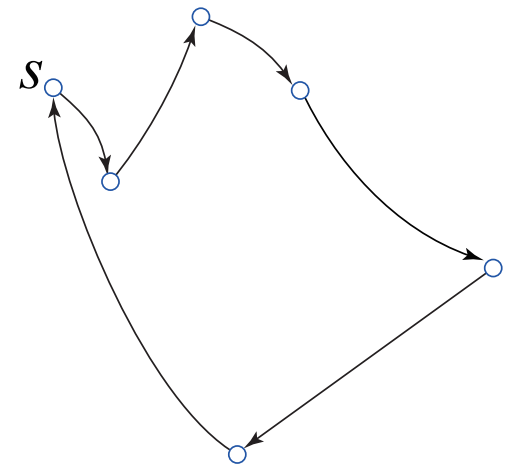
- ここでは、辺の重みは三角不等式を満たしていることを仮定します。
- 近似比が $1/2$ であることは、次のようにして示せます：
 - 最適解から辺を 1 本とり除くと全域木が得られますから、
 $(\text{最適値}) \geq (\text{最小木の重み}) = \frac{1}{2}(\text{真ん中の図の重み})$.
 - 辺の重みに関する三角不等式より、
 $(\text{真ん中の図の重み}) \geq (\text{得られた巡回路の重み})$.



最小木



各辺を 2 本ずつに



得られた巡回路