多項分布

正田 備也

masada@rikkyo.ac.jp

Contents

データのモデリング data modeling

- この授業で扱うのは、統計モデルのうち、観測データが生成 generate される過程を数式で書くことで得られるモデル
- このようなモデルを、生成モデル generative model と呼ぶ
  - 観測データがどのように生成されるかを考えないモデルを、識別モデル discriminative model と呼ぶ
- 観測データは、特定の確率分布にしたがって生成されるものとしてモデル化される
- データを生成する分布のパラメータを推定することが、生成モデルを使った統計モデリングにおける課題となる

観測変数

- データは n 回の観測の結果 $\{x_1, x_2, \ldots, x_n\}$ として表される
- 各 x_i は i 番目の観測値を表す確率変数で、観測変数と呼ぶ

  例 1. 同じ畑で獲れたトマトの重さ（連続量）

  例 2. 同じ文書に出現する単語（離散量）
- 生成モデルを使うときは、各 x_i がどのような値をとるかは特定の確率分布にしたがって確率的に決まると仮定する

  例 1. 同じ畑で獲れたトマトの重さは同じ正規分布にしたがうと仮定

  例 2. 同じ文書に出現する単語は同じカテゴリカル分布にしたがうと仮定
- 観測データが与えられているとき、その観測データを生成する確率分布を推定することが、生成的モデリングの課題

独立同分布の仮定（i.i.d. の仮定）

- n 回の観測のそれぞれ $x_1, \ldots, x_n$ が、同じ分布にしたがい、かつ、独立にしたがう、と仮定することが多い
- 同じ分布にしたがうとは、例えば、単変量正規分布でモデリングする場合、$x_1, \ldots, x_n$ の全てが同じ平均パラメータ μ と標準偏差パラメータ σ をもつ正規分布にしたがうということ
- 独立にしたがうとは、任意の i と j について、x_i がどの値をとるかは x_j がどの値をとるかに “左右されない” ということ
  - 正確には、同時累積分布関数が周辺累積分布関数の積に一致する、ということ（つまり $P(x_i < a, x_j < b) = P(x_i < a)P(x_j < b)$）

例1. 同じ畑で獲れたトマトの重さのモデル化

- 同じ畑で獲れたトマトの重さは、独立に同じ正規分布にしたがうと仮定
- 実際にある畑で獲れた327個の重さを測定したところ、
  $x_1 = 147, x_2 = 171, x_3 = 153, \ldots, x_{327} = 144$ だった（単位g）
- これら測定値の平均は 156.4、標準偏差は 20.8 だった
- ということは、この畑から取れるトマトの重さは、平均が 156.4 で標準偏差が 20.8 の正規分布にしたがうだろう
  - このように、統計モデリングでは、観測データをもとに、それを生成する確率モデルのパラメータを推定する
    - ここで示したのは、最尤推定によるパラメータ推定

例2. 同じ文書に出現する単語のモデル化

- 同じ文書に出現する単語は、独立に同じカテゴリカル分布にしたがうと仮定
- 実際に長さ 4,503 の文書に出現する単語を観測したところ、$x_1 =$ “this”, $x_2 =$ “is”, $\ldots, x_{4503} =$ “pencil” だった
- 各単語の出現回数を求めると “this” は 23 回、“is” は 55 回、“pencil” は 8 回、等々だった
- ということは、この文書での単語の出現は、“this” の出現確率が $\frac{23}{4503}$、“is” の出現確率が $\frac{55}{4503}$、“pencil” の出現確率が $\frac{8}{4503}$、等々であるカテゴリカル分布にしたがうだろう
  - これも、最尤推定によるパラメータ推定

生成モデルの評価

- まず、生成モデルのパラメータの値を推定しておく
  - 最尤推定はパラメータ推定手法の一つ
- そのパラメータの値を使って、未知データの確率を計算する
- 計算された確率の “的確さ” で生成モデルの良し悪しを評価
  - 未知データの確率の高低が、注目している事象に対応しているか
    - 確率が高い＝その生成モデルによって生成されそう
    - 確率が低い＝その生成モデルによって生成されなさそう
    - 的確さ＝注目している事象の起こりやすさに対応

例. センサ測定値の統計モデル

- 特定の環境におかれた特定のセンサの読み値の列が生成される過程を数式で書き、生成モデルを作る
- この統計モデルのパラメータを推定する
- 推定されたパラメータを使って、新たに得られた読み値の列が生成される確率を計算する
- その環境におかれたそのセンサの読み値の列として大いにありうる読み値の列については高めの確率が得られ、そうでない読み値の列については低めの確率が得られれば、良い統計モデルだと評価できるかもしれない

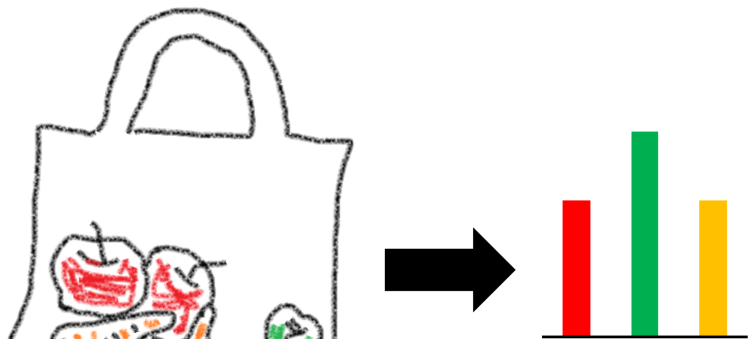
Contents

bag-of-words モデル

- テキストデータのモデリングでは、よく bag-of-words モデルが使われる
- bag-of-words モデルでは、単語の出現順序は考慮しない
- つまり、単語の出現頻度だけを考慮して生成モデルを作る
  - 語順を無視したテキストデータ分析でも、かなりのことが分かる
- 言い換えれば、bag-of-words モデルでは、文書は単語の multiset としてモデル化される
  - multiset とは、同じアイテムが重複して含まれることもある集合
    - 通常の集合 set は、要素はその集合に含まれるか含まれないかのどちらか

例. 買い物かごの中身

例. 買い物かごの中身



出現頻度から確率へ

- 各単語の出現頻度が分かっている
  - 観測データ＝各単語の出現頻度
- これをもとに各単語の出現確率を求める
  - どうすればいい？

例. 前のスライドで、リンゴの出現確率は？

- モデルが違うと答えが違ってくる
- リンゴの出現確率 $= 2/7$ はひとつの答えにすぎない

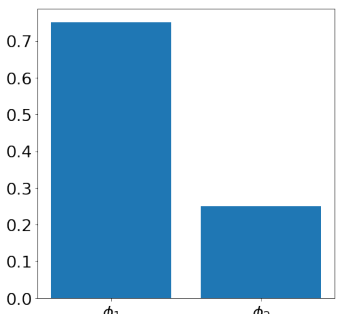
bag-of-words ではないモデル

- i 番目に出現する単語が何か、つまり、x_i がどんな値をとるかが、それ以前にどんな単語が出現したかに依存して決まる、つまり、$x_1, \ldots, x_{i-1}$ という $i-1$ 個の確率変数の値に依存して決まるモデルは、bag-of-words モデルではない
- このように、それまでに出現した単語に依存して次の単語の出現確率が定まるモデルを autoregressive モデルという
  - 逆に言えば、bag-of-words モデルは $p(x_i|x_{i-1}, \ldots, x_1) = p(x_i)$ が成り立つことを仮定している
- autoregressive モデルも、最近はよく使われる

Contents

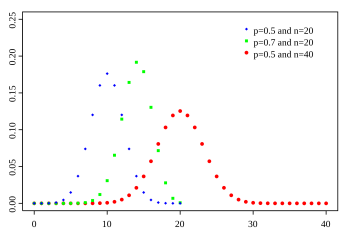
ベルヌーイ分布 Bernoulli distribution

- $V = \{\mathsf{v}_1, \mathsf{v}_2\}$ という 2 種類のアイテムの集合上に定義される確率分布
- パラメータは $\boldsymbol{\phi} = (\phi_1, \phi_2)$
  - アイテム v_1 が出現する確率 ϕ_1
  - アイテム v_2 が出現する確率 ϕ_2
  - $\phi_1 + \phi_2 = 1$ が成り立つので、自由度は 1



二項分布 binomial distribution

- ベルヌーイ分布は1回のコイン投げのモデリングに使う
- 複数回のコイン投げのモデリングには二項分布を使う
- 試行回数を n として、n 回のうち v_1 が k 回出現する確率がこれこれというふうに、$k = 0$ から $k = n$ までのすべての場合に確率を割り振る確率分布が、二項分布
- パラメータは n と $\boldsymbol{\phi} = (\phi_1, \phi_2)$
  - 試行の回数 n（これは観測データから決まる）
  - アイテム v_1 が出現する確率 ϕ_1
  - アイテム v_2 が出現する確率 ϕ_2
  - $\phi_1 + \phi_2 = 1$ が成り立つので、自由度は1



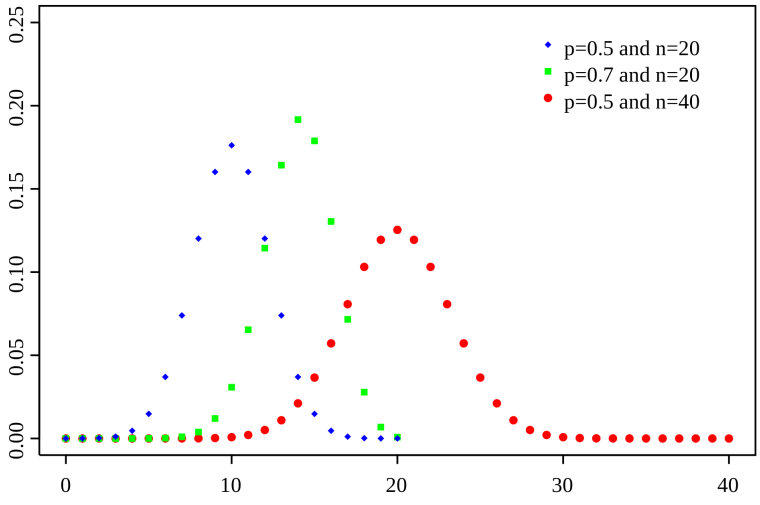


Figure: 二項分布の確率質量関数の例

二項分布の確率質量関数

- 確率質量関数 probability mass function; pmf
  - 離散確率変数に、その値をとる確率を対応させる関数
  - 二項分布の場合は、表が出る回数にその確率を対応させる
- 二項分布の確率質量関数
  - n 回の試行のうち k 回 v_1 が出現する確率は：

$$p(k; \boldsymbol{\phi}, n) = \frac{n!}{k!(n-k)!} \phi_1^k \phi_2^{n-k} \tag{1}$$

  - 「;」は、その右側にある ϕ_1 と n が自由パラメータ（我々が値を指定する必要があるパラメータ）であることを意味する

二項分布の確率質量関数の読み方

例. コイン投げ

- n 回の試行の列は n 個の確率変数 $x_1, \ldots, x_n$ が表裏どちらの値をとるかで表される（例：$x_1 = \mathsf{H}, x_2 = \mathsf{H}, \ldots, x_n = \mathsf{T}$）
- 各試行は、独立に同じベルヌーイ分布に従うと仮定
- すると、表裏が出る順番に関係なく、表が k 回、裏が $n-k$ 回出る試行の列の確率は $\phi_1^k \phi_2^{n-k}$
- つまり、表裏の出る順番が違うだけの試行の列を、二項分布によって区別してモデル化することはできない（頻度が同じなら同じ）
- よって、表 k 回、裏 $n-k$ 回が出る試行の列一つ一つの確率は $\phi_1^k \phi_2^{n-k}$ であっても、二項分布の pmf は $p(k; \boldsymbol{\phi}, n) = \frac{n!}{k!(n-k)!} \phi_1^k \phi_2^{n-k}$

Contents

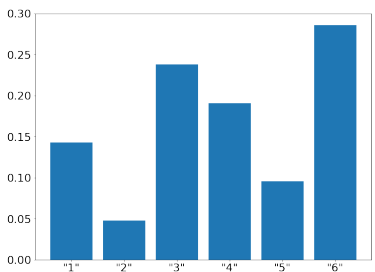
カテゴリカル分布

- $V = \{\mathsf{v}_1, \ldots, \mathsf{v}_W\}$ を W 種類のアイテムの集合とする

  例 1. サイコロの目（$W = 6$）

  例 2. 自然言語の語彙 ($W =$ 数千〜数十万)

- カテゴリカル分布は V 上に定義された離散確率分布
- パラメータは $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_W)$
  - アイテム v_w が出現する確率 ϕ_w
  - $\sum_{w=1}^{W} \phi_w = 1$ を満たす



多項分布 multinomial distribution

- カテゴリカル分布は、1 回の試行のモデリングに使う
- 複数回の独立な試行のモデリングには、多項分布を使う
- 計 n 回の試行のうち各アイテムが何回ずつ出現するか、その可能なすべての場合に確率を割り振る確率分布が多項分布
  - 次のスライド参照
- パラメータは n と $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_W)$
  - 試行の回数 n（観測データから決まる）
  - アイテム v_w の出現確率 ϕ_w（$\sum_w \phi_w = 1$ を満たす）
  - $\sum_w \phi_w = 1$ が満たされるので、自由度は $W - 1$

多項分布はどのような集合の上に定義されるか

- カテゴリカル分布はアイテムの集合の上に定義される
  - すべてのアイテムにわたって確率を合計すると 1 になる
- 多項分布は “計 n 回の試行のうち各アイテムが何回ずつ出現するかの、可能な全ての場合の集合” の上に定義される
  - 多項分布は、各アイテムの出現回数が同じで、出現順が違うだけの試行列を区別できない
  - W 種類のアイテムから重複を許して n 個を選ぶすべての場合にわたって確率を合計すると 1 になる
    - W 種類のアイテムから重複を許して n 個を選ぶ場合の数はいくら？
    - n 個の「◯」と $W-1$ 個の「｜」（仕切り）を並べる場合の数と同じ

例. 「◯ ◯ ｜ ｜ ◯ ｜ ◯ ◯ ◯」は、$n=6$ で、v_1 が 2 回、v_2 が 0 回、v_3 が 1 回、v_4 が 3 回、それぞれ出現する場合を表す

多項分布の確率質量関数

- アイテム v_w の出現回数を c_w と書くことにする
- 総試行回数を n とすると、当然 $\sum_w c_w = n$ が成り立つ
- このとき、多項分布の確率質量関数 pmf は以下のように書ける

$$p((c_1, \ldots, c_W); \boldsymbol{\phi}, n) = \frac{n!}{\prod_w c_w!} \prod_w \phi_w^{c_w} \tag{2}$$

  - $\frac{n!}{\prod_w c_w!}$ の部分は、n 回の試行のうちアイテム v_w が c_w 回出現するような試行の列の総数をあらわしている
  - 多項分布は、各アイテムの出現回数が同じで、出現順が違うだけの試行列を区別できない

Contents

多項分布によるモデリングに登場する変数

- アイテムの出現列を表す観測変数 $\boldsymbol{x} = \{x_1, \ldots, x_n\}$
  - x_i は、i 番目に出現したアイテムを表す確率変数

例. $x_i =$ “apple” は「i 番目に出現する単語は “apple”」という意味

  - 観測変数なので、値はすでに与えられている（値が既知の変数）
- 多項分布のパラメータ $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_W)$
  - ϕ_w は、アイテム v_w の出現確率を表すパラメータ

例. $\phi_w = 0.0013$ は「単語 v_w の出現確率が 0.0013」という意味

  - ϕ_w は値が未知の変数
  - この値の推定が、多項分布によるモデリングにおいて解くべき問題

多項分布の最尤推定

- 観測データ $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ はアイテムの出現の列
- 多項分布によるモデリングでは、出現順序は無視される
- つまり、各アイテム v_w の出現回数 c_w だけが問題とされる
- このとき、観測データ $\boldsymbol{x}$ の尤度は、$\boldsymbol{\phi}$ の関数として、以下のように書ける

$$p(\boldsymbol{x}; \boldsymbol{\phi}, n) = \frac{n!}{\prod_{w=1}^{W} c_w!} \prod_{w=1}^{W} \phi_w^{c_w} \tag{3}$$

- 尤度を最大化する $\boldsymbol{\phi}$ の値を推定値とするのが最尤推定
  - 最尤推定のほかにも $\boldsymbol{\phi}$ の値を推定する方法はある

問題3-1

- 観測データ $\boldsymbol{c} = (c_1, \ldots, c_W)$ の尤度

$$p(\boldsymbol{x}; \boldsymbol{\phi}, n) = \frac{n!}{\prod_{w=1}^{W} c_w!} \prod_{w=1}^{W} \phi_w^{c_w}$$

を最大化する $\boldsymbol{\phi}$ を求めよう

  - ヒント：ラグランジュの未定乗数法を使う（使わなくても解ける）

対数尤度の最大化

- 多くの場合、尤度そのものではなく対数尤度を最大化する
- 答えは同じだが、計算はしやすい

$$\ln p(\boldsymbol{x}; \boldsymbol{\phi}, n) = \sum_{w=1}^{W} c_w \ln \phi_w + const. \tag{4}$$

$$
\begin{aligned}
L(\boldsymbol{\phi}) &= \ln p(\boldsymbol{x};\boldsymbol{\phi},n) + \lambda\Big(1-\sum_{w=1}^{W}\phi_w\Big) \\
&= \sum_{w=1}^{W} c_w \ln \phi_w + \lambda\Big(1-\sum_{w=1}^{W}\phi_w\Big) + const. \qquad (5)
\end{aligned}
$$

とおく。

$$
\frac{\partial L(\boldsymbol{\phi})}{\partial \phi_w} = \frac{c_w}{\phi_w} - \lambda \qquad (6)
$$

$$
\frac{\partial L(\boldsymbol{\phi})}{\partial \lambda} = 1 - \sum_{w=1}^{W}\phi_w \qquad (7)
$$

$\frac{\partial L(\boldsymbol{\phi})}{\partial \phi_w} = 0$ より $\phi_w = \frac{c_w}{\lambda}$ であり、$\frac{\partial L(\boldsymbol{\phi})}{\partial \lambda} = 0$ より $\sum_w \frac{c_w}{\lambda} = 1$ である。
よって、$\lambda = \sum_w c_w$ となり、$\phi_w = \frac{c_w}{\sum_{w'} c_{w'}}$ を得る。

多項分布の最尤推定の答え

$$\phi_w = \frac{c_w}{\sum_{w'} c_{w'}} \tag{8}$$

- 「当たり前では？」と思ってしまうのは、良くないかも
- これはあくまで、ひとつの答え
- 総出現回数 n 回のうち各アイテム v_w が c_w 回ずつ出現しているデータをもとに、各アイテム v_w の出現確率 ϕ_w を推定する方法は、他にもある

Contents

情報検索 information retrieval

- たくさんの文書を持っている
- それらの文書をクエリに適合する（relevant な）順にソート
  - 情報検索とは、このようなことをすること
- どう実装すればいい？
- 実装例
  - ひとつひとつの文書について別々に単語出現確率 ϕ を最尤推定
  - 推定された ϕ を使って、クエリの生成確率を計算
  - この生成確率を高くする順に文書をソート

文書をランキングするための計算

- 上述の最尤推定は、検索対象の文書群のうち d 番目の文書について単語 v_w の出現確率を $\hat{\phi}_{d,w} = \frac{c_{d,w}}{\sum_w n_d}$ と与える
- この単語確率によってクエリ $\boldsymbol{x}_q$ が生成される確率は：

$$p(\boldsymbol{x}_q|\hat{\boldsymbol{\phi}}_d) = \frac{n_q!}{\prod_w c_{q,w}!} \prod_w \left(\frac{c_{d,w}}{n_d} \right)^{c_{q,w}} \tag{9}$$

  - $c_{q,w}$ はクエリにおける単語 v_w の出現頻度
- $p(\boldsymbol{x}_q|\hat{\boldsymbol{\phi}}_d)$ の降順に、文書をソートすればよい

$$p(\boldsymbol{x}_q; \boldsymbol{\phi}_d) = 2 \cdot \phi_{d,\mathrm{apple}}{}^{1} \cdot \phi_{d,\mathrm{pie}}{}^{1}$$

各文書のパラメータの推定値を使ってクエリの確率を求める

$\boldsymbol{\phi}_1 = (\phi_{1,1}, \ldots, \phi_{1,W})$

$\boldsymbol{\phi}_2 = (\phi_{2,1}, \ldots, \phi_{2,W})$

$\boldsymbol{\phi}_3 = (\phi_{3,1}, \ldots, \phi_{3,W})$

$\boldsymbol{\phi}_4 = (\phi_{4,1}, \ldots, \phi_{4,W})$

$\boldsymbol{\phi}_5 = (\phi_{5,1}, \ldots, \phi_{5,W})$

文書ごとに別々のパラメータ集合を用意し
文書ごとに最尤推定する。