知的情報処理
5. ナイーブなベイズ法(2)

櫻井彰人
慶應義塾大学理工学部

本日

- Naïve Bayes 分類器で(というより他の分類器でも)、実は問題になる、些細なしかし極めて重要な点
  - 「度数=0」問題
  - 欠測値問題
  - 連続値の取扱い(これは、離散値を対象とする分類器で、一般的な問題)

目次

- 復習
  - 簡単な例。データの数を数える。
  - 推論をしよう。度数が0の場合。
- 「度数=0」問題
  - 二項分布のパラメータ推定、最尤推定量
  - Laplace correction
  - Rでは?
- その他の問題
  - 欠測値問題
  - 数値属性の取り扱い
    - 離散化。連続分布を仮定する
- 文書分類

再掲

簡単な例で:
天気とテニス

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

(テニスを行う) Play=Yes と(テニスを行わない) Play=No の2つのクラスがある

このとき、下記の(未知、つまり学習データにない)条件では、Play=Yesであった(あろう)かPlay=Noであった(あろう)かを推定する。

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Tom Mitchell の Machine Learning という書籍から. よく使われます

再掲

(回りくどいが)データをクラスに分割

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

再掲

データの数を数えて、推定

		A1=Outlook		A2=Temperatur		A3=Humidity		A4=Windy	
度数	Sunny	2	Hot	2	High	3	False	6	
	Overcast	4	Mild	4	Normal	6	True	3	
	Rainy	3	Cool	3					
	合計	9	合計	9	合計	9	合計	9	
確率の推定	Sunny	2/9	Hot	2/9	High	3/9	False	6/9	
	Overcast	4/9	Mild	4/9	Normal	6/9	True	3/9	
	Rainy	3/9	Cool	3/9					

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

		A1=Outlook		A2=Temperatur		A3=Humidity		A4=Windy	
度数	Sunny	3	Hot	2	High	4	False	2	
	Overcast	0	Mild	2	Normal	1	True	3	
	Rainy	2	Cool	1					
	合計	5	合計	5	合計	5	合計	5	
確率の推定	Sunny	3/5	Hot	2/5	High	4/5	False	2/5	
	Overcast	0/5	Mild	2/5	Normal	1/5	True	3/5	
	Rainy	2/5	Cool	1/5					

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

ポイント

- クラス（出力値）ごとに数える。
  - 条件付き確率（すなわち、クラス="テニスに行く"ときに、、、、である確率）を考えることになる
- 属性ごとに、各属性値の発生回数を数える
  - 「属性ごとに」というのが "naïve" Bayes
  - 発生回数は、各属性値の発生確率を推定するため
  - 二項分布・多項分布を考えている。

$$
\begin{aligned}
p(m_j \mid x) &= p(x, m_j) / p(x) \\
&= p(x \mid m_j) p(m_j) / p(x) \\
&= p(a_1, \ldots, a_n \mid m_j) p(m_j) / p(x) \\
&= p(m_j) \prod_{i=1}^{n} p(a_i \mid m_j) \Big/ p(x)
\end{aligned}
$$

7

再掲

一つの表に纏めておこう

p(m) に関する説明を省きましたが（忘れた、が正しいのだが）、それは、これ

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

A1=Outlook			A2=Temperature			A3=Humidity			A4=Windy			m=Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

8

再掲ではない

推論をしよう

未知の x

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

$$
\begin{aligned}
p(m_j \mid x) &= p(x, m_j) / p(x) \\
&= p(x \mid m_j) p(m_j) / p(x) \\
&= p(a_1, \ldots, a_n \mid m_j) p(m_j) / p(x) \\
&= p(m_j) \prod_{i=1}^{n} p(a_i \mid m_j) \Big/ p(x)
\end{aligned}
$$

付け加えました

p(Play=yes | x)
= p(Outlook=Sunny | Play=yes)
  * p(Temp=Cool | Play=yes)
  * p(Humidity=High | Play=yes)
  * p(Windy=True | Play=yes)
  * p(Play=yes) / p(x)
= (2/9) * (3/9) * (3/9) * (3/9)
  *(9/14) / p(x)
= 0.0053 / p(x)

p(Play=no | x)
= p(Outlook=Sunny | Play=no)
  * p(Temp=Cool | Play=no)
  * p(Humidity=High | Play=no)
  * p(Windy=True | Play=no)
  * p(Play=no) / p(x)
= (3/5) * (1/5) * (4/5) * (3/5)
  *(5/14) / p(x)
= 0.0206 / p(x)

言い換えれば、p(Play=yes | x) < p(Play=no | x)
すなわち、「テニスは行わなかった（行わないだろう）」

注: 1/p(x) は気にしなくてよいことが分る; 比較すべき相手すべてに共通だから.

9

再掲ではない

推論をしよう

未知の x

Outlook	Temp.	Humidity	Windy	Play
Overcast	Cool	High	True	?

$$
\begin{aligned}
p(m_j \mid x) &= p(x, m_j) / p(x) \\
&= p(x \mid m_j) p(m_j) / p(x) \\
&= p(a_1, \ldots, a_n \mid m_j) p(m_j) / p(x) \\
&= p(m_j) \prod_{i=1}^{n} p(a_i \mid m_j) \Big/ p(x)
\end{aligned}
$$

p(Play=yes | x)
= p(Outlook=Overcast | Play=yes)
  * p(Temp=Cool | Play=yes)
  * p(Humidity=High | Play=yes)
  * p(Windy=True | Play=yes)
  * p(Play=yes) / p(x)
= (4/9) * (3/9) * (3/9) * (3/9)
  *(9/14) / p(x)
= 0.0106 / p(x)

p(Play=no | x)
= p(Outlook=Overcast | Play=no)
  * p(Temp=Cool | Play=no)
  * p(Humidity=High | Play=no)
  * p(Windy=True | Play=no)
  * p(Play=no) / p(x)
= (0/5) * (1/5) * (4/5) * (3/5)
  *(5/14) / p(x)
= 0.0000 / p(x)

言い換えれば、p(Play=yes | x) > p(Play=no | x)
すなわち、「テニスは行った（行うだろう）」

これでいいのか？

注: 1/p(x) は気にしなくてよいことが分る; 比較すべき相手すべてに共通だから.

再掲ではない

仮に

簡単な例で: 天気とテニス

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Mild	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Mild	Normal	False	Yes
Rainy	Mild	High	True	No

（テニスを行う）Play=Yes と（テニスを行わない）Play=No の2つのクラスがある

このとき、下記の（未知、つまり学習データにない）条件では、Play=Yesであった（あろう）かPlay=Noであった（あろう）かを推定する。

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	True	?

11

Tom Mitchell の Machine Learning という書籍から. よく使われます

再掲ではない

データの数を数えて、推定

		A1=Outlook		A2=Temperatur		A3=Humidity		A4=Windy	
度数	Sunny	2	Hot	0	High	3	False	6
	Overcast	4	Mild	6	Normal	6	True	3
	Rainy	3	Cool	3				
	合計	9	合計	9	合計	9	合計	9
確率の推定	Sunny	2/9	Hot	0/9	High	3/9	False	6/9
	Overcast	4/9	Mild	6/9	Normal	6/9	True	3/9
	Rainy	3/9	Cool	3/9				

Outlook	Temp.	Humidity	Windy	Play
Overcast	Mild	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Mild	Normal	False	Yes

		A1=Outlook		A2=Temperatur		A3=Humidity		A4=Windy	
度数	Sunny	3	Hot	2	High	4	False	2
	Overcast	0	Mild	2	Normal	1	True	3
	Rainy	2	Cool	1				
	合計	5	合計	5	合計	5	合計	5
確率の推定	Sunny	3/5	Hot	2/5	High	4/5	False	2/5
	Overcast	0/5	Mild	2/5	Normal	1/5	True	3/5
	Rainy	2/5	Cool	1/5				

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

12

再掲ではない

一つの表に纏めておこう

p(m) に関する説明を省きましたが（忘れた、が正しいのだが）、それは、これ

Outlook	Temp.	Humidity	Windy	Play
Overcast	Mid	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Overcast	Cool	Normal	True	Yes
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mid	High	True	Yes
Overcast	Mid	Normal	False	Yes

Outlook	Temp.	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Rainy	Cool	Normal	True	No
Sunny	Mild	High	False	No
Rainy	Mild	High	True	No

A1=Outlook			A2=Temperature			A3=Humidity			A4=Windy			m=Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3	Hot	0	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	6	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	0/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	6/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

再掲ではない
の
再掲でもない

推論をしよう

未知の x

Outlook	Temp.	Humidity	Windy	Play
Overcast	Hot	High	True	?

$$p(m_j \mid x) = p(x, m_j)/p(x)$$
$$= p(x \mid m_j)p(m_j)/p(x)$$
$$= p(a_1, \ldots, a_n \mid m_j)p(m_j)/p(x)$$
$$= p(m_j)\prod_{i=1}^{n} p(a_i \mid m_j)/p(x)$$

p(Play=yes | x)
= p(Outlook=Overcast | Play=yes)
  * p(Temp=Hot | Play=yes)
  * p(Humidity=High | Play=yes)
  * p(Windy=True | Play=yes)
  * p(Play=yes) / p(x)
= (4/9) * (0/9) * (3/9) * (3/9)
  *(9/14) / p(x)
= 0.0000 / p(x)

p(Play=no | x)
= p(Outlook=Overcast | Play=no)
  * p(Temp=Hot | Play=no)
  * p(Humidity=High | Play=no)
  * p(Windy=True | Play=no)
  * p(Play=no) / p(x)
= (0/5) * (2/5) * (4/5) * (3/5)
  *(5/14) / p(x)
= 0.0000 / p(x)

言い換えれば、p(Play=yes | x) = p(Play=no | x)
すなわち、どっちともいえない？？？

これでいいのか？

注: 1/p(x) は気にしなくてよいことが分る; 比較すべき相手すべてに共通だから.

目次

- 復習
  - 簡単な例。データの数を数える。
  - 推論をしよう。度数が0の場合。
- 「度数=0」問題
  - 二項分布のパラメータ推定、最尤推定量
  - Laplace correction
  - Rでは？
- その他の問題
  - 欠測値問題
  - 数値属性の取り扱い
    - 離散化。連続分布を仮定する
- 文書分類

Laplace correction これ1枚

頻度=0 問題

- もしあるクラス値に対してある属性値が一度も起こらなかったらどうなるか (e.g. “Play=No” のとき “Outlook = Overcast”)?
  - 確率 P(Outlook=Overcast | Play=no) は 0 !!
- 従って, 事後確率は 0 !!
  - 他の値がどんなに “これは起こりやすい!” と言っていても, だ.
- 治療方法:
  - すべての クラス値-属性値 の組に頻度 1 を加える (Laplace correction という);
  - 分母（正確ではない. 右を参照）には k (可能な属性値の個数) を加える (勿論, これと合せて Laplace correctionという).

P(Play=yes | E)
= P(Outlook=Sunny | Play=yes) *
  P(Temp=Cool | Play=yes) *
  P(Humidity=High | Play=yes) *
  P(Windy=True | Play=yes) *
  P(play=yes) / P(E)
= (2/9) * (3/9) * (3/9) * (3/9) *(9/14) / P(E)
= 0.0053 / P(E)

ではなく:

‘Outlook’ のとりうる値の個数

= ((2+1)/(9+3)) * ((3+1)/(9+3)) *
  ((3+1)/(9+2)) * ((3+1)/(9+2)) *(9/14) /
  P(E)
= 0.007 / P(E)

‘Windy’ のとりうる値の個数

目次

- 復習
  - 簡単な例。データの数を数える。
  - 推論をしよう。度数が0の場合。
- 「度数=0」問題
  - 二項分布のパラメータ推定、最尤推定量
  - Laplace correction
  - Rでは？
- その他の問題
  - 欠測値問題
  - 数値属性の取り扱い
    - 離散化。連続分布を仮定する
- 文書分類

以下、脱線 or 補足

二項分布のパラメータ推定

- 画鋲を放り投げる（toss）と, 2つのうちのいずれかの状態で床に落ちる: *Head* または *Tail*
- θ で (未知の) 確率 $P(H)$ をあらわす

推定課題:
一連の toss の結果 D=x[1],x[2],…,x[M] をもとに, 確率 $P(H)$= θ と $P(T)$=1- θ を推定したい

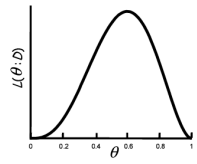
尤度関数

- 特定のθのよさはどう計るか?
  「そのθのもとで、観測データが生成されたとする」仮説の
  ありそう度合い(likelihood; 尤度)で計ればよい

$$L(\theta : D) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

Dは発生してしまった事象。その確率なんて考えられるのか?

- 例えば, 列 H,T,T,H,H に対しては :



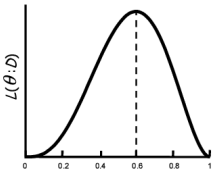
$$L(\theta : D) = \theta \cdot (1-\theta) \cdot (1-\theta) \cdot \theta \cdot \theta$$

最尤推定量は

- 最尤推定の原理 :

尤度関数を最大にするパラメータ値をとれ

- 今の例では :



$$\hat{\theta} = \frac{N_H}{N_H + N_T}$$

これが最適だと考えるのが普通だが…

最尤推定量でいいか?

- 2値ではなく、多値であったら、それもかなり多かったら:
  - 同じ原理が適用できる. でも、観測されない値が出るであろう. その頻度は 0 と考えてよいか? いや、たまたま、出現しなかっただけという可能性が結構ある …
- 例えば、変数Xは、値 x1, x2,…,x20 を等確率で取るとする。サンプルは30個しかない。となったら、実現していない変数値は、いくつもありそう。例えば、Rで

```
> set.seed(100)
> x <- sample(1:20, 30, replace=T)
> table(x)
x
 2  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 1  2  1  3  1  4  1  2  4  1  1  2  2  2  1  2
```


では、どうしたらよいか?

- Laplace correction を考える
- 「パラメータも分布する」と考える
  - ベイズ的考え方
  - つまり、P(D|θ) ではなく、P(D|θ)P(θ) を考える
  - これは、ベイズの定理を用いると

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

$$P(\theta|N_H, N_T) = \frac{\theta^{N_H}(1-\theta)^{N_T}P(\theta)}{P(D)}$$

  - となるゆえ、最も確率の高い θ を推定値とすればよい。
  - このとき、N_H が0であっても、θ=0 は推定値とはならない。

Laplace correction 前後

A1=Outlook			A2=Temperature			A3=Humidity			A4=Windy			m=Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3	Hot	0	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	6	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	0/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	6/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

改変後の値です

A1=Outlook			A2=Temperature			A3=Humidity			A4=Windy			m=Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	3	4	Hot	1	3	High	4	5	False	7	3	10	6
Overcast	5	1	Mild	7	3	Normal	7	2	True	4	4		
Rainy	4	3	Cool	4	2								
Sunny	3/12	4/8	Hot	1/12	3/8	High	4/11	5/7	False	7/11	3/7	9/14	5/14
Overcast	5/12	1/8	Mild	7/12	3/8	Normal	7/11	2/7	True	4/11	4/7		
Rainy	4/12	3/8	Cool	4/12	2/8								

目次

- 復習
  - 簡単な例。データの数を数える。
  - 推論をしよう。度数が0の場合。
- 「度数=0」問題
  - 二項分布のパラメータ推定、最尤推定量
  - Laplace correction
  - Rでは?
- その他の問題
  - 欠測値問題
  - 数値属性の取り扱い
    - 離散化。連続分布を仮定する
- 文書分類

ちょっとわき道。しかし、これも重要

Sub 目次: Laplace correction

- Laplace の「連」の法則
  - コイン投げで、表が続けて10回出たら、次に表が出る確率はいくらか？
    - 答え 11/12
  - 最尤推定とは（一応）別の考え方
- Laplace correction:
  - 生起回数 0 の現象があるときの、生起確率の推定
    - そうしたことを想定して、0 でない事象に対しても、生起確率の推定の仕方を変える
  - Laplaceの「連」の法則の結果と同じことを，最尤法の枠組みで得る方法
- Laplace correction の拡張

次の車もアンティーク？

Laplace's Law of Succession

- ラプラスの「連の法則」とでも訳しましょうか。
- 同じ事象がn回連続して起こった場合に、その事象がn+1回目も起こる確率は、$(n+1)/(n+2)$ である。

- この定理の不思議さ:
  - 普通に、コイン投げを考えよう。
  - 表ばかり10回続いた。表が出る確率は最尤推定すると 10/10 = 1 である。従って、次に表となる確率は1である。
    - まぁ、10回も続けば、いかさまコインだと思うよね。
    - しかし、コインであれば、裏がないということはなかろう
  - ラプラスによれば、11/12
    - いかさまには変わりはない。

最尤推定 vs Laplace

- 10回などといわずに、2回にしてみよう。すなわち、表が2回続いたとしよう。
- 最尤推定なら、表が出る確率は 2/2=1ゆえ、次に表の出る確率は1。
- Laplaceによれば、3/4。
- どっちが正しそうか？

- では、表が1回続いた（続いてない？）
- 最尤推定なら、表の出る確率は1/1=1ゆえ、次に表の出る確率は1。
- Laplaceによれば、2/3。

- さて、どちらが正しそうか？

証明

- ここでは、ベルヌーイ試行を考える。
- ただしコインはN+1種類あり（表の出る確率は異なる）、どのコインであるかは、出題者は知っているが回答者は知らないとする。
- i番目のコインの表の出る確率は i/N $(i=0,..,N)$ とする。
- 出題者は、N+1個のコインを等確率で選び、一回選べば、あとはそれを使い続けるとする。
- n回連続して表が出たときに、n+1回目も表が出る確率を求めたい。

- コインiを選んだという条件下で、n回連続して表が出る確率は $(i/N)^n$ である。従って、n回連続して表が出る確率は $(1/(N+1))\sum_{i=0}^{N}(i/N)^n$ これを $F_{N,n}$ と表す。
- n+1回連続して表が出る確率は $F_{N,n+1}=(1/(N+1))\sum_{i=0}^{N}(i/N)^{n+1}$ である
- 従って、求める確率は $F_{N,n+1}$ / $F_{N,n}$ である（式は複雑）。そこで次頁。

証明（続）

- $N\to\infty$ の極限を考えよう。
  - これは、コインの表が出る確率が、[0,1] 上の一様分布に従うことを意味する。
- $F_{N,n}=(1/(N+1))\sum_{i=0}^{N}(i/N)^n \to \int_0^1 x^n\,dx = 1/(n+1)$
- 従って、求める確率は、$N\to\infty$ のとき、$F_{N,n+1}/F_{N,n} \to F_{\infty,n+1}/F_{\infty,n} = (n+1)/(n+2)$

- つまり、より正確に述べれば、Laplace's law of succession は、(coin tossingを例にすれば）出題者は、表が出る確率を一様分布に従ってランダムに決め、それを用いてcoin tossをする、一方回答者は、表が出る確率を知らないで（しかし、一様分布に従っていることは知っている）、n回連続して表が出たときに、n+1回目も表が出る確率を求めるという状況において、その確率は、$(n+1)/(n+2)$ であると主張している。

Laplace correction

Laplace smoothing とも言います

- これは、先ほどの問題で、最尤法を用いてより直感に合う結果を得る方法でもある。
- 再び最尤推定で考えよう。
- 最初からn回続けて表が出れば、表が出る確率の最尤推定量は $n/n = 1$.
- しかし、これはおかしい。裏が出る確率が0というのはありえない。仮に、回答者が見る前に、表・裏一回ずつ出たものとしてみよう（これが Laplace correction）。
- そうすると、全部で n+2回試行し、n+1回表が出たことになるので、表が出る確率の最尤推定量は、$(n+1)/(n+2)$

- Laplace's law of succession の結果とぴったり合う！

パラメータの事前分布

Bayesの定理使用

- 最尤推定ではなく、事後確率分布によるパラメータの期待値を推定値としてみよう。すなわち、表が出る確率 p を
  $\bar{p} = \int_0^1 p\, P(\, p \mid n\text{回表}\,)\, dp$
  で推定したい。

–ではなく∧にしたいのですが、できませんでした。

- この推定を行うには、パラメータ p の事前分布を知る必要がある。仮に、$[0,1]$上の一様分布だと考えよう。そうすると、
  $\bar{p} = \int_0^1 p\, P(n\text{回表} \mid p)\, P(p)/P(n\text{回表})\, dp$
  $= \int_0^1 p^{n+1} / (\int_0^1 p^n\, dp)\, dp$
  $= (n+1)/(n+2)$

Bayesの定理による

$P(n\text{回表})$
$= \int_0^1 P(n\text{回表} \mid p) P(p) dp$
$= \int_0^1 p^n\, dp$

- これはLaplace's law of successionの結果に一致する！
- すなわち、Laplace's law of succession の結果は（Laplace correctionの結果は）、パラメータの事前分布を一様分布と仮定し、事後分布によるパラメータの期待値を推定値とすることに相当する！

先ほどの Laplace's law of succession の証明と同じことを示しているだけ。

パラメータの事前分布（続）

- 一様分布では気に食わない？
  - 誰でもそう思う。
- では、ベータ分布 $f(x;\alpha,\beta)=x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha,\beta)$ を考えよう。なお$f(x;1,1)$が一様分布である。
  - なぜベータ分布か？って？ 二項分布の双対分布だから。
  - なぜ双対分布か？って？ そうでないと計算ができないから、、、、
- このとき、事後分布によるパラメータの期待値を推定値とすると、その値は、$(n+\alpha)/(n+\alpha+\beta)$ となる

- これをLaplace correction の拡張として用いることができる。$(n+1)/(n+2)$ の代わりに $(n+1/2)/(n+1)$ や$(n+1/3)/(n+2/3)$ などを用いる方法となる。
- もっとも、そうなると、どれを使ったらよいのだろうか？という迷いが生まれる
  - 選択肢が多いと困ることもある

マーケティングでよく知られた話（とはいえ、議論があるので注意のこと）

- 実は、（理論的な）答えは、ない

Iyengar SS, Lepper MR. When choice is demotivating: can one desire too much of a good thing? J Pers Soc Psychol. 2000 Dec;79(6):995-1006.
Scheibehenne, B., Greifeneder, R. & Todd, P. M. (2009). What moderates the too-much-choice effect? Psychology & Marketing, 26, 229-253.

Laplace correction の拡張

- Laplace's law of succession において、$(n+1)/(n+2)$ の代わりに $(n+\alpha)/(n+\alpha+\beta)$ を用いる。例えば、$(n+1/2)/(n+1)$ や$(n+1/3)/(n+2/3)$ などを用いる。
- もっとも、どれがよいかは、統計的にはいえない。

- これは、パラメータ分布の事前分布として、ベータ分布 $f(x;\alpha,\beta)=x^{\alpha-1}(1-x)^{\beta-1}/B(\alpha,\beta)$ を考えたことに相当する。
  - $f(x;1,1)$が一様分布なので、これは 基本Laplace correction の拡張といえる。

補足: スムージング

- 確率モデルの推定において、訓練データに出現しない事象に対して微小な確率値を割り当てること。平滑化とも呼ばれる。

http://www.jaist.ac.jp/project/NLP_Portal/doc/glossary/index.html

- 自然言語処理では、単語や文字 n 個の連なり（n-gram）の出現頻度をよく用いる。n が少し大きくなると、出現しない単語列・文字列が出てくる。それらを出現頻度0とするといろいろまずいことが起こる。そこで、様々なスムージング法が提案されてきた。
  - Laplaceスムージング（加算スムージング）
  - 線形補間法(Interpolation)
  - グッドチューリング
  - カッツ・スムージング
  - チャーチ・ゲイル・スムージング
  - ウィトン・ベル・スムージング
  - Kneser-Neyスムージング
  - ….
  - 階層的Pitman-Yor言語モデル

目次

Laplace correction: Rでは？

```
m <- naiveBayes(xy[,-5], xy[,5],laplace=1) #とすればよい
```

```
> m <- naiveBayes(xy[,-5], xy[,5],laplace=1)
> table(xy[,5],predict(m,xy[,-5]))

     No Yes
  No  4   1
  Yes 0   9
> m

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = xy[, -5], y = xy[, 5], laplace = 1)

A-priori probabilities:
xy[, 5]
       No       Yes
0.3571429 0.6428571

Conditional probabilities:
       Outlook
xy[, 5]  Overcast     Rainy     Sunny
    No  0.1250000 0.3750000 0.5000000
    Yes 0.4166667 0.3333333 0.2500000

       Temp.
xy[, 5]       Cool       Hot      Mild
    No  0.2500000 0.3750000 0.3750000
    Yes 0.3333333 0.2500000 0.4166667

       Humidity
xy[, 5]      High    Normal
    No  0.7142857 0.2857143
    Yes 0.3636364 0.6363636

       Windy
xy[, 5]     False      True
    No  0.4285714 0.5714286
    Yes 0.6363636 0.3636364
```

Laplace correction: Rでは？（続）

参考: 違いを探してください

```
> library(e1071)
> setwd("D:/R/Sample")
> xy<-read.csv("PlayTennis.csv", header=TRUE)
> m <- naiveBayes(xy[,-5], xy[,5])
> table(xy[,5],predict(m,xy[,-5]))

     No Yes
  No  4   1
  Yes 0   9
> m

Naive Bayes Classifier for Discrete Predictors

Call:
naiveBayes.default(x = xy[, -5], y = xy[, 5])
```

```
A-priori probabilities:
xy[, 5]
       No       Yes
0.3571429 0.6428571

Conditional probabilities:
       Outlook
xy[, 5]  Overcast     Rainy     Sunny
     No  0.0000000 0.4000000 0.6000000
     Yes 0.4444444 0.3333333 0.2222222

       Temp.
xy[, 5]       Cool       Hot      Mild
     No  0.2000000 0.4000000 0.4000000
     Yes 0.3333333 0.2222222 0.4444444

       Humidity
xy[, 5]       High    Normal
     No  0.8000000 0.2000000
     Yes 0.3333333 0.6666667

       Windy
xy[, 5]      False      True
     No  0.4000000 0.6000000
     Yes 0.6666667 0.3333333
```

目次

- 復習
  - 簡単な例。データの数を数える。
  - 推論をしよう。度数が0の場合。
- 「度数=0」問題
  - 二項分布のパラメータ推定、最尤推定量
  - Laplace correction
  - Rでは？
- その他の問題
  - 欠測値問題
  - 数値属性の取り扱い
    - 離散化。連続分布を仮定する
- 文書分類

38

次の話題

欠測値（属性値が不明）問題

- 問題: 属性 *A* の値がない事例があるとどうなるか?
  - しばしば, 訓練時やテスト時に, 必ずしも全ての属性値が入手できるとは限らない
    - これを欠測値という
  - 例: 医療診断
    - <*Fever* = *true*, *Blood-Pressure* = *normal*, …, *Blood-Test* = *?*, …>
    - 値は、本当になかったり, またあっても信頼度が低かったりする
  - 欠測値対応策: 学習時 versus 分類時
    - 学習時: その属性値のみ無視をする（数えない）
    - 分類時: その属性値の確率は参入しない（1と数える）

39

対策例: 欠測値問題

- 一つの単純だが乱暴な解決案:
- **学習時**: 当の サンプル は頻度計算には算入しない
- **分類時**: 確率の計算から当の属性は省く
- 例:

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

P(Play=yes | E)
= P(Temp=Cool | Play=yes) *
P(Humidity=High | Play=yes) *
P(Windy=True | Play=yes) *
P(Play=yes) / P(E)
= (3/9) * (3/9) * (3/9) *(9/14) / P(E)
= 0.0238 / P(E)

P(Play=no | E)
= P(Temp=Cool | Play=no) *
P(Humidity=High | Play=no) *
P(Windy=True | Play=no) *
P(Play=no) / P(E)
= (1/5) * (4/5) * (3/5) *(5/14) / P(E)
= 0.0343 / P(E)

P(E)を求めれば: P(Play=yes | E) = **41%**, P(Play=no | E) = **59%**

次の話題

数値属性の取扱い

- 初めから数値属性を扱っている分類器では、考える必要がない。
- しかし、nave Bayes 分類器や、決定木では考える必要がある。

- 2つの方法がある。
  - 一つは、離散化する方法、
  - 他の一つは、その属性値の分布としてある分布を仮定し、数値から、確率に換算する

  方法である。

41

対策1: 離散化する方法

- Vector quantization（ベクトル量子化）がある。
- 多くはもっと単純な方法で、そこそこに大丈夫。
- Naïve Bayes の各属性は一次元なので、より簡単にできる。
  - 決定木と同様である。
- 例えば、

  Fayyad, U.M. and Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In Proc. of the 13th IJCAI (pp. 1022–1027)

  これは Weka に入っている

42

対策2: 分布を仮定する

- よくある仮定: 属性値は正規分布をなす（クラス値が所与）
- 正規分布（ガウス分布）のパラメータは2個. 推定値は:
- 標本平均 μ $\hat{\mu} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
- 不偏分散 σ^2 $\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$
- 密度関数 f(x): $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$

R の e1071 パッケージの naiveBayes では、この方法がとられている。実際、The standard naive Bayes classifier (at least this implementation) assumes independence of the predictor variables, and Gaussian distribution (given the target class) of metric predictors.

天気とテニスの例（改変）

Outlook			Temperature			Humidity			Windy			Play	
	Yes	No		Yes	No		Yes	No		Yes	No	Yes	No
Sunny	2	3		83	85		86	85	False	6	2	9	5
Overcast	4	0		70	80		96	90	True	3	3		
Rainy	3	2		68	65		80	70					
				…	…		…	…					
Sunny	2/9	3/5	mean	73	74.6	mean	79.1	86.2	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	std dev	6.2	7.9	std dev	10.2	9.7	True	3/9	3/5		
Rainy	3/9	2/5											

確率値

$$f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi}6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

44

未知データの分類

・ 持ちデータ E:

Outlook	Temp.	Humidity	Windy	Play
Sunny	66	90	True	?

P(Play=yes | E) =
P(Outlook=Sunny | Play=yes) *
P(Temp=66 | Play=yes) *
P(Humidity=90 | Play=yes) *
P(Windy=True | Play=yes) *
P(Play=yes) / P(E)
= (2/9) * (0.0340) * (0.0221) * (3/9) *(9/14) / P(E)
= 0.000036 / P(E)

P(Play=no | E) =
P(Outlook=Sunny | Play=no) *
P(Temp=66 | Play=no) *
P(Humidity=90 | Play=no) *
P(Windy=True | Play=no) *
P(Play=no) / P(E)
= (3/5) * (0.0291) * (0.0380) * (3/5) *(5/14) / P(E)
= 0.000136 / P(E)

45

目次

- 復習
  - 簡単な例。データの数を数える。
  - 推論をしよう。度数が0の場合。
- 「度数=0」問題
  - 二項分布のパラメータ推定、最尤推定量
  - Laplace correction
  - Rでは？
- その他の問題
  - 欠測値問題
  - 数値属性の取り扱い
    - 離散化。連続分布を仮定する
- 文書分類

46

文書分類

- 文書分類とは:
  - 文書（メール、ニュース、webページ等々。それらの一段落ということも、また、一文ということも）を分類すること
  - 分りやすいのは、メールがスパムか否かの分類
  - ブログを、スプログか否かに分類する、という課題もある
  - ニュースが（ある人にとって）興味のあるものか否かを分類する、というのもある。さらに、
  - ある商品の評判を（良い評判も悪い評判も）集めるにも「分類」が必要。そして、良い評判と悪い評判とに分ける。
  - レビューを、信頼できる評価か信用できない評価かに分けるのも、文書分類
  - アンケート調査のうち、自由記述文の分類。
  - コールセンターでの、QAの分類

47

文書分類の学習

- 共通性質:
  - 例えば、「こういう特徴があれば、スパムメールである」という分類規則を、人間が作るのは、大変。
    - 最初に作るのが大変
    - 規則に誤りや過不足が多いので、それを調べ、訂正するのが大変。
    - 変化に追従するのが大変。
  - 「学習」できると便利（実際には、「必須」）

- Naïve Bayes が結構うまくいく
  - どうやって Naïve Bayes を用いるか?
  - ポイント: どう事例（すなわち、1文書）を表現するか? 属性は何か?

http://www.state-itc.org/rtc2004/accessible/4_Managing_Risks_files/images/image66.png

Using Text Categorization Techniques for Intrusion Detection
From http://www.usenix.org/events/sec02/tech.html

最も簡単な表現方法

- Bag-of-words, すなわち、袋一杯の単語 or 袋詰めの単語
  - ある文書を、それぞれの単語が何回現れたかで表現する。
    - "Bag" で、もとの文書のどこにあったかを忘れることを表している。
    - また、単語の連なりも考えないことを表している。
    - 例えば、仮に、「慶應」「義塾」「大学」がそれぞれ単語なら、「慶應義塾大学」も「慶應大学義塾」も「義塾慶應大学」も同じと考えることになる。
  - 「何を単語とするか」が結構重要。文書ごとに変わってはいけない。
    - 英語であれば、dog と dogs といったような語形変化は無視した方がよい。
  - 文書分類に役立ちそうもない単語は考えない
    - 日本語で言えば、助詞（は、が、も、や、、、）がその代表。
    - 英語で言えば、前置詞がその代表
    - こういった、文法機能を持ち、単語単独では意味のない単語を機能語という
  - ノイズの可能性が高い単語は考えない。
    - 文書集合内で、出現頻度が極めて低い（一回等）もの

49

最も簡単な表現方法

- この表現って naïve Bayes?
  - ベイズ推論とは直接には関係しないので、naïve Bayes ではないが、naïve な表現であることは間違いない。
  - しかし、naïve Bayes 的に、文書の出現確率を書くことができる。
  - 文書の属するクラスごとに、文書内にある特定の単語が出現する確率 $P(w_1|c_j), P(w_2|c_j), \ldots, P(w_n|c_j)$ が決まっているとすると、w_1, w_2,…, w_n が文書中に含まれる単語であるとき、そのような文書が出現する確率を次のように書く

$$P(\mathrm{doc}|c_j)=P(w_1|c_j)^{\mathrm{TF}(w1)}\,P(w_2|c_j)^{\mathrm{TF}(w2)}\ldots\,P(w_n|c_j)^{\mathrm{TF}(wn)}$$

ただし TF(w) は単語 w の doc における出現度数(term frequency)

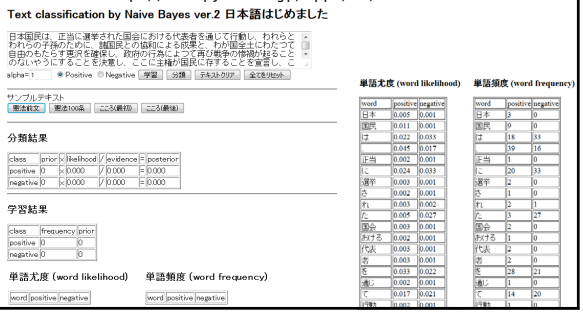
出現確率をこう書けば naïve Bayes といえよう

50

デモ

Text classification by Naive Bayes ver.2 日本語はじめました

http://sleepyheads.jp/apps/nb/nb2.html



Naïve Bayes による文書分類

- ある文書 doc につき

$$c_{NB} = \arg\max_{c_j \in C} P(c_j) \prod_{w_k \in Voc} P(w_k \mid c_j)^{TF(w_k,doc)}$$

ただし、$TF(w_k,\mathrm{doc})$=doc中のw_kの出現度数、Voc は全単語（考えている全単語）集合とした

- 単語の出現確率については、Laplace correction が必須。そこで、下記の推定値を使用; ただし、n_j=クラスc_j中の全単語出現度数, $n_{k,j}$=クラスc_j中の単語w_k出現度数

$$P(w_k \mid c_j) = \frac{n_{k,j}+1}{n_j+|Voc|}$$

52

Twenty News Groups (Joachims 1996)

- 各グループ1000の訓練文書
- 新規の文書を、もとのnewsgroupに割振る

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x & rec.sport.hockey	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

T. Joachims. *A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization.* In Proceedings of the 14th International Conference on Machine Learning, Nashville, TN, 1997, pp.143--151.

サンプル

Xref: cantaloupe.srv.cs.cmu.edu misc.headlines:41573 talk.politics.guns:53299

--- 略

Lines: 57
NNTP-Posting-Host: sandman.caltech.edu
manes@magpie.linknet.com (Steve Manes) writes:

>hambidge@bms.com wrote:
>: In article <C4psoG.C6@magpie.linknet.com>, manes@magpie.linknet.com (Steve Manes) wr

>: >: Rate := per capita rate. The UK is more dangerous.
>: >: Though you may be less likely to be killed by a handgun, the average
>: >: individual citizen in the UK is twice as likely to be killed
>: >: by whatever means as the average Swiss. Would you feel any better
>: >: about being killed by means other than a handgun? I wouldn't.
>:
>: >What an absurd argument. Switzerland is one-fifth the size of the
>: >UK with one-eigth as many people therefore at any given point on
>: >Swiss soil you are more likely to be crow bait. More importantly,
>: >you are 4x as likely to be killed by the next stranger approaching
>: >you on a Swiss street than in the UK.

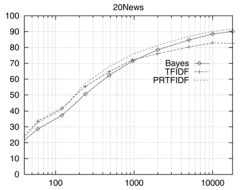
54

Killed by handgun, or killed? If I'm dead, I don't much care if it

Twenty News Groups (Joachims 1996)

- Naive Bayes: 89% 分類正解率
  - 頻出単語上位100 個 (the, and, of, ...) は除去
    - このように文法機能を担う単語や、文書を類別するのに有効でない単語を stop words として除去するのが普通
  - 頻度が3回に満たない単語は除去
  - 残った単語は、約 38,500 語

ただし、この正解率は高すぎ。20 Newsgroups の各投稿には、分類に非常に役立つ subject フィールドがある。今ではこれらは除去することになっているが、当時では、除去せずに、分類実験をした可能性がある。



20 Newsgroups: Rでは？

正味40MB ぐらいしかないのですが

- データが多すぎて、Rのパッケージに含まれる naïve Bayes 分類器は使えない。
  - データ行列（さきほどのRプログラムでは、xy, xy, tt といった行列）が巨大になる（行数が文書数（約2,000）、列数が単語数（約40,000））。
  - しかし、非零要素は非常に少ないので、スパース行列表示を用いればよい。
  - それでもオーバーヘッドが大きい。
  - それなら自分でプログラムを書いてしまおう。
- なお、Weka にもスパース行列が表現できて、原理的には取り扱える。しかし、大きなメモリが必要で、しかも遅い。

20 Newsgroups: データ

- "20 Newsgroups" というサイトにあり
  - http://qwone.com/~jason/20Newsgroups/
- 前処理（単語の切り出し等）が終わって、単語の個数のデータに編集が終わったものを用いる。Matlab で使いやすい形になっている。
  - 20news-bydate-matlab.tgz
  - 05data-20news-bydate-matlab.zip として講義資料サイトに
- このうち、train.data, train.label, test.data, test.label を用いる。
- プログラムは資料として web頁に掲載しておきます。
- 結果のうち、confusion matrix を次頁に示します。
- 正解率は、約78.2%です。


```
> cm
          correct
predicted   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
        1 237   3   3   0   0   0   0   1   0   4   2   0   2  10   3   7   2  12   7  47
        2   0 299  33   8   8  42   9   1   1   1   0   5  18   7   8   2   0   1   1   3
        3   0   7 208  15  10   8   4   0   0   0   0   1   0   1   0   1   0   0   0   0
        4   0  12  58 306  38  10  49   2   0   1   0   1  28   3   0   0   0   0   0   0
        5   0   7  11  21 275   2  21   0   0   1   0   2   8   0   0   1   1   0   0   0
        6   1  21  30   2   3 306   1   1   0   2   0   1   3   0   2   2   0   0   1   0
        7   0   1   0   4   4   1 227   5   1   3   1   1   1   1   0   0   2   0   0   0
        8   0   3   2   6   4   0  32 356  25   3   1   0   9   3   0   0   2   2   1   0
        9   0   1   2   0   1   2   5   4 353   1   0   0   2   0   1   0   1   1   0   1
       10   0   0   2   0   1   1   0   2   2 345   4   0   0   2   0   0   1   1   0   0
       11   1   0   1   1   0   0   1   0   0  16 381   0   0   0   1   0   0   1   0   0
       12   1  16  17   5   5  10   3   1   1   2   1 361  45   0   3   1   3   4   3   1
       13   1   4   1  23  16   0  11   4   1   2   0   3 260   3   4   0   0   0   0   0
       14   2   3   4   0   7   0   2   0   1   0   2   2   6 324   4   1   1   0   3   3
       15   3   6   4   1   2   3   3   2   0   0   1   0   3   3 333   0   2   0   7   5
       16  43   4   5   0   0   1   3   0   1   3   2   2   6  16   5 377   3   7   2  69
       17   3   0   0   0   3   1   1   5   4   1   0   7   0   3   1   2 324   3  95  19
       18   9   0   0   0   0   1   3   1   2   2   1   0   2   6   2   2   2 323   5   5
       19   7   2   9   0   6   2   6   9   5   9   3   8   0  10  24   1  16  21 184   8
       20  10   0   1   0   0   0   1   1   0   1   0   1   0   1   1   1   4   0   1  90
>
```


まとめ

- Naïve Bayes 分類器は簡単で強力
- Naïve Bayes 分類器で考えるべき重要な点
  - 「度数=0」問題
  - 欠測値問題
  - 連続値の取扱い（これは、離散値を対象とする分類器で、一般的な問題）
- 有効な応用分野としての文書分類がある

本日の課題

  - 難しくはないのですが（つまり過去のプログラムのコピー＆ペースト＋編集でできるのですが）、注意深く行う必要があります。
- COM実験で用いた文字認識のデータを用います。
- 学習データは optdigits.tra.csv、テストデータは optdigits.tes.csv です。
- 読み込むとき、read.csv の引数に colClasses="factor" を追加してください。
  - これは、数値データを factor として読み込むためのおまじないです。
  - なお、この csv ファイルには、ヘッダーはありません。ですから、read.csvの引数を一箇所直す必要があります。考えてください。ヤマ勘であたります。
- 属性の個数が異なります。以前は（次のスライドにコピーがあります）、5個の属性でしたが、今度は65個あります。以前は5番目が分類すべきクラスでしたが、今度は65番目が分類すべきクラス（今回は数字の種類（0～9））です。
- テストデータがたくさんありますので、一個ずつの予測は印字しないで、confusion matrix のみ印字してください。
  - Confusion matrix の作り方は、次頁に書きました。

パッケージ e1071 をインストールしてください。
library(e1071) として下さい

本日の課題の参考

```
> xy<-read.csv("PlayTennis.csv", header=TRUE)
> xyt<-read.csv("PlayTennisTest02.csv",header=TRUE,as.is=TRUE)
> tt<-data.frame(factor(xyt[,1],levels=levels(xy[,1])))
> for (i in 2:5) {
+   tt<-data.frame(tt,factor(xyt[,i],levels=levels(xy[,i])))
+ }
> names(tt)<-names(xy)
> m <- naiveBayes(xy[,-5], xy[,5])
> # prediction and confusion matrix for test data
> predictedClassTest <-predict(m, tt)        # prediction
> table(tt[,5], predictedClassTest)          # confusion mtrx
     predictedClassTest
      No Yes
  No   1   0
  Yes  0   1
```

予測した No/Yes

tt[,5] に書いた No/Yes

PlayTennisTest02.csv

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	No
Rainy	Mild	Normal	False	Yes

本日の課題に関するメモ

(1) 変数 tt を作るときに、例えば、

```
tt<-data.frame(factor(xyt[,1],levels=levels(xy[,1])))
for (i in 2:5) {
  tt<-data.frame(tt,factor(xyt[,i],levels=levels(xy[,i])))
}
```

としたあと、間違い(5ですね)に気が付き

```
for (i in 2:65) {
  tt<-data.frame(tt,factor(xyt[,i],levels=levels(xy[,i])))
}
```

とすると、このループの中で tt は「前の値に基づいて次の値を作る」という動作をしているため、間違った tt にさらに（今度は正しい値ですが）値を継ぎ足すということをしてしまいます。

対策：一行前の tt の初期化の部分から実行しなおす必要がある。つまり

```
tt<-data.frame(factor(xyt[,1],levels=levels(xy[,1])))
for (i in 2:65) {
  tt<-data.frame(tt,factor(xyt[,i],levels=levels(xy[,i])))
}
```

です。

(2-1) colClasses="factor" の追加し忘れ。これは追加すればよい。例えば、

```
xy<-read.csv("optdigits.tra.csv", header=FALSE,
                                  colClasses="factor")
```

とすればよい。
(2-2) colClasses="factor" をコピーし、MsWordファイルにペーストし、MsWordファイルから、コピーしてRにペーストする。そうすると、MsWordの機能で、ダブルクォートが全角文字に変わってしまう。こんな風に。
colClasses=”factor”
このときRは何も言わずに何もしないようです。
何も言わないので、例えば、変数xy ができたように見えますが、実は作成されていません。

対策：pdfファイルから直接コピーペーストする、手で打ち込む、または、（MsWordではなく）メモ帳経由にする。

62

本日の課題＋α

ちょっとした「＋α」です。興味のある方、どうぞ。

- 本日の課題において、学習データ及びテストデータの、confusion matrix と accuracy を求めなさい。
  - accuracy は、分類正解数 / データ数 です。
  - accuracy は、（それぞれ）式一つで求まります。

63