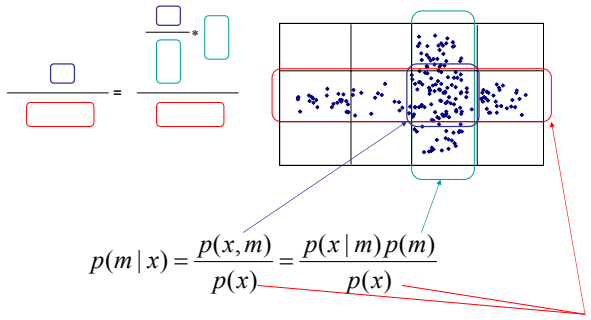
知的情報処理
4. ナイーブなベイズ法

櫻井彰人
慶應義塾大学理工学部

Bayesの定理



$$p(m \mid x) = \frac{p(x,m)}{p(x)} = \frac{p(x \mid m)p(m)}{p(x)}$$

生成モデルのベイズ推定

条件付き確率　事前確率

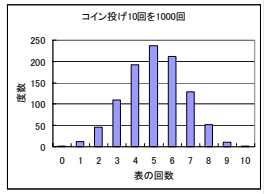
$$\underbrace{p(m \mid x)}_{\text{事後確率}} = \frac{p(x,m)}{p(x)} = \frac{\overbrace{p(x \mid m)}\overbrace{p(m)}}{p(x)}$$

を用いて、x が観測されたとき、最もありうる m をその事後分布を用いて推定することを考える。

- p(m) はクラスmの頻度で推定すればよい
- では、p(x|m) はどうしたら推定できるだろうか？
  - p(x|m) はモデルmからデータxが生成される確率を表す。個別のxに対する p(x|m) を知るには、任意のxに対するp(x|m)をしっているればよい（あたりまえ）。一般形はモデルの記述そのものである。
- その問題を解決した一つの方法が naïve Bayes法

属性数が問題

- 前のスライドの図を思い出してください。
- コイン投げをして、コインの表が出る確率を推定する問題と考えてください。
- 正解は 0.5 です。
- しかし、10回投げたうち、3回以下しか表が出ない場合が169回、7回以上出てしまう場合が192回もある。
- つまり、2値属性のパラメータを1個推定するにも、サンプル10個では不足だということである。
- 独立な属性が10個あれば、それらのパラメータをまあまあの精度で推定するには、10 2^{10} =1万個のサンプルが必要になる。
- 一般にはなかなか難しい。



では、どうするか？

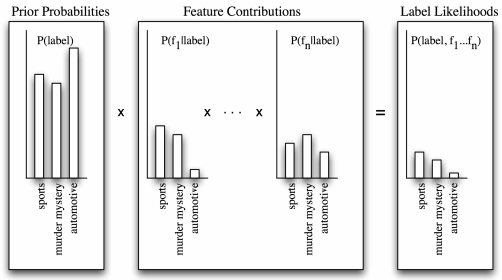
定式化: Bayes 推論使用時の課題

- サンプルの属性 $x = <a_1,\ldots,a_n>$が与えられたとき, x が属するクラス v を最尤推定するには?

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, \ldots, a_n)$$
$$= \arg\max_{v_j \in V} \frac{P(a_1, a_2, \ldots, a_n \mid v_j) P(v_j)}{P(a_1, a_2, \ldots, a_n)}$$
$$= \arg\max_{v_j \in V} P(a_1, a_2, \ldots, a_n \mid v_j) P(v_j)$$

m　x　Bayesの定理
無視しても結構

- 問題: $P(a_1 \ldots a_n \mid v_j)$ を評価するのに大量のデータが必要. 2値属性としても、属性数が n なら 10 2^n 個程度のサンプルが必要



http://nltk.googlecode.com/svn/trunk/doc/images/naive_bayes_bargraph.png

Naïve であること

- ありえない仮定をおく
- Naïve Bayes の仮定: 属性同士は、属するクラスが所与なら、独立である。すなわち
  - $P(a_1,\ldots,a_n|v_j) = P(a_1|v_j)\ P(a_2|v_j)\ \ldots\ P(a_n|v_j)$
  - *条件付独立性*（クラスが所与の時）とも
- この仮定のもと, v_{MAP} は

$$v_{NB} = \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n \mid v_j) P(v_j)$$
$$= \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i \mid v_j)$$

「ある事例について」を明記すると

- ある事例 E の属するクラス（事例に未知の属性があって、それが「クラス」であってよい）を推定する

$$v_{MAP} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, ..., a_n, E)$$
$$= \arg\max_{v_j \in V} \frac{P(a_1, a_2, ..., a_n \mid v_j, E) P(v_j \mid E)}{P(a_1, a_2, ..., a_n \mid E)}$$
$$= \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n \mid v_j, E) P(v_j \mid E)$$

- naïve Bayesであれば

$$v_{NB} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, ..., a_n, E)$$
$$= \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n \mid v_j, E) P(v_j \mid E)$$
$$= \arg\max_{v_j \in V} P(v_j \mid E) \prod_i P(a_i \mid v_j, E)$$

蛇足: naïve とは

英辞郎より

「ナイーブ」とはかなり意味が違う。ご注意。

- 【名】うぶな人
- 【形】
- 世間知らずの、〔世間を〕なめた、経験の少ない、〔思考が〕単純な、だまされやすい、ばか正直な、うぶな、無警戒な、認識の甘い、愚直な
  ・How can you be so naive? : よくそんなに単純でいられるな―。
  ・In spite of her appearance, she has a naive side. : ああ見えても彼女はうぶなところがある。
  ・Society is not so naive. : 世間はそれほど甘いものではない。
  ・You're naive! : まだ青いな！
  ・You're too naive. : おまえは世間知らずだ。
- 純真な、純情な、天真らんまんな、あどけない、素朴な、純朴な、無邪気な、初々しい
- 批判する能力に欠ける
- 〔マウスなどが〕実験未使用の、投薬を受けたことがない

http://www.elfwood.com/~shuo2/The_Naive_One_(colored).3217704.html
http://blogs.uct.ac.za/blog/call-me-cassandra/page/7

解説が必要ですね

- 次に、確率変数間の独立性。
- 二つの確率変数 X, Y 間では常に P(X,Y)=P(X)P(Y) となるなら、この二変数は独立であるという
- XとYの間に因果関係や相関関係があると独立ではなくなる。
- 独立なら、因果関係や相関関係がないということ。

- しかし、ここで（naïve Bayesで）重要なことは、確率の計算が簡単になることであって、理論的に or 実質的に独立かどうかではない。
- どうして簡単になるかって？

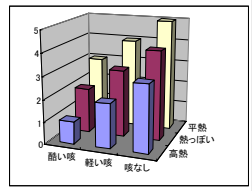
上右図: http://img.sparknotes.com/101s/psychology/01.3.variables.jpg
確率変数では、「片方が独立、他方が独立でない」ということはない

解説が必要ですね

- まず、何をすべきであったか？
- すべきことは、p(m) と p(x|m) の推定。
- どうすればできるか？
- 属性もクラスも離散変数であるとしよう。
- p(m) は、例えば、p(風邪)とp(¬風邪)であるが、これは、数を数えれば、推定できる。
  - 統計を取ればよい。p(風邪)は「全患者数分の風邪の患者数」で推定できる。
- p(x|m)も同様である。数を数えればよい。
  - X=<咳、熱>であるとすれば、例えば、p(酷い咳, 高熱|風邪) は、「風邪の患者数分の酷い咳・高熱がある患者数」で推定できる。

独立でないと何が大変か

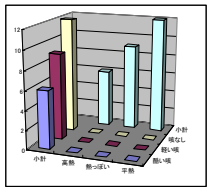
- Xの値は、酷い咳、軽い咳、咳なし、Yの値は高熱、熱っぽい、平熱、とする。
- まずは、独立でないときの話し。
- モデル（分布）を推定するには、これら3x3個の組合せについて、頻度を推定する必要がある。
  - 数を数えればよい
- これらの組合せは排他的だから、一つのサンプルは一箇所にしか入れない。
- 頻度を推定するには、（確率0を除き）各組合せに最低でも1個のサンプルが必要（勿論、本当に1個しかないのではお話にならないが）。
- つまり、3値の属性がn個あれば3^n個、2値の属性がn個あれば2^n個のサンプルは、絶対に必要（それでも不十分）ということになる。
- 係数を気にしなくてすむようにオーダー記法を用いれば、$O(3^n)$ 個や $O(2^n)$ 個 必要、ということになる

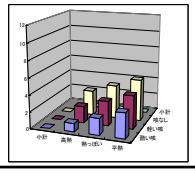


この図（いい加減な図です）では、9個の属性値組合せに対し、27個のサンプルとなっている

独立であると何がよいか

- 独立であるとしよう。
- p(X,Y)=p(X)p(Y) であるから、3x3個のp(X,Y)を推定するのに、実は、3個のp(X)と3個のp(Y)を推定すればよいことになる
  - 周辺分布を推定すればよいということ
- もともと、3x3個の組合せについて頻度を推定する必要があるのだが、3 + 3 個に対して推定すればよいことになる。
- つまり、絶対に必要なサンプル数が、3値の属性がn個あれば3n個、2値の属性がn個あれば2n個に減る。
- つまり、必要なサンプル数は、オーダー記法を用いれば、O(n) 個ということになる。
- なお、排他性も弱くなっている。属性一つ一つの中では排他性がある（一つのサンプルは一つの属性値しかとれない）が、複数個の属性に渡る排他性はない。右図で、「熱」の中、「咳」の中では排他的だが、それだけ、ということ。





独立ならよいことばかりか？

- 本当に独立なら、よいことばかりである。
- しかし、本当に独立なわけはない
  - 咳が酷ければ、喉が炎症を起こし、熱が出る
- 独立でないのに、独立を仮定すると何が起こるか？
- めちゃくちゃになる（何をしているか分からなくなる）はず。
  - 実際、naïve Bayesによって推定した確率値はまったく合っていないといわれている。
- しかし、実際には、naïve Bayes がうまく機能することが多い。これは、
  - 誤った独立性仮定による誤りの増加より、独立性仮定によってパラメータ数を減らしてパラメータの推定精度を向上させたことによる誤りの減少が勝っている
    - 分布を推定しているわけではなく、クラス・分類を推定しているのである。
  - 実際には、独立でなくとも独立として十分近似できることが多い

からではないかと考えられる。

多項分布

- 実は、これまで曖昧にしてきたのだが、確率分布として、多項分布を仮定していた。
  - 確率変数が離散値をとるときには、よく使用する
- 多項分布は
  - n 個の独立なベルヌーイ試行の結果の分布であり
  - 各試行の結果は有限個（k個）の値をとり、
  - それぞれの値をとる確率は $p_1, ..., p_k$
  - 確率変数 X_i は n回の試行で i 番目の結果が起こる回数を示すとするときの確率分布であり、
  - それは

$$p(x_1,...,x_k;n,p_1,...,p_k)=\frac{n!}{x_1!\cdots x_k!}p_1^{x_1}\cdots p_k^{x_k}$$

となる

多項分布: naïve Bayes の場合

- naïve Bayes の場合、各属性値は、各インスタンス（各人）について、そうなるかならないか（高熱か否か）を表していることが多い。
- 前のスライドでいえば、x_i =1 or 0 というような分布である。
- 例えば、p_1=高熱となる確率、p_2=熱っぽい確率、p_3=平熱の確率、とすれば、ある人について

$$p(\text{高熱})=\frac{1!}{1!0!0!}p_1^1p_2^0p_3^0=p_1,\ p(\text{熱っぽい})=\frac{1!}{0!1!0!}p_1^0p_2^1p_3^0=p_2,\ p(\text{平熱})=\frac{1!}{0!0!1!}p_1^0p_2^0p_3^1=p_3$$

- これを n 人の風邪の患者（n個の事例（風邪患者））について調べたところ、x_1人が高熱、x_2人が熱っぽく、x_3人が平熱であったとしよう。n人の風邪患者についてそうなる確率は、

$$p(x_1,x_2,x_3;n,p_1,p_2,p_3)=\frac{n!}{x_1!x_2!x_3!}p_1^{x_1}p_2^{x_2}p_3^{x_3}$$

多項分布: パラメータ推定

- これを n 人の風邪の患者（n個の事例（風邪患者））について調べたところ、n_1人が高熱、n_2人が熱っぽく、n_3人が平熱であったとしよう。n人の風邪患者についてそうなる確率は、

$$p(n_1,n_2,n_3\mid n,p_1,p_2,p_3)=\frac{n!}{n_1!n_2!n_3!}p_1^{x_1}p_2^{x_2}p_3^{x_3}$$

- n_1,n_2,n_3 が与えられたとき、これを最大にする p_1,p_2,p_3は

$$\hat{p}_1=\frac{n_1}{n},\hat{p}_2=\frac{n_2}{n},\hat{p}_3=\frac{n_3}{n}$$

- すなわち、$\left\langle\frac{n_1}{n},\frac{n_2}{n},\frac{n_3}{n}\right\rangle$ は $\langle p_1,p_2,p_3\rangle$ の最尤推定量である

本来は最初のスライド: Naïve Bayes 分類方法

- 単純だが（だから？）よく知られた方法
  - 単純な割には高精度
  - 単純なだけに、高速
- Bayes 推論を用いて分類クラスを決定する分類課題において、属性値の（分類クラスを条件とする）条件付独立性を仮定した方法。
- Bayes 定理 + 仮定 *条件付独立*
  - 実際には成り立たないことが多い仮定
  - それにも関わらず, 実際にはしばしば、良い結果が得られる
- 成功事例:
  - 文書分類
  - その他、なんでも。まずはこれを試す

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

未知の一日

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

この例では, 結果があるというより, (属性Playの値が不明な)ある事例があるというべきであろう

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

天気とテニス

$$v_{NB} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, ..., a_n, E)$$
$$= \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n \mid v_j, E) P(v_j \mid E)$$
$$= \arg\max_{v_j \in V} P(v_j \mid E) \prod_i P(a_i \mid v_j, E)$$

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

← テストサンプル E

Outlook=Sunny,…,Windy=True を A と略記
E が右表と同じ情報源から得られたとすると

P(Play=yes | A, E)
= P(A | Play=yes, E) * P(Play=yes | E)
= P(Outlook=Sunny | Play=yes, E)
  * P(Temp=Cool | Play=yes, E)
  * P(Humidity=High | Play=yes, E)
  * P(Windy=True | Play=yes, E)
  * P(Play=yes, E) / P(E)

さて、P(Outlook=Sunny | Play=yes, E) 等はどう考えたらよいのか？

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

天気とテニス: パラメータ推定

- n回の事例の Outlook について調べたところ、n_1回が sunny、n_2回が overcast、n_3回がrainyであったとしよう。そうなる確率は、 $\frac{n!}{n_1! n_2! n_3!} p_1^{n_1} p_2^{n_2} p_3^{n_3}$
- ただし、p_1= P(Outlook=sunny | 右表) , p_2=P(Outlook=overcast | 右表), p_3=P(Outlook=rainy | 右表)
- p_1, p_2, p_3 の最尤推定量は、n_1/n, n_2/n, n_3/n となる

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

各確率（の推定値）を計算しよう

Outlook			Temperature			Humidity			Windy			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								

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

天気とテニス

$$v_{NB} = \arg\max_{v_j \in V} P(v_j \mid a_1, a_2, ..., a_n, E)$$
$$= \arg\max_{v_j \in V} P(a_1, a_2, ..., a_n \mid v_j, E) P(v_j \mid E)$$
$$= \arg\max_{v_j \in V} P(v_j \mid E) \prod_i P(a_i \mid v_j, E)$$

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

← テストサンプル E

Outlook=Sunny,…,Windy=True を A と略記
E が右表と同じ情報源から得られたとすると

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

1/P(E) は気にしなくてよい; 比較すべき相手すべてに共通だから.

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

天気とテニス

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

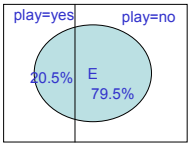
← テストサンプル E

P(Play=no | A, E)
= P(A | Play=no, E) * P(Play=no | E)
= P(Outlook=Sunny | Play=no, E)
  * P(Temp=Cool | Play=no, E)
  * P(Humidity=High | Play=no, E)
  * P(Windy=True | Play=no, E)
  * P(Play=no, E) / P(E)
= (3/5) * (1/5) * (4/5) * (3/5) *(5/14) / P(E)
= 0.0206 / P(E)

右上図: http://www.ucsactive.org.uk/pages/53_tennis.aspx

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

“気にしなくてよい” 項 P(E) は



P(Play=yes, E) + P(Play=no, E) = P(E) i.e.

P(Play=yes, E)/P(E) + P(Play=no, E)/P(E) = 1 i.e.

P(Play=yes | E) + P(Play=no | E) = 1 i.e.

0.0053 / P(E) + 0.0206 / P(E) = 1 i.e.

P(E) = 0.0053 + 0.0206

というわけで,

P(Play=yes | E) = 0.0053 / (0.0053 + 0.0206) = **20.5%**

P(Play=no | E) = 0.0206 / (0.0053 + 0.0206) = **79.5%**

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=Yes” のとき “Humidity = High”)?
  - 確率 P(Humidity=High|Play=yes) 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

欠測値問題: 属性値が不明

- 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
  - 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: 訓練時 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)

欠測値問題

- 一つの単純だが乱暴な解決案:
- **学習時**: 当の クラス値-属性値 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%**

数値属性の取扱い

- 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

$$\hat{\sigma}^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2$$

- 密度関数 f(x):

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

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

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

新しい日の分類

・ 新たな日 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)

P(E)を求めると: P(play=yes | E) = **20.9%**, P(play=no | E) = **79.1%**

デモ applet

- naïve Bayes は実に簡単な分類器である。しかし、結構優秀でもある。下記 applet で他の分類器と比較してみよう
  - もとの分布が正規分布ならうまくいくのは当然ではある
  - 他の手法がどのようなものであるかは、今は、知らなくてよい

http://www.cs.technion.ac.il/~rani/LocBoost/index.html

文書分類の学習

- 適用事例:
  - どのニュースが興味あるかを学習する
  - あるニュースがどのニュースグループのものかを判定できるように学習する
  - web ページをトピックで分類することを学習する

- 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

事例の表現 or 属性

- 属性 = 単語の出現位置
  - i.e. 属性 i は文書中の第 i 番目の単語位置
  - 属性値 = その位置に現れる単語
    - doc = $(a_1=w_1, a_i=w_k, \ldots, a_n=w_n)$
  - 更なる仮定: ある特定の単語がある確率は、その位置とは独立
    - ある文書 doc = $(a_1=w_1, a_i=w_k, \ldots, a_n=w_n)$ について
    - $P(a_i=w_k|v_j) = P(a_m=w_k|v_j) = P(w_k|v_j)\ \forall i,m$
    - $P(\text{doc}|v_j)=P(a_1=w_1, a_2=w_2, \ldots, a_n=w_n|v_j)$
      $= P(w_1|v_j)^{TF(w1)} P(w_2|v_j)^{TF(w2)} \ldots P(w_n|v_j)^{TF(wn)}$

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

Naïve Bayes による文書分類

- ある doc = $(a_1=w_1,\ldots, a_i=w_k, \ldots, a_n=w_n)$ につき

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i=1}^{|doc|} P(a_i \mid v_j)$$

$$= \arg\max_{v_j \in V} P(v_j) \prod_{w_k \in Voc} P(w_k \mid v_j)^{TF(w_k, doc)}$$

ただし、$TF(w_k,\text{doc})$=doc中のw_kの出現度数

- なお下記の推定値を使用; ただし、n_j=クラスv_j中の全単語出現度数, n_{kj}=クラスv_j中の単語w_k出現度数

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

アルゴリズム

procedure learn_naïve_bayes_text(*E*: 文書集合, V: クラス集合)
Voc = E に現れる全ての単語とトークン(stop word は除く)の集合
E 中の w_k と V 中の v_j すべてについて, $P(v_j)$ と $P(w_k|v_j)$ を推定する:
N_j = クラス j の文書の数
N = 文書の総数
$P(v_j) = N_j/N$
n_{kj} = クラス j の全文書中の単語 w_k の出現数
n_j = クラス j 中の単語出現数
$P(w_k|v_j) = (n_{kj}+1)/(n_j+|\text{Voc}|)$

procedure classify_naïve_bayes_text(A: 文書)
A から, Voc にない単語とトークンすべてを除去
return $\text{argmax}_{vj \in V} P(v_j) \prod_i P(a_i|v_j) = \text{argmax}_{vj \in V} P(v_j) \prod_{wk \in Voc} P(w_k|v_j)^{TF(wk,A)}$

Twenty News Groups (Joachims 1996)

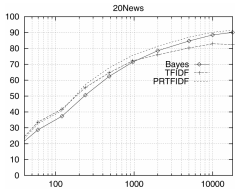
- 各グループ1000の訓練文書
- 新規の文書を、もとのnewsgroupに割振る

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.

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



精度対訓練データ数（1/3はテスト用にとりおいた）

NewsWeeder (Lang 1995)

- 目標概念 “usenet articles that I find interesting” を学習する
- ユーザはネットニュースを読むときに、興味深さの点数をつける
- 点数のついた文書を訓練例とする
- 点数を自動的につけた文書のうち上位 10% に興味深い文書が含まれる割合は、ユーザが普通に読む文書集合に含まれる割合の 3〜4倍高かった

Lang, K (1995). *NewsWeeder: Learning to Filter News*. Proceedings of the 12th International Conference on Machine Learning, 331-339, Lake Tahoe, CA.