LectureNaiveBayes

NaiveBayes Classifiers 101

Introduction

(This are some quick notes. For more details, see OURMINE.

A Bayes classifier is a simple statistical-based learning scheme.

Advantages:

- Tiny memory footprint
- Fast training, fast learning
- Simplicity
- Often works surprisingly well

Assumptions

- Learning is done best via statistical modeling
- Attributes are
  - equally important
  - statistically independent (given the class value)
  - This means that knowledge about the value of a particular attribute doesn't tell us anything about the value of another attribute (if the class is known)
- Although based on assumptions that are almost never correct, this scheme works well in practice Domingos97

Table 1. Classification accuracies and sample standard deviations, averaged over 20 random training/test splits. "Bayes" is the Bayesian classifier with discretization and "Gauss" is the Bayesian classifier with Gaussian distributions. Superscripts denote confidence levels for the difference in accuracy between the Bayesian classifier and the corresponding algorithm, using a one-tailed paired t test: 1 is 99.5%, 2 is 99%, 3 is 97.5%, 4 is 95%, 5 is 90%, and 6 is below 90%.

Data Set	Bayes	Gauss	C4.5	PEBLS	CN2	Def.
Audiology	73.0±6.1	73.0±6.1^{6}	72.5±5.8^{6}	75.8±5.4^{3}	71.0±5.1^{5}	21.3
Annealing	95.3±1.2	84.3±3.8^{1}	90.5±2.2^{1}	98.8±0.8^{1}	81.2±5.4^{1}	76.4
Breast cancer	71.6±4.7	71.3±4.3^{6}	70.1±6.8^{5}	65.6±4.7^{1}	67.9±7.1^{1}	67.6
Credit	84.5±1.8	78.9±2.5^{1}	85.9±2.1^{3}	82.2±1.9^{1}	82.0±2.2^{1}	57.4
Chess endgames	88.0±1.4	88.0±1.4^{6}	99.2±0.1^{1}	96.9±0.7^{1}	98.1±1.0^{1}	52.0
Diabetes	74.5±2.4	75.2±2.1^{6}	73.5±3.4^{5}	71.1±2.4^{1}	73.8±2.7^{6}	66.0
Echocardiogram	69.1±5.4	73.4±4.9^{1}	64.7±6.3^{1}	61.7±6.4^{1}	68.2±7.2^{6}	67.8
Glass	61.9±6.2	50.6±8.2^{1}	63.9±8.7^{6}	62.0±7.4^{6}	63.8±5.5^{6}	31.7
Heart disease	81.9±3.4	84.1±2.8^{1}	77.5±4.3^{1}	78.9±4.0^{1}	79.7±2.9^{3}	55.0
Hepatitis	85.3±3.7	85.2±4.0^{6}	79.2±4.3^{1}	79.0±5.1^{1}	80.3±4.2^{1}	78.1
Horse colic	80.7±3.7	79.3±3.7^{1}	85.1±3.8^{1}	75.7±5.0^{1}	82.5±4.2^{2}	63.6
Hypothyroid	97.5±0.3	97.9±0.4^{1}	99.1±0.2^{1}	95.9±0.7^{1}	98.8±0.4^{1}	95.3
Iris	93.2±3.5	93.9±1.9^{6}	92.6±2.7^{6}	93.5±3.0^{6}	93.3±3.6^{6}	26.5
Labor	91.3±4.9	88.7±10.6^{6}	78.1±7.9^{1}	89.7±5.0^{6}	82.1±6.9^{1}	65.0
Lung cancer	46.8±13.3	46.8±13.3^{6}	40.9±16.3^{5}	42.3±17.3^{6}	38.6±13.5^{3}	26.8
Liver disease	63.0±3.3	54.8±5.5^{1}	65.9±4.4^{1}	61.3±4.3^{6}	65.0±3.8^{3}	58.1
LED	62.9±6.5	62.9±6.5^{6}	61.2±8.4^{6}	55.3±6.1^{1}	58.6±8.1^{2}	8.0
Lymphography	81.6±5.9	81.1±4.8^{6}	75.0±4.2^{1}	82.9±5.6^{6}	78.8±4.9^{3}	57.3
Post-operative	64.7±6.8	67.2±5.0^{3}	70.0±5.2^{1}	59.2±8.0^{2}	60.8±8.2^{4}	71.2
Promoters	87.9±7.0	87.9±7.0^{6}	74.3±7.8^{1}	91.7±5.9^{3}	75.9±8.8^{1}	43.1
Primary tumor	44.2±5.5	44.2±5.5^{6}	35.9±5.8^{1}	30.9±4.7^{1}	39.8±5.2^{1}	24.6
Solar flare	68.5±3.0	68.2±3.7^{6}	70.6±2.9^{1}	67.6±3.5^{6}	70.4±3.0^{2}	25.2
Sonar	69.4±7.6	63.0±8.3^{1}	69.1±7.4^{6}	73.8±7.4^{1}	66.2±7.5^{5}	50.8
Soybean	100.0±0.0	100.0±0.0^{6}	95.0±9.0^{3}	100.0±0.0^{6}	96.9±5.9^{3}	30.0
Splice junctions	95.4±0.6	95.4±0.6^{6}	93.4±0.8^{1}	94.3±0.5^{1}	81.5±5.5^{1}	52.4
Voting records	91.2±1.7	91.2±1.7^{6}	96.3±1.3^{1}	94.9±1.2^{1}	95.8±1.6^{1}	60.5
Wine	96.4±2.2	97.8±1.2^{3}	92.4±5.6^{1}	97.2±1.8^{6}	90.8±4.7^{1}	36.4
Zoology	94.4±4.1	94.1±3.8^{6}	89.6±4.7^{1}	94.6±4.3^{6}	90.6±5.0^{1}	39.4

It has some drawbacks: it can offer conclusions put it is poor at explaining how those conclusions were reached. But that is something we'll get back to below.

Example

```
weather.symbolic.arff

outlook   temperature  humidity   windy   play
-------   -----------  --------   -----   ----
rainy     cool         normal     TRUE    no
rainy     mild         high       TRUE    no
sunny     hot          high       FALSE   no
sunny     hot          high       TRUE    no
sunny     mild         high       FALSE   no
overcast  cool         normal     TRUE    yes
overcast  hot          high       FALSE   yes
overcast  hot          normal     FALSE   yes
overcast  mild         high       TRUE    yes
rainy     cool         normal     FALSE   yes
rainy     mild         high       FALSE   yes
rainy     mild         normal     FALSE   yes
sunny     cool         normal     FALSE   yes
sunny     mild         normal     TRUE    yes
```

This data can be summarized as follows:

```
           Outlook            Temperature           Humidity
====================   =================   =================
          Yes    No             Yes   No             Yes    No
Sunny       2     3     Hot      2     2    High      3     4
Overcast    4     0     Mild     4     2    Normal    6     1
Rainy       3     2     Cool     3     1
           -----------          ---------           ----------
Sunny     2/9   3/5     Hot    2/9   2/5    High     3/9   4/5
Overcast  4/9   0/5     Mild   4/9   2/5    Normal   6/9   1/5
Rainy     3/9   2/5     Cool   3/9   1/5

            Windy         Play
=================     ========
      Yes     No      Yes   No
False 6       2       9     5
True  3       3
      ----------    ----------
False  6/9    2/5    9/14  5/14
True   3/9    3/5
```

So, what happens on a new day:

```
Outlook       Temp.          Humidity    Windy         Play
Sunny         Cool           High        True          ?%%
```

First find the likelihood of the two classes

- For "yes" = 2/9 * 3/9 * 3/9 * 3/9 * 9/14 = 0.0053
- For "no" = 3/5 * 1/5 * 4/5 * 3/5 * 5/14 = 0.0206
- Conversion into a probability by normalization:
  - P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205
  - P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

So, we aren't playing golf today.

Bayes' rule

More generally, the above is just an application of Bayes' Theorem.

- Probability of event H given evidence E:

$$Pr(H \mid E) = \frac{Pr(E \mid H) * Pr(H)}{Pr(E)}$$

- A priori probability of H= Pr(H)
  - Probability of event before evidence has been seen
- A posteriori probability of H= Pr[H|E]
  - Probability of event after evidence has been seen
- Classification learning: what's the probability of the class given an instance?
  - Evidence E = instance
  - Event H = class value for instance
- Naive Bayes assumption: evidence can be split into independent parts (i.e. attributes of instance!

$$Pr(H \mid E) = \frac{Pr(E1 \mid H) * Pr(E2 \mid H) * .... * Pr(En \mid H) * Pr(H)}{Pr(E)}$$

- We used this above. Here's our evidence:

```
Outlook       Temp.          Humidity    Windy         Play
Sunny         Cool           High        True          ?
```

- Here's the probability for "yes":

$$\begin{aligned} Pr(\text{yes} \mid E) &= Pr(\text{Outlook} = \text{Sunny} \mid \text{yes}) * \\ &\quad Pr(\text{Temperature} = \text{Cool} \mid \text{yes}) * \\ &\quad Pr(\text{Humidity} = \text{High} \mid \text{yes}) * Pr(\text{yes}) \\ &\quad Pr(\text{Windy} = \text{True} \mid \text{yes}) * Pr(\text{yes}) / Pr(E) \\ &= (2/9 * 3/9 * 3/9 * 3/9) * 9/14) / Pr(E) \end{aligned}$$

Return the classification with highest probability

- Probability of the evidence Pr(E)
  - Constant across all possible classifications;
  - So, when comparing N classifications, it cancels out

Numerical errors

From multiplication of lots of small numbers

- Use the standard fix: don't multiply the numbers, add the logs

Missing values

Missing values are a problem for any learner. Naive Bayes' treatment of missing values is particularly elegant.

- During training: instance is not included in frequency count for attribute value-class combination
- During classification: attribute will be omitted from calculation

```
Example: Outlook    Temp.    Humidity    Windy    Play
         ?          Cool     High        True     ?%%
```

- Likelihood of "yes" = 3/9 * 3/9 * 3/9 * 9/14 = 0.0238
- Likelihood of "no" = 1/5 * 4/5 * 3/5 * 5/14 = 0.0343
- P("yes") = 0.0238 / (0.0238 + 0.0343) = 41%
- P("no") = 0.0343 / (0.0238 + 0.0343) = 59%

The "low-frequencies problem"

What if an attribute value doesn't occur with every class value (e.g. "Humidity = high" for class "yes")?

- Probability will be zero!
- Pr(Humidity = High | yes) = 0
- A posteriori probability will also be zero! Pr(yes | E) = 0 (No matter how likely the other values are!)

So use an estimators for low frequency attribute ranges

- Add a little "m" to the count for every attribute value-class combination
  - The Laplace estimator
  - Result: probabilities will never be zero!

And use an estimator for low frequency classes

- Add a little "k" to class counts

- The M-estimate

Magic numbers: m=2, k=1

And we'll return to the low frequency problem, below.

Pseudo-code

Here's the pseudo code of the the Naive Bayes classifier preferred by Yang03 (p4).

```
function train(   i) {
   Instances++
   if (++N[$Klass]==1) Klasses++
   for(i=1;i<=Attr;i++)
     if (i != Klass)
      if ($i !~ /\?/)
         symbol(i,$i,$Klass)
}
function symbol(col,value,klass) {
   Count[klass,col,value]++;
}
```

When testing, find the likelihood of each hypothetical class and return the one that is most likely.

The (K,M) variables handle low frequency cases.

```
function likelihood(l,         klass,i,inc,temp,prior,what,like) {
   like = -10000000000;    # smaller than any log
   for(klass in N) {
      prior=(N[klass]+K)/(Instances + (K*Klasses));
      temp= log(prior)
      for(i=1;i<=Attr;i++) {
         if (i != Klass)
            if ( $i !~ /\?/ )
                temp += log((Count[klass,i,$i]+M*prior)/(N[klass]+M))
      }
      l[klass]= temp
      if ( temp >= like ) {like = temp; what=klass}
   }
   return what
}
```

Handling Numerics

The above code assumes that the attributes are discrete. If you have numeric attributes then either discretize the values (sort, group into sets of size 30), or use a Gaussian approximation (usually, discretization beats Gaussians).

The probability density function for the normal (Gaussian) distribution is defined by the mean and standardDev (standard deviation)

Given:

- n: the number of values;
- sum: the sum of the values; i.e. sum = sum + value;
- sumSq: the sum of the square of the values; i.e. sumSq = sumSq + value*value

Then:

```
    function mean(sum,n)  {
        return sum/n
    }
    function standardDeviation(sumSq,sum,n)  {
        return sqrt((sumSq-((sum*sum)/n))/(n-1))
    }
    function gaussianPdf(mean,standardDev,x) {
       pi= 1068966896 / 340262731; #: good to 17 decimal places
       return 1/(standardDev*sqrt(2*pi)) ^
                   (-1*(x-mean)^2/(2*standardDev*standardDev))
    }
```

For example:

```
outlook  temperature humidity windy play
-------  ----------- -------- ----- ---
sunny    85          85       FALSE no
sunny    80          90       TRUE  no
overcast 83          86       FALSE yes
rainy    70          96       FALSE yes
rainy    68          80       FALSE yes
rainy    65          70       TRUE  no
overcast 64          65       TRUE  yes
sunny    72          95       FALSE no
sunny    69          70       FALSE yes
rainy    75          80       FALSE yes
sunny    75          70       TRUE  yes
overcast 72          90       TRUE  yes
overcast 81          75       FALSE yes
rainy    71          91       TRUE  no
```

This generates the following statistics:

```
             Outlook          Temperature             Humidity
====================    =================      ================
          Yes     No            Yes     No           Yes      No
Sunny       2      3             83      85            86      85
Overcast    4      0             70      80            96      90
Rainy       3      2             68      65            80      70
           -----------           ----------           ----------
Sunny     2/9    3/5    mean     73      74.6  mean    79.1    86.2
Overcast  4/9    0/5    std dev   6.2     7.9  std dev 10.2     9.7
Rainy     3/9    2/5

              Windy            Play
===================     ===========
           Yes   No      Yes      No
False       6     2       9        5
True        3     3
            -------     ----------
False     6/9   2/5     9/14  5/14
True      3/9   3/5
```

Example density value:

- f(temperature=66|yes)= gaussianPdf(73,6.2,66) =0.0340
- Classifying a new day:

```
Outlook     Temp.     Humidity     Windy     Play
Sunny       66        90           true      ?%%
```

- Likelihood of "yes" = 2/9 * 0.0340 * 0.0221 * 3/9 * 9/14 = 0.000036
- Likelihood of "no" = 3/5 * 0.0291 * 0.0380 * 3/5 * 5/14 = 0.000136
  - P("yes") = 0.000036 / (0.000036 + 0. 000136) = 20.9%
  - P("no") = 0. 000136 / (0.000036 + 0. 000136) = 79.1%

Note: missing values during training: not included in calculation of mean and standard deviation

BTW, an alternative to the above is apply some discretization policy to the data; e.g. Yang03. Such discretization is good practice since it can dramatically improve the performance of a Naive Bayes classifier (see Dougherty95.

Not so "Naive" Bayes

Why does Naive Bayes work so well? Domingos97 offer one analysis:

- They offer one example with three attributes where the performance where a "Naive" and a "optimal" Bayes perform nearly the same.
- They generalized that to conclude that "Naive" Bayes is only really Naive in a vanishingly small number of cases.

Their three attribute example is given below. For the generalized case, see Domingos97.

Consider a Boolean concept, described by three attributes A, B and C .

Assume that the two classes, denoted by + and - are equiprobable

```
P(+) = P(-) = 1/2
```

Let A and C be independent, and let A = B (i.e., A and B are completely dependent). Therefore B should be ignored, and the optimal classification procedure for a test instance is to assign it to (i) class + if

```
P(A|+) * P(C|+) -  P(A|-) * P(C|-) > 0,
```

and (ii) to class (if the inequality has the opposite sign), and (iii) to an arbitrary class if the two sides are equal.

Note that the Bayesian classifier will take B into account as if it was independent from A, and this will be equivalent to counting A twice. Thus, the Bayesian classifier will assign the instance to class + if

```
P(A|+)^2 *  P(C|+) -  P(A|-)^2  * P(C|-) > 0,
```

and to - otherwise.

Applying Bayes' theorem, P(A|+) can be re-expressed as

```
P(A) * P(+|A)/P(+)
```

and similarly for the other probabilities.

Since P(+) = P(-), after canceling like terms this leads to the equivalent expressions

```
P(+|A) * P(+|C ) - P(-|A) * P(-|C ) > 0
```

for the optimal decision, and

```
P(+|A)^2 *  P(+|C ) - P(-|A)^2 * P(-|C) > 0
```

for the Bayesian classifier. Let

```
P(+|A) = p
P(+|C) = q.
```

Then class + should be selected when

```
pq - (1 - p)*(1 - q) > 0
```

which is equivalent to

```
q > 1 - p   [Optimal Bayes]
```

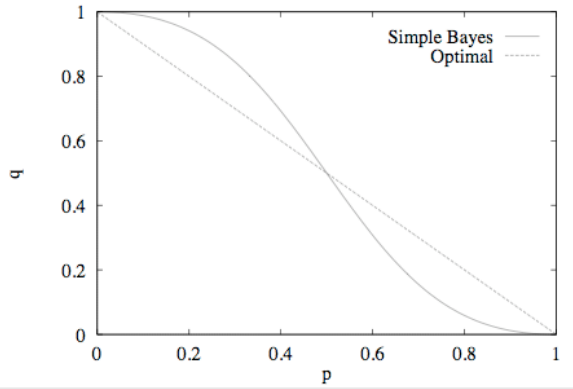
With the Bayesian classifier, it will be selected when

```
p^2 * q  - (1 - p)^2 *  (1 - q) > 0
```

which is equivalent to

```
q >  (1 - p)^2 *  p^2 +(1 - p)^2    [Simple  Bayes]
```

The two curves are shown in following figure. The remarkable fact is that, even though the independence assumption is decisively violated because B = A, the Bayesian classifier disagrees with the optimal procedure only in the two narrow regions that are above one of the curves and below the other; everywhere else it performs the correct classification.



Thus, for all problems where (p, q) does not fall in those two small regions, the Bayesian classifier is effectively optimal.