Naïve Bayes Classifiers

William W. Cohen
TREE PRUNING
Another view of a decision tree

- Sepal_length < 5.7
- Sepal_width > 2.8
Another view of a decision tree

- Sepal_length > 5.7
- Sepal_width > 2.8
- length > 4.6
- width > 3.1
- length > 5.1

Diagram showing data points in a scatter plot with decision tree rules for classification.
Another view of a decision tree

- Sepal_length > 5.7
  - Sepal_width > 2.8
    - width > 3.1
      - length > 5.1
        - Y
      - N
    - N
  - N
- Y

---

- Sepal_length > 5.7
  - Sepal_width > 2.8
    - width > 3.1
      - length > 5.1
        - Y
      - N
    - N
  - N
Another view of a decision tree
Another view of a decision tree
britain :- a1 ~ gordon (34/3).
britain :- a1 ~ england (16/0).
britain :- a1 ~ british__2 (12/5).
britain :- a1 ~ technology (10/4).
britain :- a1 ~ jobs__3 (3/0).

training: 20.49%
test: 39.43%

britain :- a1 ~ britain__2, a1 ~ an__4 (23/0).
britain :- a1 ~ british, a1 ~ more__3, a1 ~ are__4 (18/0).
britain :- a1 ~ brown, a1 ~ already (10/0).
britain :- a1 ~ england, a1 ~ have__2 (13/0).
britain :- a1 ~ london__2, a1 ~ their (7/0).
britain :- a1 ~ designed, a1 ~ it__6 (7/0).
britain :- a1 ~ british, a1 ~ government__2, a1 ~ britain (6/0).
britain :- a1 ~ tory (3/0).
britain :- a1 ~ '6', a1 ~ migration (3/0).
britain :- a1 ~ paying, a1 ~ will__2 (3/0).
britain :- a1 ~ printing (2/0).
britain :- a1 ~ darkness, a1 ~ energy (1/0).
britain :- a1 ~ fear, a1 ~ almost, a1 ~ vaccines (1/0).
britain :- a1 ~ '4__2', a1 ~ mediadirectory (1/0).

training: 5.64%
test: 45.08%
CLASSIFICATION WITH BAYES RULE AND THE JOINT DISTRIBUTION
Terminology

• $X_1 \ldots X_n$ are features
• $Y$ is the class label
• A joint distribution $P(X_1, \ldots, X_n, Y)$ assigns a probability to every combination of values for $<x_1, \ldots, x_n, y>$ for $X_1, \ldots, X_n, Y$.
• A combination of feature values $<x_1, \ldots, x_n>$ is an instance
• An instance plus a class label $y$ is an example.
CLASSIFICATION WITH BAYES RULE AND THE JOINT DISTRIBUTION

\[ P(A | B) = \frac{P(B | A)P(A)}{P(B)} \]

\( A = \text{“Joe’s cheating”} \)
\( B = \text{“4 critical hits”} \)

\( Y: \text{“this patient has cancer”, “this email is spam”, “this transaction is fraud”,…} \)
\( X’s: \text{measurements, observations, … of “this patient”, “this email”, …} \)

\[ P(Y | X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n | Y)P(Y)}{P(X_1,\ldots,X_n)} = \frac{P(X_1,\ldots,X_n | Y)P(Y)}{\text{const}} \]
Joint distributions are powerful!

<table>
<thead>
<tr>
<th>D1</th>
<th>D2</th>
<th>Roll 1</th>
<th>Roll 2</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>L</td>
<td>1</td>
<td>1</td>
<td>1/3 * 1/3 * 1/2 * 1/2</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>1</td>
<td>2</td>
<td>1/3 * 1/3 * 1/2 * 1/10</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
<td>1</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>F</td>
<td>L</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>F</td>
<td>L</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>F</td>
<td>L</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>H</td>
<td>L</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

Comment

A joint probability table shows $P(X_1=x_1$ and $\ldots$ and $X_n=x_n)$ for every possible combination of values $x_1,x_2,\ldots,x_n$

With this you can compute any $P(A)$ where $A$ is any boolean combination of the primitive events ($X_i=X_n$), e.g.

- $P(\text{doubles})$
- $P(\text{seven or eleven})$
- $P(\text{total is higher than 5})$
- $\ldots$
Get some data

% which die was used first and second
dice1 = randi(3,[n,1]);
dice2 = randi(3,[n,1]);

% did the 'loading' happen for die 1 and die 2
load1 = randi(2,[n,1]);
load2 = randi(2,[n,1]);

% simulate rolling the dice…
r1 = roll(dice1,load1,randi(5,[n,1]),randi(6,[n,1]));
r2 = roll(dice2,load2,randi(5,[n,1]),randi(6,[n,1]));

% append the column vectors
D = [dice1,dice2,r1,r2];
Get some data

```matlab
>> D(1:10,:)
ans =
    1     1     6     4
    2     1     1     3
    2     3     1     1
    3     1     1     2
    ...

>> [X,I] = sort(4*D(:,1) + D(:,2));
>> S=D(I,:);
>> imagesc(S);
```
Get some data

>> D(1:10,:)  

ans =

    1     1     6     4
    2     1     1     3
    2     3     1     1
    3     1     1     2
    ...

>> [X,I] = sort(4*D(:,1) + D(:,2));  
>> S=D(I,:);  
>> imagesc(S);  
>> D34 = D(:,3:4);  
>> hist3(D34,[6,6])
Estimate a joint density

```matlab
>> [H,C] = hist3(D34,[6,6]);
>> H

>> H

H =

60  35  24  29  30  60
42  16  14  19  14  22
27  19  15  10  17  45
44  19  17  18  20  29
31  11  12   9  22  40
51  26  44  37  17  55

>> P = H/1000

P =

0.0600  0.0350  0.0240  0.0290  0.0300  0.0600
0.0420  0.0160  0.0140  0.0190  0.0140  0.0220
0.0270  0.0190  0.0150  0.0100  0.0170  0.0450
0.0440  0.0190  0.0170  0.0180  0.0200  0.0290
0.0310  0.0110  0.0120  0.0090  0.0220  0.0400
0.0510  0.0260  0.0440  0.0370  0.0170  0.0550
```
Inference with the joint

What is $P(\text{both die fair} \mid \text{roll} > 10)$?

$P(\text{both die fair and roll} > 10) / P(\text{roll} > 10)$

```matlab
>> sum((D(:,1)==1) & (D(:,2)==1) & (D(:,3)+D(:,4) >= 10))
ans =
  9

>> sum((D(:,3)+D(:,4) >= 10))
ans =
 112
>> 9/112
ans =
0.0804
Another joint

```
>> D = [randn(1000,2)+1.5; randn(1000,2)-1.5];
>> hist3(D,[20 20]);

>> sum( (D(:,1)>0) & (D(:,2)<0) )/2000
ans = 0.0570
```
Another joint

```matlab
>> D = [randn(1000,2)+1.5; randn(1000,2)-1.5];
>> imagesc(H);
>> imagesc(D);
```
Another joint

```matlab
[H,C]=hist3(D,[20 20]);
[I,J]=meshgrid(C{1},C{2});
hold off
plot3(D(:,1),D(:,2),zeros(2000,1)-0.002,'r*');
hold on
surf(I,J,H/2000);
alpha(0.2);

>> sum( (D(:,1)>0) & (D(:,2)<0) )/2000
ans = 0.0570
```
NAÏVE BAYES
Problem with the joint

• The joint density table is too big to estimate easily
  – With $n$ features there are $2^n$ probabilities to estimate.
  – You need lots of data or a very small $n$
Conditional independence

• Two variables A, B are *independent* if

\[ P(A \land B) = P(A) \times P(B) \]
\[ \forall a, b : P(A = a \land B = b) = P(A = a) \times P(B = b) \]

• Two variables A, B are *conditionally independent given* C if

\[ P(A,B | C) = P(A | C) \times P(B | C) \]
\[ \forall a,b,c : P(A = a \land B = b | C = c) = P(A = a | C = c) \times P(B = b | C = c) \]
Estimating conditional probabilities

| A | C | P(A | C) |
|---|---|--------|
| 0 | 0 | 0.2    |
| 0 | 1 | 0.5    |
| 1 | 0 | 0.8    |
| 1 | 1 | 0.5    |

| A | B | C | P(A,B | C) |
|---|---|---|--------|
| 0 | 0 | 0 | ...    |
| 0 | 0 | 1 | ...    |
| 0 | 1 | 0 | ...    |
| 0 | 1 | 1 |        |
| 1 | 0 | 0 |        |
| 1 | 0 | 1 |        |
| 1 | 1 | 0 |        |
| 1 | 1 | 1 |        |

| B | C | P(B | C) |
|---|---|--------|
| 0 | 0 | 0.1    |
| 0 | 1 | 0.9    |
| 1 | 0 | 0.9    |
| 1 | 1 | 0.1    |
Estimating conditional probabilities

| A | C | P(A | C) |
|---|---|-------|
| 0 | 0 | 0.2   |
| 0 | 1 | 0.5   |
| 1 | 0 | 0.8   |
| 1 | 1 |       |

| B | C | P(B | C) |
|---|---|-------|
| 0 | 0 | 0.1   |
| 0 | 1 | 0.9   |
| 1 | 0 | 0.9   |
| 1 | 1 | 0.1   |

| D | C | P(D | C) |
|---|---|-------|
| 0 | 0 | 0.1   |
| 0 | 1 | 0.1   |
| 1 | 0 | 0.9   |
| 1 | 1 | 0.1   |

| A | B | D | C | P(A,B,D | C) |
|---|---|---|---|-------|
| 0 | 0 | 0 | 0 |       |
| 0 | 0 | 1 | 0 |       |
| 0 | 1 | 0 | 0 |       |
| 0 | 1 | 1 | 0 |       |
| 1 | 0 | 0 | 0 |       |
| 1 | 0 | 1 | 0 |       |
| 1 | 1 | 0 | 0 |       |
| 1 | 1 | 1 | 0 |       |
| 0 | 0 | 0 | 1 |       |
| 0 | 0 | 1 | 0 |       |
| ... | ... | ... | ... | ... |
Reducing the number of parameters to estimate

\[ P(Y \mid X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n \mid Y)P(Y)}{P(X_1, \ldots, X_n)} \]

\[ \forall x, y : P(Y = y \mid X_1, \ldots, X_n = x) = \frac{P(X_1, \ldots, X_n = x \mid Y)P(Y = y)}{P(X_1, \ldots, X_n = x)} \]

Pr(\(X=x\mid Y=y\)) is a huge table: if \(X\)'s and \(Y\) are binary it has …. probability estimates
Reducing the number of parameters to estimate

\[ P(Y \mid X_1,\ldots,X_n) = \frac{P(X_1,\ldots,X_n \mid Y)P(Y)}{P(X_1,\ldots,X_n)} \]

To make this tractable we “naively” assume conditional independence of the features given the class: ie

\[ P(X_1\ldots,X_n \mid Y) = P(X_1 \mid Y) \cdot P(X_2 \mid Y) \cdots P(X_n \mid Y) \]

Now: I only need to estimate … parameters:

\[ P(X_1 \mid Y), P(X_2 \mid Y), \ldots, P(X_n \mid Y), P(Y) \]
About Naïve Bayes

• This is a terrible density estimator:
  – $\Pr(X=x \mid Y=y)$ tends to extreme values

• In many cases, using this density estimator for classification often gives amazingly competitive results!
  – See: (Domingos and Pazzani, 1997)

• Naïve Bayes is fast and easy to implement

• See: “Naïve Bayes at 40” (Lewis, 1998)
Breaking it down: the math

I want:

$$\arg\max_y P(Y = y \mid X_1, \ldots, X_n = x) = \frac{P(X_1, \ldots, X_n = x \mid Y = y)P(Y = y)}{P(X_1, \ldots, X_n = x)}$$

$$= \arg\max_y P(Y = y \mid X_1, \ldots, X_n = x) = P(X_1, \ldots, X_n = x \mid Y = y)P(Y = y)$$

why? \(P(X=x)\) doesn’t affect order of the y’s

$$= \arg\max_y P(X_1 = x_1 \mid Y = y) \ast \ldots \ast P(X_n = x_n \mid Y = y)P(Y = y)$$

why? conditional independence assumption
Breaking it down: the code

• From the data D, estimate class priors.
  – For each possible value of Y, estimate $Pr(Y=y_1), Pr(Y=y_2), \ldots, Pr(Y=y_k)$
  – usually a MLE is fine: $p(k)=Pr(Y=y_i) = \#D(Y=y_i)/|D|$

• From the data, estimate the conditional probabilities
  – If every $X_i$ has values $x_{i1}, \ldots, x_{ik}$
    • for each $y_i$ and each $X_i$ estimate $q(i,j,k)=Pr(X_i=x_{ij} | Y=y_i)$
      eg with $q(i,j,k) = \#D(X_i=true and Y=y_i)/\#D(Y=y_i)$
    • or better, using a MAP estimate:
      – $q(i,j,k) = (\#D(X_i=x_{ij} and Y=y_i) + q0) / (\#D(Y=y_i) + 1)$

$q0$ is probably a uniform estimate
Breaking it down: the code

• Given a new instance with $X_i = x_{ij}$ compute the following for each possible value $y$ of $Y$

$$
= \arg \max_{y_k} P(X_1 = x_{j,1} \mid Y = y) \times \ldots \times P(X_n = x_{j,n} \mid Y = y_k) P(Y = y_k)
$$

$$
= \arg \max_{y_k} \prod_i P(X_i = x_{ji} \mid Y = y_k) P(Y = y_k)
$$

$$
= \arg \max_k \sum_{i=1}^n \log q(i, j, k) + \log p(k)
$$
Comments and tricks

• Smoothing is important for naïve Bayes
  – If estimate for \( \text{Pr}(X=x | Y=y) \) is zero that’s a bad thing.
  – It’s especially important when you’re working with text
Comments and tricks

- Smooth the conditional probabilities
  - If every $X_i$ has values $x_{i1}, \ldots, x_{ik}$
    - for each $y_i$ and each $X_i$ MLE estimate is:
      $$q(i,j,k) = \frac{\#D(X_i = x_{ij} \text{ and } Y = y_i)}{\#D(Y = y_i)}$$
    - better is a MAP estimate:
      $$q(i,j,k) = \frac{\#D(X_i = x_{ij} \text{ and } Y = y_i) + q0}{\#D(Y = y_i) + 1}$$
- For text
  - we usually model $\Pr(X_i \mid Y)$ as a multinomial with many possible outcomes (one per observed word), so $q0 = 1/|V|$.
  - we also combine counts for $X_1, X_2, \ldots$ so we just estimate one conditional distribution over words (per class $y$).
  - i.e., an $m$-word message contains $m$ rolls of one $|V|$-sided die
Comments and tricks

- $X_1 =$ scientists
- $X_2 =$ successfully
- $X_3 =$ teach
- $\ldots$
- $X_{28} =$ to
- $X_{29} =$ humans
- $Y =$ onion
- $\ldots$

**Estimate:**

$Pr(X_1=\text{aardvark} \mid Y=\text{onion})$

$\ldots$

$Pr(X_1=\text{zymurgy} \mid Y=\text{onion})$

$Pr(X_2=\text{aardvark} \mid Y=\text{onion})$

$\ldots$
Comments and tricks

- \( X = \) scientists
- \( X = \) successfully
- \( X = \) teach
- \( X = \) to
- \( X = \) humans
- \( Y = \) onion
- \( Y = \) economist

Estimate:

\[
\Pr(X=\text{aardvark} \mid Y=\text{onion}) \\
\ldots \\
\Pr(X=\text{zymurgy} \mid Y=\text{onion}) \\
\Pr(X=\text{aardvark} \mid Y=\text{economist}) \\
\ldots
\]

\[
\text{Doc} = 29 \text{ draws of } X, \text{ not one draw each of } X_1, \ldots, X_{29}
\]
Comments and tricks

• For text
  – we usually model $\Pr(X_i \mid Y)$ as a multinomial with many possible outcomes (one per observed word), so $q_0=1/\mid V \mid$
  – we also combine counts for $X_1, X_2, \ldots$ so we just estimate one conditional distribution over words (per class $y$)
  – i.e., an $m$-word message contains $m$ rolls of one $|V|$-sided die

$$
\Pr(Y = Y \mid X = x) = \frac{1}{Z} \prod_{i=1}^{m} \Pr(X = w_i \mid Y = y) \Pr(Y = y)
$$

where $w_i$ is $i$-th word in $x$
Comments and tricks

• What about continuous data?
  – If every $X_i$ is continuous
    • For each possible value of $Y$, $y_1$, $y_2$, … find the set of $x_i$’s that cooccur with $y_k$ and compute their mean $\mu_k$ and standard deviation $\sigma_k$
    • Estimate $\Pr(X_i \mid Y=y_k)$ using a Gaussian with $N(\mu_k, \sigma_k)$
    • Note: co-variances are assumed to be zero.
VISUALIZING NAÏVE BAYES
%% Import the IRIS data
load fisheriris;
X = meas;
pos = strcmp(species,'setosa');
Y = 2*pos - 1;

%% Visualize the data
imagesc([X,Y]);
title('Iris data');
%% Visualize by scatter plotting the first two dimensions
figure;
s = scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
s = scatter(X(Y>0,1),X(Y>0,2),'bo');
title('Iris data');
Compute the mean and SD of each class

\[
\text{PosMean} = \text{mean}(X(Y>0,:)); \\
\text{PosSD} = \text{std}(X(Y>0,:)); \\
\text{NegMean} = \text{mean}(X(Y<0,:)); \\
\text{NegSD} = \text{std}(X(Y<0,:));
\]

Compute the NB probabilities for each class for each grid element

\[
[G1,G2]=\text{meshgrid}(3:0.1:8, 2:0.1:5); \\
Z1 = \text{gaussmf}(G1,[\text{PosSD}(1),\text{PosMean}(1)]); \\
Z2 = \text{gaussmf}(G2,[\text{PosSD}(2),\text{PosMean}(2)]); \\
Z = Z1 .* Z2;
\]

\[
V1 = \text{gaussmf}(G1,[\text{NegSD}(1),\text{NegMean}(1)]); \\
V2 = \text{gaussmf}(G2,[\text{NegSD}(2),\text{NegMean}(2)]); \\
V = V1 .* V2;
\]
% Add them to the scatter plot
figure;
scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
scatter(X(Y>0,1),X(Y>0,2),'bo');
contour(G1,G2,Z);
contour(G1,G2,V);
% Add them to the scatter plot
figure;
scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
scatter(X(Y>0,1),X(Y>0,2),'bo');
contour(G1,G2,Z);
contour(G1,G2,V);
%% Now plot the difference of the probabilities
figure;
scatter(X(Y<0,1),X(Y<0,2),'r*');
hold on;
scatter(X(Y>0,1),X(Y>0,2),'bo');
contour(G1,G2,Z-V);
NAÏVE BAYES IS LINEAR
Recall: density estimation vs classification

Regressor
- Prediction of real-valued output
- Input Attributes

Density Estimator
- Probability
- Input Attributes

Classifier
- Prediction of categorical output or class
- One of a few discrete values
- Input Attributes

Regressor
- Prediction of real-valued output
- Input Attributes
Recall: density estimation vs classification

To classify $x$
1. Use your estimator to compute $\hat{P}(x,y_1), \ldots, \hat{P}(x,y_k)$
2. Return the class $y^*$ with the highest predicted probability

Ideally is correct with $\hat{P}(x,y^*) = \hat{P}(x,y^*)/(\hat{P}(x,y_1) + \ldots + \hat{P}(x,y_k))$
Classification vs density estimation
Question: what does the *boundary* between positive and negative look like for Naïve Bayes?
\[
\text{argmax}_y \prod_i P(X_i = x_i \mid Y = y)P(Y = y) \\
= \text{argmax}_y \sum_i \log P(X_i = x_i \mid Y = y) + \log P(Y = y) \\
= \text{argmax}_{y \in \{+1, -1\}} \sum_i \log P(x_i \mid y) + \log P(y) \quad \text{(two classes only)} \\
= \text{sign} \left( \sum_i \log P(x_i \mid y_+) - \sum_i \log P(x_i \mid y_-) + \log P(y_+) - \log P(y_-) \right) \\
= \text{sign} \left( \sum_i \log \frac{P(x_i \mid y_+)}{P(x_i \mid y_-)} + \log \frac{P(y_+)}{P(y_-)} \right) \quad \text{(rearrange terms)}
\]
\[
\arg\max_y \prod_i P(X_i = x_i \mid Y = y) P(Y = y) \\
= \text{sign} \left( \sum_i \log \frac{P(x_i \mid y_+)}{P(x_i \mid y_-)} + \log \frac{P(y_+)}{P(y_-)} \right)
\]

if \( x_i = 1 \) or 0 ....

\[
u_i = \left( \log \frac{P(x_i = 1 \mid y_+)}{P(x_i = 1 \mid y_-)} - \log \frac{P(x_i = 0 \mid y_+)}{P(x_i = 0 \mid y_-)} \right)
\]

\[
= \text{sign} \left( \sum_i x_i u_i + u_0 \right)
\]

\[
= \text{sign} \left( \sum_i x_i u_i + x_0 u_0 \right) = \text{sign}(x \cdot u)
\]

\( x_0 = 1 \) for every \( x \) (bias term)