Naive Bayes Classifier

Mitchell [6.7-6.9]

Bayes Classifier

- Assume we want to classify **x** described by attributes $[a_1,...a_n]$.
- Bayes theorem tells us to find Cj for which this is maximum:

$$P(C_j \mid \mathbf{x}) = P(\mathbf{x} \mid C_j)\, P(C_j) / P(\mathbf{x})$$

- This is expressed as

$$C = \underset{j}{\text{argmax}}\ P(\mathbf{x} \mid C_j)\, P(C_j)$$

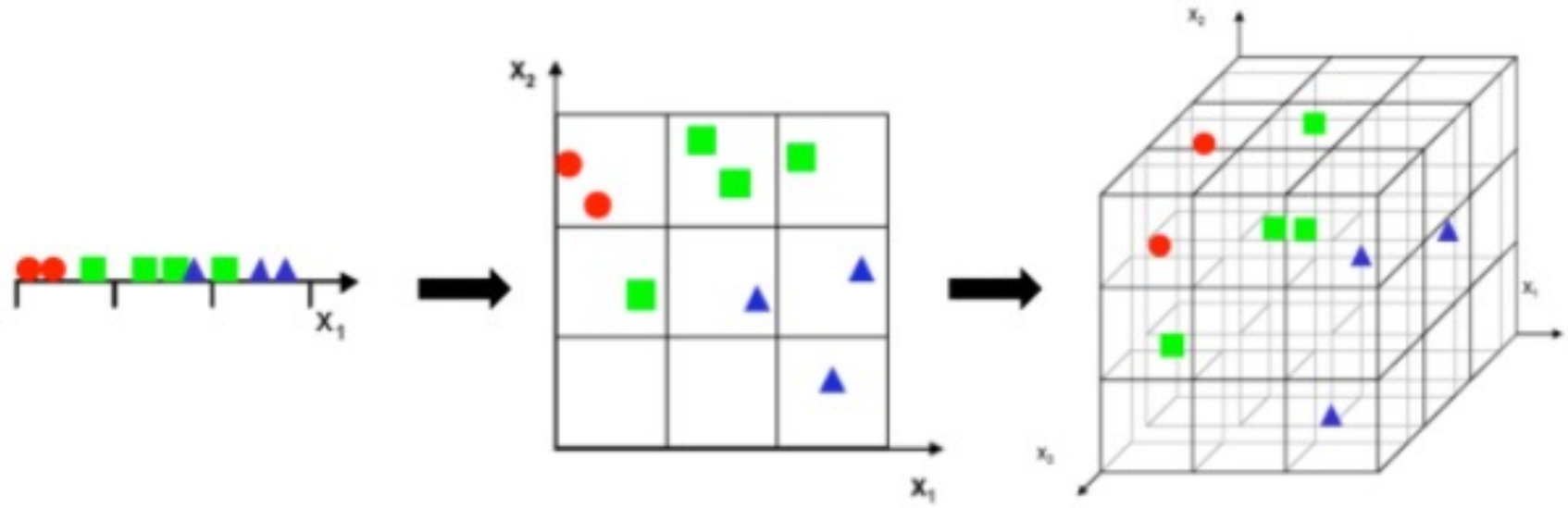
Curse of Dimensionality

- In a toy learning problem (thx to Gutierrez-Osuna), our algorithm:
  - divides the feature space uniformly into bins and
  - for each new example that we want to classify, we just need to figure out the bin the example falls into and find the predominant class in that bin as the label.

  - Consider a single feature where the input space is divided into 3 bins:

  - Noticing the overlap, we decide to add one more feature:

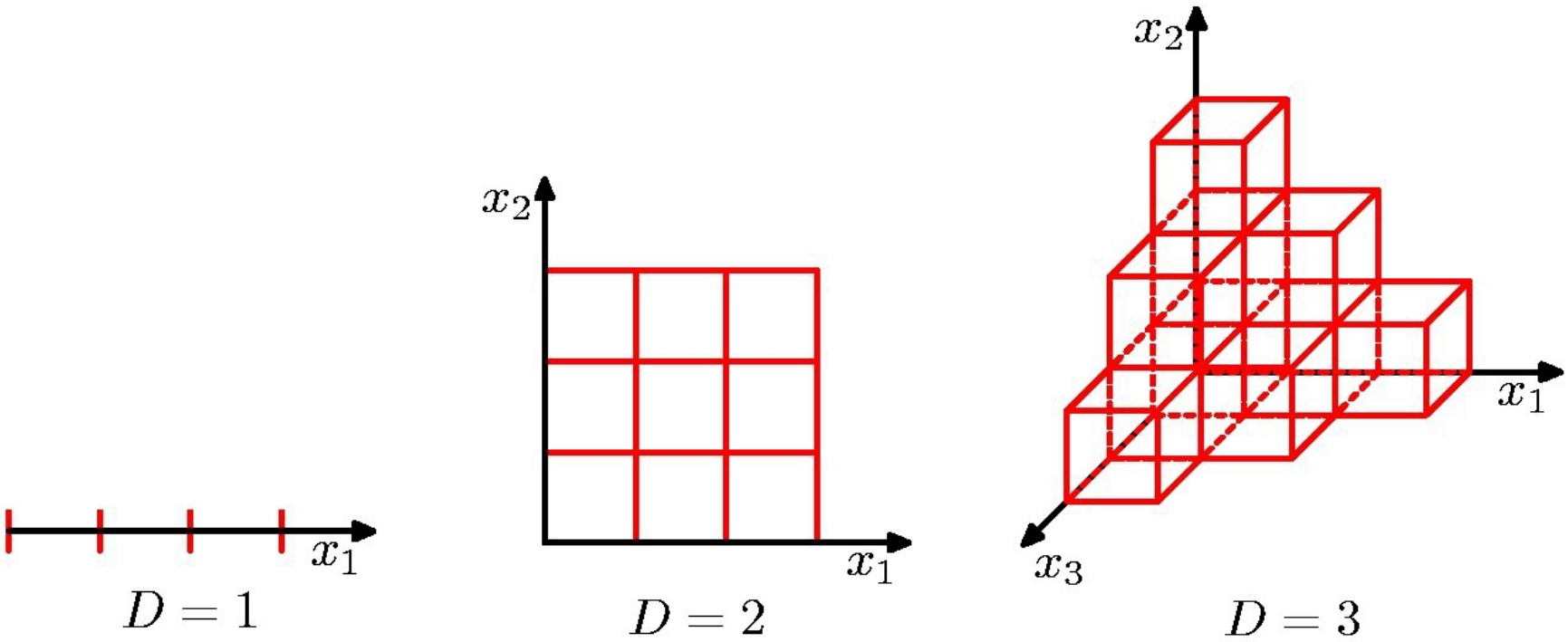
Curse of Dimensionality



Now the problem is apparent: **With increasing dimensionality, the number of bins required to cover the feature space increases exponentially and there won't be enough data to populate each bin.**

Finding the predominant class in each bin is very difficult in high dimensional spaces.

Curse of Dimensionality



As dimensionality D increases, the amount of data needed increases exponentially with D. (e.g. Instead of 100 for 1D, 100^3 for 3D)

Bayes Classifier

- But it requires a lot of data to estimate (roughly $O(|A|^n)$ parameters for each class):

$$P(a_1,a_2,\ldots a_n|\ C_j)$$

- **Naïve Bayesian Approach:** We assume that the attribute values are conditionally independent given the class C_j so that

$$P(a_1,a_2,..,a_n|C_j) = \prod_i P(a_i|C_j)$$

Naive Bayes Assumption

Naïve Bayes uses assumption that the X_i are conditionally independent, given Y. E.g., $P(X_1|X_2, Y) = P(X_1|Y)$

Given this assumption, then:

$$P(X_1, X_2|Y) = P(X_1|X_2, Y)P(X_2|Y)$$

Chain rule

$$P(X_1, X_2|Y) = P(X_1|Y)P(X_2|Y)$$

Naïve Bayes Assumption

- More generally:

$$P(X_1 \ldots X_d|Y) = \prod_{i=1}^{d} P(X_i|Y)$$

Independence

X and Y are said to be independent if f P(X,Y)=P(X)P(Y)

- Since P(X,Y)= P(X | Y) P(Y) by definition, we have the equivalent definition of P(X | Y) = P(X) .
  - This says that Y does not give me any extra information to change my belief about the probability of X.

- Independence and conditional independence are important because they significantly reduce the number of parameters needed and reduce computation time.
  - Consider estimating the joint probability distribution of two random variables A and B:
    - 10^2=100 vs 10+10=20 if each have 10 possible outcomes
    - 100^2=10,000 vs 100+100=200 if each have 100 possible outcomes

Conditional Independence

X is conditionally independent of Y given *Z* if
the probability distribution governing *X* is independent of the value of *Y* **given *Z***.

$$(\forall x_i, y_j, z_k)\ P(X=x_i|Y=y_j,Z=z_k)=P(X=x_i|Z=z_k)$$

or simply: $P(X|Y,Z)=P(X|Z)$

Note that using the Bayes thm, we can also show the other way around:

$P(X,Y|Z) = P(X|Z)xP(Y|Z)$ since:

$$\|$$

$P(X|Y,Z)xP(Y|Z)$

$$\|$$

$P(X|Z)xP(Y|Z)$

Exercise

P(WearsEarring | Gender) =?= P(WearsEarring)

P(WearsEarring | WearsSkirt) =?= P(WearsEarring)

Exercise

P(WearsEarring | Gender) =?= P(WearsEarring)

- No, so the probability of wearing earing depends on gender
- I.e., gender changes the probability we assign to someone wearing an earring.

P(WearsEarring | WearsSkirt) =?= P(WearsEarring)

- No, so the probability of wearing earing is also affected by whether they are wearing a skirt (more likely)

But we can say that, once we know the gender, wearing a skirt does not give us further info to change the probability we assign to someone wearing an earring. I.e. a woman/man is as likely to wear an earing whether he/she wears a skirt or not*.

- In the case of a man, not very likely, in the case of a woman, more likely
- Thus P(WearsEarring =true | WearsSkirt = true, Gender=woman) =
  P(WearsEarring =true | Gender=woman)
- *Note, this could be an approximation, to simplify things, as long as the knowledge of wearing a skirt does not significantly change the probabilities.

Naïve Bayes Classification

Bayes rule:

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y = y_k) P(X_1 \dots X_n | Y = y_k)}{\sum_j P(Y = y_j) P(X_1 \dots X_n | Y = y_j)}$$

Assuming conditional independence among X_i's:

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y = y_k) \prod_i P(X_i | Y = y_k)}{\sum_j P(Y = y_j) \prod_i P(X_i | Y = y_j)}$$

(estimate in training)

So, to pick most probable Y for $X^{new} = (X_1, \dots, X_n)$

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

(testing)

Naive Bayes in a Nutshell

- Train Naïve Bayes (examples)

  for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \pi_k \prod_i \theta_{ijk}$$

Naive Bayes Classifier - Derivation

- **Naïve Bayesian Approach:** We assume that the attribute values are conditionally independent given the class v_j so that:

$$P(a_1,a_2,..,a_n|C_j) = \prod_i P(a_i|C_j)$$

- Use repeated applications of the definition of conditional probability (chain rule).

$$P(a_1,a_2,a_3|\ C) = P(a_3|a_1,a_2,C)\ P(a_2|a_1,C)\ P(a_1|C)$$

- If we assume that a_i are conditionally independent given C.

$$P(a_i \mid a_j, C) = P(a_i \mid C)$$

- Then we have (dropping irrelevant terms in the exact formula):

$$P(a_1,a_2,a_3|\ C) = P(a_3|C)\ P(a_2|C)\ P(a_1|C)$$

Naïve Bayes in a Nutshell

- Train Naïve Bayes (examples)

  for each* value y_k

  estimate $\pi_k \equiv P(Y = y_k)$

  for each* value x_{ij} of each attribute X_i

  estimate $\theta_{ijk} \equiv P(X_i = x_{ij} | Y = y_k)$

- Classify (X^{new})

$$Y^{new} \leftarrow \arg\max_{y_k} \; P(Y = y_k) \prod_i P(X_i^{new} | Y = y_k)$$

$$Y^{new} \leftarrow \arg\max_{y_k} \; \pi_k \prod_i \theta_{ijk}$$

From Oznur hoca's slides

How Naïve Bayes assumption Simplifies Parameter Estimation

Number of Parameters

$$\mathbf{P}(Y \mid X) = \frac{\mathbf{P}(X \mid Y)\mathbf{P}(Y)}{\mathbf{P}(X)}$$

- Suppose $X = [X_1, \ldots, X_n]$ and all X_is and Y are boolean variables.
- To estimate $\mathbf{P}(X_1, \ldots, X_n \mid Y)$ how many parameters do we need to estimate?
- How many parameters are needed to define $\mathbf{P}(Y)$

Suppose $X = (X_1, X_2)$

where X_i and Y are boolean RV's

To estimate $P(Y | X_1, X_2)$

X_1	X_2	$P(Y = 1 \mid X_1, X_2)$	$P(Y = 0 \mid X_1, X_2)$
0	0	0.1	0.9
1	0	0.24	0.76
0	1	0.54	0.46
1	1	0.23	0.77

Y	X_1	X_2	…	X_N	P(X|Y)
0	0	0	…	0	0.1
0	1	0	…	…	0.02
…	…	…	…	…	…
…	…	…	…	…	…
…	…	…	…	…	…
0	1	1	1	1	0.16

$2^n - 1$

Y	X_1	X_2	…	X_N	P(X|Y)
1	0	0	…	0	
1	1	0	…	…	
…	…	…	…	…	
…	…	…	…	…	
…	…	…	…	…	
1	1	1	1	1	

Should I compute these as well?

If n=30, (2^n-1) ~ 1 billion

Bayes vs Naïve Bayes

How many parameters to describe $P(X_1 \ldots X_n | Y)$? $P(Y)$?

- Without conditional indep assumption? $2(2^n - 1)$ and 1
- With conditional indep assumption? 2n and 1

What matters mainly is not really time or space considerations, but when we have a large number of attributes, we cannot estimate these parameters very well, because we will not have enough data.

Example with categorical variables

Example from Mitchell Chp 3.

PlayTennis: training examples

Target

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Training

First the probabilities of the different target values can easilty be estimated based on their frequencies over the 14 training examples.

This is based on the intuitive solution provided by the Maximum Likelihood Estimation. That is the probability of a class is estimated over the ratio of that class observed in the training data:

$$P(PlayTennis = yes) = 9/14 = 0.64$$
$$P(PlayTennis = no) = 5/14 = 0.36$$

Similarly, we can estimate the conditional probabilities as the ratios observed in the given class, for the particular attribute value. For example, those for $Wind = strong$ are:

$$P(Wind = strong \mid PlayTennis = yes) = 3/9 = 0.33$$
$$P(Wind = strong \mid PlayTennis = no) = 3/5 = 0.60$$

Training

Consider each feature independenlty and for each class label y_k and $x_{i,j}$ value of feature i estimate $\mathbf{P}\left(X_i = x_{i,j} \,|\, Y = y_k\right)$:

$$\mathbf{P}\left(O = \texttt{sunny} \,|\, Play = \texttt{Yes}\right) \qquad \mathbf{P}\left(O = \texttt{sunny} \,|\, Play = \texttt{No}\right)$$
$$\mathbf{P}\left(O = \texttt{overcast} \,|\, Play = \texttt{Yes}\right) \qquad \mathbf{P}\left(O = \texttt{overcast} \,|\, Play = \texttt{No}\right)$$
$$\ldots$$
$$\mathbf{P}\left(T = \texttt{hot} \,|\, Play = \texttt{Yes}\right) \qquad \mathbf{P}\left(T = \texttt{hot} \,|\, Play = \texttt{No}\right)$$
$$\ldots$$
$$\mathbf{P}\left(H = \texttt{high} \,|\, Play = \texttt{Yes}\right) \qquad \mathbf{P}\left(H = \texttt{high} \,|\, Play = \texttt{No}\right)$$
$$\ldots$$
$$\mathbf{P}\left(W = \texttt{true} \,|\, Play = \texttt{Yes}\right) \qquad \mathbf{P}\left(W = \texttt{true} \,|\, Play = \texttt{No}\right) \ldots$$

And estimate the class prior $\mathbf{P}\left(Y = y_k\right)$:

$\mathbf{P}\left(Play = \texttt{Yes}\right)$ (Note that $\mathbf{P}\left(Play = \texttt{No}\right) = 1 - \mathbf{P}\left(Play = \texttt{Yes}\right)$)

Testing

- Posterior probability for a new instance with the feature vector:
- X_{new} = (sunny, cool, high, strong)

Posterior **Likelihood** **Prior**

$$\mathbf{P}(Play \mid X) \propto \mathbf{P}(X \mid Play)\,\mathbf{P}(Play)$$

$$\mathbf{P}(Play = Y \mid X) \propto \mathbf{P}(X \mid Play = Y)\,\mathbf{P}(Play = Y)$$
$$\mathbf{P}(Play = N \mid X) \propto \mathbf{P}(X \mid Play = N)\,\mathbf{P}(Play = N)$$

Testing/Query

Outlook Temperature Humidity Wind

X = [sunny, cool, high, strong]

$\mathbf{P}(O \mid Play = Y)$, $\mathbf{P}(T \mid Play = Y)$, $\mathbf{P}(H \mid Play = Y)$, and $\mathbf{P}(W \mid Play = Y)$

$\mathbf{P}(O \mid Play = N)$, $\mathbf{P}(T \mid Play = N)$, $\mathbf{P}(H \mid Play = N)$, and $\mathbf{P}(W \mid Play = N)$

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temperature	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Humidity	Play=*Yes*	Play=*No*
High	3/9	4/5
Normal	6/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

$P(\text{Play}=Yes) = 9/14$ $P(\text{Play}=No) = 5/14$

Example

- Estimating the likelihood:

$$\mathbf{P}(X \mid Play = Y) = \mathbf{P}(O = sunny \mid Play = Y)\,\mathbf{P}(T = cool \mid Play = Y)\,\mathbf{P}(H = high \mid Play = Y)\,\mathbf{P}(W = strong, Play = Y)$$
$$= \frac{2}{9}\cdot\frac{3}{9}\cdot\frac{3}{9}\cdot\frac{3}{9} \approx 0.0082$$
$$\mathbf{P}(X \mid Play = N) = \mathbf{P}(O = sunny \mid Play = N)\,\mathbf{P}(T = cool \mid Play = N)\,\mathbf{P}(H = high \mid Play = N)\,\mathbf{P}(W = strong \mid Play = N)$$
$$= \frac{3}{5}\cdot\frac{1}{5}\cdot\frac{4}{5}\cdot\frac{3}{5} = 0.0576$$

- Estimating the posterior:

$$\mathbf{P}(Play = Y \mid X) \propto \mathbf{P}(X \mid Play = Y)\,\mathbf{P}(Play = Y) = 0.0082 * \frac{9}{14} \approx 0.0052$$

$$\mathbf{P}(Play = N \mid X) \propto \mathbf{P}(X \mid Play = N)\,\mathbf{P}(Play = N) = 0.0576 * \frac{5}{14} \approx 0.0205$$

- Class label predicted for X is then Play = No

Obtaining Normalized Probabilities

By normalizing the above quantities to sum to one we can calculate the conditional probability that the target value is no, given the observed attribute values.

$$P(\text{Play=No} \mid X) = \frac{0.0206}{0.0206+0.0053} = 0.795$$

Notice that we use equality here, taking into consideration P(X).

- What is P(X) here? Try to write it in terms of known terms.

Overview of NB

Naïve Bayes Subtleties

1. Usually features are not conditionally independent

$$P(X_1 \ldots X_n|Y) \neq \prod_i P(X_i|Y)$$

- It does not produce accurate probability estimates when its independence assumptions are violated, but it works well in many cases, as picks the correct maximum-probability class [Domingos&Pazzani, 1996].

- Typically handles noise well since it does not even focus on completely fitting the training data.

Naive Bayes Subtleties

2. What if none of the training instances with target value Y_j have attribute value a_i?

$$\hat{P}(a_i|Y_j) = 0, \text{ and...}$$

$$\hat{P}(Y_j)\prod_i \hat{P}(a_i|Y_j) = 0$$

- Naively setting zero probabilities to small number results in probabilities not summing to 1 (what if there are many attribute values).
- **Solution**: In Laplace smoothing, we assume **each attribute value** is observed in one added virtual instance and add as many virtual instances as there are **attribute values** to the denominator.

Laplace Smoothing

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

Laplace Smoothing

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

P(O=Sunny | Play = Yes) = (2+1)/(9+3) = 3/12

P(O=Overcast | Play = Yes) = (4+1)/(9+3) = 5/12

P(O=Rain | Play = Yes) = (3+1)/(9+3) = 4/12

P(O=Sunny | Play = No) = (3+1)/(5+3) = 4/8

P(O=Overcast | Play = No) = (0+1)/(5+3) = 1/8

P(O=Rain | Play = No) = (2+1)/(5+3) = 3/8

Naive Bayes Subtleties

More general solution is using the Bayesian estimate for $\hat{P}(a_i|Y_j)$

$$\hat{P}(a_i|Y_j) \leftarrow \frac{n_c + mp}{n + m}$$

where

- n is number of training examples for which $Y = Y_j$
- n_c number of training examples for which $Y = Y_j$ and $a = a_i$
- p is prior estimate for $\hat{P}(a_i|Y_j)$
- m is weight given to prior (i.e. number of "virtual" examples)

E.g. assume m=100 virtual samples in addition to n=1000 observed samples and assign them 0.1 , 0.3, and 0.6 to three attributes a_1, a_2, a_3.

In Laplace smoothing, we have m=number of attribute values (1 virtual sample per attribute value) with probability 1/m for each attribute (hence mp = 1).

Naïve Bayes subleties

3. Naive Bayes posteriors often unrealistically close to 1 or 0 ☹

   This is a common problem, not very specific to NB.

3. Naive Bayes is not affected by missing attribute values! ☺

NB Missing Values

- How can we do that?
- Based on conditional independence. Assume X_j is missing:

$$\begin{aligned}\mathbf{P}(X_1, \ldots, X_j, ..., X_n \,|\, Y) &= \mathbf{P}(X_1 \,|\, Y) \ldots \mathbf{P}(X_j \,|\, Y) \ldots \mathbf{P}(X_n \,|\, Y) \\ &= \mathbf{P}(X_1 \,|\, Y) \ldots \sum_{x_j} \mathbf{P}(X_j = x_j \,|\, Y) \ldots \mathbf{P}(X_n \,|\, Y) \\ &= \mathbf{P}(X_1 \,|\, Y) \ldots 1 \ldots \mathbf{P}(X_n \,|\, Y)\end{aligned}$$

- So we can just ignore the missing value/dimension..

NB Multi-class classification

- If we have more than one class, it naturally handles multiple classes

Practical Detail

- We are multiplying lots of small numbers. Danger of underflow!

- Underflow occurs when you perform an operation that's smaller than the smallest magnitude non-zero number.
- Solution:
  - $p1 * p2 = e^{\log(p1)+\log(p2)}$
  - Perform all computations by summing logs of probabilities rather than multiplying probabilities.

Incremental Updates

- Training is fast (linear in the number of examples, features and classes)

- If the model is going to be updated very often as new data come, you may implement it such that it allows cheap incremental updates.

Incremental Updates

- Training is fast (linear in the number of examples, features and classes)
- If the model is going to be updated very often as new data come, you may implement it such that it allows cheap incremental updates.
- For example: Store raw counts instead of probabilities
  - New example of class k:
    - For each feature update the counts based on the example feature vector
    - Update the class counts, update the number of training data
    - When need to classify compute the probabilities

Questions to think about

- Can you use Naïve Bayes for a combination of discrete and continuous features?

- How can we model just 2 of n attributes as dependent?

Naive Bayes

- Very fast, low storage requirements
- Robust to irrelevant features
- Optimal if the conditional independence assumptions hold
- A good dependable baseline for text classification

What you Should Know

- Training and using classifier based on Bayes rule
- Conditional independence
  - What it is
  - Why it is important
- Naïve Bayes
  - What it is
  - How to estimate the parameters
  - How to make predictions

Acknowledgements

- Oznur Tastan
- Tom Mitchell, Victor Lavrenko, Richard Zemel, Raquel Urtasun and Sanja Fidler