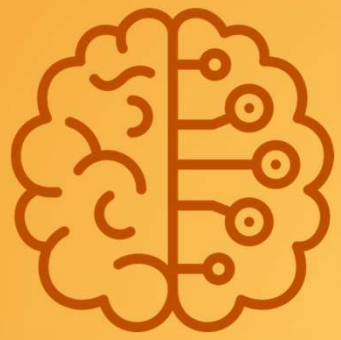
CS 4375 – Introduction to Machine Learning

Naïve Bayes

Erick Parolin

THE UNIVERSITY OF TEXAS AT DALLAS



[Slides adapted from Dr. Vibhav Gogate and Dr. Nicholas Ruozzi]

Supervised Learning

Input: *Dataset* $D = (x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})$

where $x^{(i)}$ is the i^{th} data point and $y^{(i)}$ is the i^{th} label associated to this data point.

Want: find a function h such that $h(x)$ is a good approximation to y.

Later: Use function h to predict unseen data points x

Bayesian Approach

Bayes Rule: Given x, compute the following for each $Y=y_i$

$$P(Y = y_i|x) = \frac{P(x|Y = y_i)\, P(Y=y_i)}{P(x)}$$

Then, assign to *x* the class with the highest probability:

$$\boldsymbol{Class\ of\ x} = arg \max_{y_i} \mathrm{P}(Y = y_i|x)$$

Not necessary to compute the normalization constant *P(x):*

$$\boldsymbol{Class\ of\ x} = arg \max_{y_i} P(x|Y = y_i)\, P(Y = y_i)$$

Bayesian Approach

- Assume x is composed by $\boldsymbol{n}$ features $x_1, x_2, \ldots, x_n$, and we have $\boldsymbol{m}$ distinct classes in dataset D

- If we want to compute $\mathbf{P}(\boldsymbol{x}|\mathbf{Y} = \boldsymbol{y_i})\ \mathbf{P}(\mathbf{Y} = \boldsymbol{y_i})$ to classify x, then we need to compute and store:

  - Distribution ***P(Y)***
  - Conditional joint distribution $\mathbf{P}(\boldsymbol{x}|\mathbf{Y} = \boldsymbol{y_i})$ for all $x_1, x_2, \ldots, x_n$ and Y's.

  e.g., P(x_1=1, x_2=1, …, x_n=1 | Y=1) , P(x_1=1, x_2=1, …, x_n=1 | Y=2) , …, P(x_1=1, x_2=1, …, x_n=1 | Y=m),
  P(x_1=0, x_2=1, …, x_n=1 | Y=1), P(x_1=0, x_2=1, …, x_n=1 | Y=2) , …, P(x_1=0, x_2=1, …, x_n=1 | Y=m),
  P(x_1=0, x_2=0, …, x_n=1 | Y=1), P(x_1=0, x_2=0, …, x_n=1 | Y=2), …, P(x_1=0, x_2=0, …, x_n=1 | Y=m),
  ……

- That's quite impractical: Assuming predictor variables are all Boolean, we would have $m \cdot 2^n$

Naïve Bayes

- What if we assume conditional independence?
  - "All features are conditionally independent of each other given the class variable."
- "A and B are conditionally independent given C if and only if, given knowledge that C occurs, knowledge of whether A occurs provides no information on the likelihood of B occurring, and knowledge of whether B occurs provides no information on the likelihood of A occurring"

$$P(A,B|C) = P(A|C)\,P(B|C)$$

Naïve Bayes

- Assuming conditional independence, we have:

$$\mathbf{P}(\boldsymbol{x}|\mathbf{Y} = \boldsymbol{y_i}) = \prod_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{P}(\boldsymbol{x_j}|\boldsymbol{Y} = \boldsymbol{y_i})$$

where $x = (x_1, \ldots, x_n)$ denotes the assignment of values to all features $X_j \in X$ such that feature X_j is assigned the value x_j .

- Now, the number of parameters is linear: O(m·n), where $\boldsymbol{m}$ denotes the number of classes and **n** is the number of features

- Model description:
  - m Class Priors: $\boldsymbol{P(Y)}$
  - m · n Conditional Distributions: for each $\boldsymbol{X_j}$ $\boldsymbol{P}(\boldsymbol{X_j}|\boldsymbol{Y})$

Naïve Bayes: Learning Algorithm

Maximum Likelihood Estimate (MLE)

- **Want:** Estimate the probability tables $\boldsymbol{P(Y)}$ and $\boldsymbol{P(x_j|Y = y_i)}$
- The log-likelihood of the data is

$$\log \prod_{d=1}^{|D|} P(Y = y^{(d)}) \prod_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{P}(x_j^{(d)}|\mathrm{Y} = y^{(d)})$$

where $\mathrm{D} = \{(\mathrm{x}^{(1)}, \mathrm{y}^{(1)}), \ldots, (\mathrm{x}^{(|\mathrm{D}|)}, \mathrm{y}^{(|\mathrm{D}|)})\}$ denotes the training dataset

Naïve Bayes: Learning Algorithm

Maximum Likelihood Estimate (MLE)

- The log-likelihood of the data is

$$\log \prod_{d=1}^{|D|} P(Y = y^{(d)}) \prod_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{P}(x_j^{(d)} | \mathrm{Y} = y^{(d)})$$

- Taking the derivatives and setting to zero, we have:
  - Estimate of $\boldsymbol{P(Y = y_i)} = \frac{\mathbf{Count}(Y=y_i)}{|D|}$
  - Estimate of $\boldsymbol{P(X_j = x_j | Y = y_i)} = \frac{\mathbf{Count}(Y=y_i \text{ and } X_j=x_j)}{\mathbf{Count}(Y=y_i)}$

NB for Classification

How to Classify a Test Example?

- Just need to plug the estimations

$$y = h_{NB}(x) = arg\max_{y_i} P(Y = y_i)\, P(x_1, \ldots, x_n | Y = y_i)$$

$$= arg\max_{y_i}\ P(Y = y_i) \prod_{j=1}^{n} P(x_i | Y = y_i)$$

Subtleties of Naive Bayes

- The conditional independence assumption is often violated in practice, but Naïve Bayes still works surprisingly well!
  - **Plausible reason:** Only need the probability of the correct class to be the largest!
  - **Example (for binary classification)**: just need to figure out the correct side of 0.5 and not the actual probability (0.51 is the same as 0.99).
- What if you never see a training instance (X_j=a, Y=y_i)?

  If $P(x_1|Y = y_i) = 0$, then no matter what values $x_2, \dots, x_n$ will take

  $$P(x_1, x_2, \dots, x_n|Y = y_i) = 0$$

Laplace Smoothing

- Pretend that you saw each outcome ***k*** extra times than it occurred…

$$\text{Estimate of } P(X_j = x_j | Y = y_i) = \frac{Count(Y=y_i \text{ and } X_j=x_j) + \boldsymbol{k}}{\text{Count}(Y=y_i) + n\boldsymbol{k}}$$

where **n** denotes the number of features

- In practice, the hyperparameter k is usually set to 1 (1-Laplace).

- This is the same as a MAP estimation of $P(X_j = x_j | Y = y_i)$
  - Setting k=0 for Laplace will be the same as MLE

Gaussian Naïve Bayes

What if the features $X_i \in X$ are continuous?

- Class Priors: $\boldsymbol{P(Y = y_i)}$ does not change!
- $\boldsymbol{P(X_j = x_j | Y = y_i)}$ will be given by $\mathcal{N}(\mu_{i,j}, \sigma^2_{i,j})$, a normal distribution with mean $\mu_{i,j}$ and variance $\sigma^2_{i,j}$.
- Maximum Likelihood Estimates:
  - Estimate of $\mu_{i,j} = \widehat{\mu_{i,j}} = \frac{\sum_d^I x_j^{(d)}}{|I|}$
  - Estimate of $\sigma^2_{i,j} = \widehat{\sigma^2_{i,j}} = \frac{\sum_d^I \left(x_j^{(d)} - \widehat{\mu_{i,j}}\right)^2}{|I| - 1}$

  where $I \subseteq D \; such \; that \; class(I) = i$ (All training datapoints whose class is i)

Gaussian Naïve Bayes

What if the features $X_i \in X$ are continuous?

- Then during classification, it is just plugging the estimated parameters into normal distribution:

$$P(X_j = x_j | Y = y_i) = \frac{1}{\sqrt{2\pi\widehat{\sigma_{i,j}^2}}} \; e^{-\frac{\left(x_j - \widehat{\mu_{i,j}}\right)^2}{2\widehat{\sigma_{i,j}^2}}}$$

- **Example:**
  - Want to compute $P(temperature = 85 \mid playTennis = Yes)$
  - $\widehat{\mu_{\text{temp, play}}} = 89.75$
  - $\widehat{\sigma^2_{\text{temp play}}} = 40.19$

$$P(temperature = 85 \mid playTennis = Yes) = \frac{1}{\sqrt{2\pi \cdot 40.19}} \; e^{-\frac{(85-89.75)^2}{2 \cdot 40.19}} = 0.0833$$

Numerical Example

PlayTennis: training examples

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

Given the weather conditions (*outlook, temperature, humidity and wind*), we want to predict whether it is a good day to play tennis (class label is *PlayTennis*).

Numerical Example

PlayTennis: training examples

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

Computing estimates of $\boldsymbol{P(Y = y_i)}$ and $\boldsymbol{P(X_j = x_j | Y = y_i)}$

P(Outlook=*o* | Play=*b*)

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

P(Temperature=*t* | Play=*b*)

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

P(Humidity=*h* | Play=*b*)

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

P(Wind=*w* | Play=*b*)

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

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

Numerical Example

P(Outlook=*o* | Play=*b*)

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

P(Temperature=*t* | Play=*b*)

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

P(Humidity=*h* | Play=*b*)

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

P(Wind=*w* | Play=*b*)

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

P(Play=*Yes*) = 9/14 *P*(Play=*No*) = 5/14

Predicting the class for a new instance *X*'

X'=(**Outlook**=*Sunny*, **Temperature**=*Cool*, **Humidity**=*High*, **Wind**=*Strong*)

From look up tables, we have:

P(**Outlook**=*Sunny* | **PlayTennis**=*Yes*)=2/9
P(**Temperature**=*Cool* | **PlayTennis**=*Yes*)=3/9
P(**Humidity**=*High* | **PlayTennis**=*Yes*)=3/9
P(**Wind**=*Strong*| **PlayTennis**=*Yes*)=3/9
P(**PlayTennis**=*Yes*)=9/14

P(**Outlook**=*Sunny* | **PlayTennis**=*No*)=3/5
P(**Temperature**=*Cool* | **PlayTennis**=*No*)=1/5
P(**Humidity**=*High* | **PlayTennis**=*No*)=4/5
P(**Wind**=*Strong*| **PlayTennis**=*No*)=3/5
P(**PlayTennis**=*No*)=5/14

P(**PlayTennis**=*Yes*| X') = P(Sunny|Yes) P(Cool|Yes) P(High|Yes) P(Strong|Yes) P(Yes) = 0.0053
P(**PlayTennis**=*No*| X') = P(Sunny|No) P(Cool|No) P(High|No) P(Strong|No) P(No) = 0.0206

P(**PlayTennis**=*No*| X') > P(**PlayTennis**=*Yes*| X') ➔ **Class(X') = No**

Numerical Example

Another example…

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Given the attributes (*GiveBirth, CanFly, LiveInWater, HaveLegs*), we want to predict if the animal is mammal or non-mammal.

Numerical Example

Another example…

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Predicting the class for a new instance X'

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

$$P(M|X') = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(M|X')P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(N|X') = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(N|X')P(N) = 0.0042 \times \frac{13}{20} = 0.0027$$

$$P(M|X')P(M) > P(N|X')P(N) \Rightarrow \boldsymbol{mammal}$$

Naïve Bayes

Summary

- It is simple and easy to implement
- It handles both discrete and continuous (Gaussian NB) data
- Scalability: highly scalable with the number of predictors and data points (both for training and classifying)
- Robust to irrelevant attributes
- Assumes all features are conditionally independent of each other given the class variable
- "Zero-frequency" problem can be handled using Laplace or other smoothing technique.

Readings

- **Machine Learning by Tom Mitchell – Sections 6.9 – 6.10**
- **Machine Learning: A Probabilistic Perspective by Kevin Murphy – Section 3.5**