Lecture 9: Naive Bayes, SVM, Kernels

Instructor: Saravanan Thirumuruganathan
Outline

1. Probability basics
2. Probabilistic Interpretation of Classification
3. Bayesian Classifiers, Naive Bayes
4. Support Vector Machines
Probability Basics
Sample Space

- **Sample Space:** A space of events that we assign probabilities to
- Events can be binary, multi-valued, or continuous
- Events are mutually exclusive
- Examples:
  - Coin flip: \{head, tail\}
  - Die roll: \{1,2,3,4,5,6\}
  - English words: a dictionary
  - Temperature: $\mathbb{R}_+$ (Kelvin)
A variable, $X$, whose domain is the sample space, and whose value is somewhat uncertain

Examples:
- $X = \text{coin flip outcome}$
- $X = \text{first word in tomorrow’s headline news}$
- $X = \text{tomorrow’s temperature}$
Probability $P(X = a)$ is the fraction of times $x$ takes value $a$

Often we write it as $P(a)$

Examples:

- Fair Coin: $P(\text{head}) = P(\text{tail}) = 0.5$
- Slightly Biased Coin: $P(\text{head}) = 0.51$, $P(\text{tail}) = 0.49$
- Two Face’s Coin: $P(\text{head}) = 1$, $P(\text{tail}) = 0$
- Fair Dice: $P(\text{getting 1 in a die roll}) = 1/6$
Probability for Discrete Events

- \( P(A=\text{“head or tail in a fair coin”}) \)
  \[
  0.5 + 0.5 = 1
  \]

- \( P(A=\text{“even number in a fair dice roll”}) \)
  \[
  \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = 0.5
  \]

- \( P(A=\text{“two dice rolls sum to 2 in a fair dice”}) \)
  \[
  \frac{1}{6} \times \frac{1}{6} = \frac{1}{36}
  \]
Axioms of Probability

- $P(A) \in [0,1]$
- $P(\text{true})=1$, $P(\text{false})=0$
- $P(A \lor B) = P(A) + P(B) - P(A \land B)$
Simple Corollaries

- \( P(A') = 1 - P(A) \)
- If \( A \) can take \( k \) different values \( a_1, \ldots, a_k \),
  \[
  P(A = a_1) + \ldots + P(A = a_k) = 1
  \]

**Law of Total Probability:**
- \( P(A) = P(A \cap B) + P(A \cap B') \)
- \( P(A) = \sum_{i=1}^{k} P(A \cap B = b_i) \) if \( B \) takes \( k \) values \( b_1, \ldots, b_k \)
Definition: A partition of the sample space $\Omega$ is a collection of disjoint events $B_1$, $B_2$, … $B_k$ whose union is $\Omega$. Such a partition divides any set $A$ into disjoint pieces:
### Probability Table

#### Weather

<table>
<thead>
<tr>
<th></th>
<th>Sunny</th>
<th>Cloudy</th>
<th>Rainy</th>
</tr>
</thead>
<tbody>
<tr>
<td>P(Weather) = sunny</td>
<td>$\frac{200}{365}$</td>
<td>$\frac{100}{365}$</td>
<td>$\frac{65}{365}$</td>
</tr>
</tbody>
</table>

- $P(\text{Weather} = \text{sunny}) = P(\text{sunny}) = \frac{200}{365}$
- $P(\text{Weather}) = \{\frac{200}{365}, \frac{100}{365}, \frac{65}{365}\}$
Joint Probability Table

<table>
<thead>
<tr>
<th></th>
<th>Sunny</th>
<th>Cloudy</th>
<th>Rainy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>hot</strong></td>
<td>150/365</td>
<td>40/365</td>
<td>5/365</td>
</tr>
<tr>
<td><strong>cold</strong></td>
<td>50/365</td>
<td>60/365</td>
<td>60/365</td>
</tr>
</tbody>
</table>

- \( P(\text{temp}=\text{hot}, \text{weather}=\text{rainy}) = P(\text{hot}, \text{rainy}) = \frac{5}{365} \)

- The full joint probability table between \( N \) variables, each taking \( k \) values, has \( k^N \) entries (**that's a lot!**)
### Marginal Probability Table

- **Sum over other variables**

<table>
<thead>
<tr>
<th>temp</th>
<th>Sunny</th>
<th>Cloudy</th>
<th>Rainy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>hot</strong></td>
<td>150/365</td>
<td>40/365</td>
<td>5/365</td>
</tr>
<tr>
<td><strong>cold</strong></td>
<td>50/365</td>
<td>60/365</td>
<td>60/365</td>
</tr>
</tbody>
</table>

\[
P(\text{Weather}) = \{200/365, 100/365, 65/365\}
\]

- The name comes from the old days when the sums are written on the margin of a page.
Marginal Probability Table

- **Sum over other variables**

<table>
<thead>
<tr>
<th>temp</th>
<th>Sunny</th>
<th>Cloudy</th>
<th>Rainy</th>
<th>(\Sigma)</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>150/365</td>
<td>40/365</td>
<td>5/365</td>
<td>195/365</td>
</tr>
<tr>
<td>cold</td>
<td>50/365</td>
<td>60/365</td>
<td>60/365</td>
<td>170/365</td>
</tr>
</tbody>
</table>

\[ P(\text{temp}) = \{195/365, 170/365\} \]

- This is nothing but \(P(B) = \sum_{i=1}^{k} P(B \land A=a_i)\), if \(A\) can take \(k\) values
Conditional Probability

- $P(A = a | B = b) = \text{fraction of times when random variable } A \text{ took a value of } a, \text{ within the region where random variable } B = b$

The definition $P(A | B) = \frac{P(A \cap B)}{P(B)}$ restricts the sample space to $B$, and rescales to give $P(B|B) = 1$.
Conditional Probability

Consider a roll of a fair dice

- $A$: it rolled 1. $P(A) = 1/6$

- $B$: it rolled an odd number. $P(B) = 3/6 = 0.5$

Suppose, I knew that $B$ happened. What is the probability that $A$ happened?
Consider a roll of a fair dice

- $A$: it rolled 1. $P(A) = 1/6$
- $B$: it rolled an odd number. $P(B) = 3/6 = 0.5$

Suppose, I knew that $B$ happened. What is the probability that $A$ happened?

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{1/6}{1/2} = \frac{1}{3}$$
Conditional Probability

- **Conditional Probability:** \( P(A|B) = \frac{P(A \cap B)}{P(B)} \)
- **Multiplication Rule:** \( P(A \cap B) = P(A|B)P(B) \)
- **Chain Rule:**
  - \( P(A_1, A_2) = P(A_1)P(A_2|A_1) \)
  - \( P(A_1, A_2, A_3) = P(A_3|A_1, A_2)P(A_1, A_2) \)
    \[ = P(A_3|A_1, A_2)P(A_2|A_1)P(A_1) \]
  - \( P(A_1, A_2, A_3) = P(A_1|A_2, A_3)P(A_2, A_3) \)
    \[ = P(A_1|A_2, A_3)P(A_2|A_3)P(A_3) \]
  - \( P(\cap_{i=1}^k A_i) = \prod_{i=1}^k P\left(A_i|\cap_{j=1}^{i-1} A_j\right) \)
Bayes Theorem

- \( P(A|B) = \frac{P(A \cap B)}{P(B)} \)
- \( P(B|A) = \frac{P(A \cap B)}{P(A)} \)
- **Proof of Bayes Theorem:**

\[
P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)
\]

\[
P(A|B)P(B) = P(B|A)P(A)
\]

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

Two events $A$, $B$ are independent, if (all 3 definitions are equivalent)

- $P(A \cap B) = P(A)P(B)$
- $P(A|B) = P(A)$
- $P(B|A) = P(B)$
A famous statistician would never travel by airplane, because he had studied air travel and estimated that the probability of there being a bomb on any given flight was one in a million, and he was not prepared to accept these odds.

One day, a colleague met him at a conference far from home. "How did you get here, by train?"

"No, I flew"

"What about the possibility of a bomb?"

"Well, I began thinking that if the odds of one bomb are 1:million, then the odds of two bombs are (1/1,000,000) x (1/1,000,000). This is a very, very small probability, which I can accept. So now I bring my own bomb along!"

An innocent old math joke
Independence

- Independence between random variables is typically obtained via domain knowledge.
- Suppose $A$ and $B$ be two independent random variables that can take $k$ different values $a_1, \ldots, a_k$ and $b_1, \ldots, b_k$.
- The joint probability table typically has $k^2$ parameters.
- If random variables are independent, then only $2k - 2$ parameters.
  - $k = 2$, 4 vs 2
  - $k = 10$, 100 vs 18
  - $k = 100$, 10,000 vs 198
- This is something great for data mining!
Conditional Independence

- Random variables can be dependent, but **conditionally independent**
- Your house has an alarm
  - Neighbor John will call when he hears the alarm
  - Neighbor Mary will call when she hears the alarm
  - Assume John and Mary don't talk to each other
- JohnCall independent of MaryCall?
Conditional Independence

- Random variables can be dependent, but **conditionally independent**
- Your house has an alarm
  - Neighbor John will call when he hears the alarm
  - Neighbor Mary will call when she hears the alarm
  - Assume John and Mary don’t talk to each other
- **JohnCall** independent of **MaryCall**?
  - No - If John called, likely the alarm went off, which increases the probability of Mary calling
  - \( P(MaryCall|JohnCall) \neq P(MaryCall) \)
Conditional Independence

- If we know the status of the alarm, JohnCall won’t affect Mary at all
- \[ P(\text{MaryCall}|\text{Alarm, JohnCall}) = P(\text{MaryCall}|\text{Alarm}) \]
- We say JohnCall and MaryCall are conditionally independent, given Alarm
- In general A, B are conditionally independent given C
  - \[ P(A|B, C) = P(A|C) \]
  - \[ P(B|A, C) = P(B|C) \]
  - \[ P(A, B|C) = P(A|C) \times P(B|C) \]
Probabilistic Interpretation of Classification
Probabilistic Classifiers

- Type of classifiers that, given an input, produces a *probability distribution* over a set of classes
  - Probability that this email is spam is X and not spam is Y
  - Probability that this person has tumour is X and no tumour is Y
  - Probability that this digit is 0 is X, 1 is Y, ...  

- Most state of the art classifiers are probabilistic

- Even $k$-NN and Decision trees have probabilistic interpretations
Prior Probability

- $P(A)$: Prior or unconditional probability of $A$
- Your belief in $A$ in the absence of additional information
- Uninformative priors
  - Principle of indifference: Assign equal probabilities to all possibilities
  - Coin toss, $P(\text{head}) = P(\text{tail}) = 1/2$
- Often you get from domain knowledge or from data
- $P(\text{email is spam}) = 0.8$
Conditional Probability as Belief Update

- $P(A)$: Prior belief in $A$
- $P(A|B)$: Belief after obtaining information $B$
- $P(A|B, C)$: Belief after obtaining information $B$ and $C$
Suppose you work as security guard in Airport

Your job: look at people in security line and choose some for additional screening

You want to pick passengers with high “risk”

A: Passenger is high risk

By experience, you know only 0.1% of passengers are high risk (Prior probability)

\[^2\text{http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work}\]
Consider a random person:

- The probability that this person is high risk is $A$ is 0.1%
- Suppose you notice that the person is male

---

Consider a random person:
- The probability that this person is high risk is \( A \) is 0.1%
- Suppose you notice that the person is male
  - There are more male criminals than female ones
- The passenger is nervous

---

http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work
Consider a random person:

- The probability that this person is high risk is $A$ is 0.1%
- Suppose you notice that the person is male
  - There are more male criminals than female ones
- The passenger is nervous
  - Most criminals are nervous but most normal passengers are not
- The passenger is a kid

---

3http://www.quora.com/In-laymans-terms-how-does-Naive-Bayes-work
Consider a random person:
- The probability that this person is high risk is $A$ is 0.1%
- Suppose you notice that the person is male
  - There are more male criminals than female ones
- The passenger is nervous
  - Most criminals are nervous but most normal passengers are not
- The passenger is a kid

Then we observe candies drawn from some bag: 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭 🍭

What kind of bag is it? What flavour will the next candy be?
Conditional Probability

- $X, A = \langle A_1, A_2, \ldots, A_d \rangle$: Input feature vector
- $Y, C$: Class value to predict
- $P(C|A)$ vs $P(A|C)$
Conditional Probability

- \(X, A = \langle A_1, A_2, \ldots, A_d \rangle\): Input feature vector
- \(Y, C\): Class value to predict
- \(P(C|A)\) vs \(P(A|C)\)
- Key terms
  - \(P(A), P(C)\): Prior probability
  - \(P(A|C)\): Class conditional probability or likelihood (from training data)
  - \(P(C|A)\): Posterior probability
Likelihood vs Posterior

- $P(A|C)$: Likelihood, $P(C|A)$: Posterior

Examples:
- $P(\text{Viagra}|\text{Spam})$ and $P(\text{Spam}|\text{Viagra})$:
Likelihood vs Posterior

- $P(A|C)$: Likelihood, $P(C|A)$: Posterior

Examples:
- $P($Viagra$|$Spam) and $P($Spam$|$Viagra$)$: Likelihood, Posterior
- $P($High temperature$|$Flu$)$ and $P($Flu$|$High Temperature$)$:
Likelihood vs Posterior

- $P(A|C)$: Likelihood, $P(C|A)$: Posterior
- Examples:
  - $P$(Viagra|Spam) and $P$(Spam|Viagra): Likelihood, Posterior
  - $P$(High temperature|Flu) and $P$(Flu|High Temperature): Likelihood, Posterior
  - $P$(Fever, Headache, Cough|Flu) and $P$(Flu|Fever, Headache, Cough)
  - $P$(Viagra, Nigeria, Lottery|Spam) and $P$(Spam|Viagra, Nigeria, Lottery)
Bayes’ Theorem

- Training data gives us likelihood and prior
- Prediction requires Posterior
- Bayes rule allows us to do statistical inference
- \[ P(C|A) = \frac{P(A|C)P(C)}{P(A)} \]
- \[ P(C|A) \propto P(A|C)P(C) \]
- Posterior \( \propto \) Likelihood \( \propto \) Prior
Bayes Decision Rule

- “When you hear hoofbeats, think of horses not zebras”
- When predicting, assign the class with highest posterior probability
- To categorize email:
  - Compute $P(\text{spam}|\text{email})$ and $P(\text{not spam}|\text{email})$
  - If $P(\text{spam}|\text{email}) > P(\text{not spam}|\text{email})$, decide email as spam.
  - Else as not-spam
Bayesian Classifiers
Bayes Classifier

- A probabilistic framework for solving classification problems
- Conditional Probability:
  \[ P(C | A) = \frac{P(A, C)}{P(A)} \]
  \[ P(A | C) = \frac{P(A, C)}{P(C)} \]
- Bayes theorem:
  \[ P(C | A) = \frac{P(A | C)P(C)}{P(A)} \]
Example of Bayes Theorem

- **Given:**
  - A doctor knows that meningitis causes stiff neck 50% of the time
  - Prior probability of any patient having meningitis is 1/50,000
  - Prior probability of any patient having stiff neck is 1/20

- If a patient has stiff neck, what’s the probability he/she has meningitis?

$$P(M \mid S) = \frac{P(S \mid M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$
Bayesian Classifiers

- Consider each attribute and class label as random variables

- Given a record with attributes \((A_1, A_2, \ldots, A_n)\)
  - Goal is to predict class \(C\)
  - Specifically, we want to find the value of \(C\) that maximizes \(P(C| A_1, A_2, \ldots, A_n)\)

- Can we estimate \(P(C| A_1, A_2, \ldots, A_n)\) directly from data?
Bayesian Classifiers

- Approach:
  - compute the posterior probability $P(C \mid A_1, A_2, \ldots, A_n)$ for all values of $C$ using the Bayes theorem

$$
P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C)P(C)}{P(A_1 A_2 \ldots A_n)}$$

- Choose value of $C$ that maximizes $P(C \mid A_1, A_2, \ldots, A_n)$

- Equivalent to choosing value of $C$ that maximizes $P(A_1, A_2, \ldots, A_n \mid C) P(C)$

- How to estimate $P(A_1, A_2, \ldots, A_n \mid C)$?
- Assume independence among attributes $A_i$ when class is given:
  \[
  P(A_1, A_2, \ldots, A_n | C) = P(A_1 | C_j) \cdot P(A_2 | C_j) \cdots P(A_n | C_j)
  \]
- Can estimate $P(A_i | C_j)$ for all $A_i$ and $C_j$.
- New point is classified to $C_j$ if $P(C_j) \prod P(A_i | C_j)$ is maximal.
How to Estimate Probabilities from Data?

Class: $P(C) = \frac{N_c}{N}$
- e.g., $P(\text{No}) = \frac{7}{10}$, $P(\text{Yes}) = \frac{3}{10}$

For discrete attributes:
$$P(A_i \mid C_k) = \frac{|A_{ik}|}{N_{C_k}}$$
- where $|A_{ik}|$ is number of instances having attribute $A_i$ and belongs to class $C_k$
- Examples:
  - $P(\text{Status=Married}\mid\text{No}) = \frac{4}{7}$
  - $P(\text{Refund=Yes}\mid\text{Yes}) = 0$
How to Estimate Probabilities from Data?

- For continuous attributes:
  - **Discretize** the range into bins
    - one ordinal attribute per bin
    - violates independence assumption
  - **Two-way split**: $(A < v)$ or $(A > v)$
    - choose only one of the two splits as new attribute
  - **Probability density estimation**:
    - Assume attribute follows a normal distribution
    - Use data to estimate parameters of distribution (e.g., mean and standard deviation)
    - Once probability distribution is known, can use it to estimate the conditional probability $P(A_i|c)$
How to Estimate Probabilities from Data?

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Evade</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Yes</td>
<td>Single</td>
<td>125K</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Married</td>
<td>100K</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>Single</td>
<td>70K</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>Married</td>
<td>120K</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>No</td>
<td>Divorced</td>
<td>95K</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>No</td>
<td>Married</td>
<td>60K</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Yes</td>
<td>Divorced</td>
<td>220K</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>No</td>
<td>Single</td>
<td>85K</td>
<td>Yes</td>
</tr>
<tr>
<td>9</td>
<td>No</td>
<td>Married</td>
<td>75K</td>
<td>No</td>
</tr>
<tr>
<td>10</td>
<td>No</td>
<td>Single</td>
<td>90K</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- **Normal distribution:**
  \[
P(A_i \mid c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} e^{-\frac{(A_i - \mu_i)^2}{2\sigma_{ij}^2}}
\]
  - One for each \((A_i, c_i)\) pair

- **For (Income, Class=No):**
  - If Class=No
    - sample mean = 110
    - sample variance = 2975

\[
P(\text{Income} = 120 \mid \text{No}) = \frac{1}{\sqrt{2\pi(54.54)}} e^{-\frac{(120-110)^2}{2(2975)}} = 0.0072
\]
Example of Naive Bayes Classifier

Given a Test Record:

\[ X = (\text{Refund} = \text{No}, \text{Married}, \text{Income} = 120K) \]

naive Bayes Classifier:

- \[ P(X|\text{Class}=\text{No}) = P(\text{Refund}=\text{No}|\text{Class}=\text{No}) \]
  \[ \times P(\text{Married}| \text{Class}=\text{No}) \]
  \[ \times P(\text{Income}=120K| \text{Class}=\text{No}) \]
  \[ = \frac{4}{7} \times \frac{4}{7} \times 0.0072 = 0.0024 \]

- \[ P(X|\text{Class}=\text{Yes}) = P(\text{Refund}=\text{No}| \text{Class}=\text{Yes}) \]
  \[ \times P(\text{Married}| \text{Class}=\text{Yes}) \]
  \[ \times P(\text{Income}=120K| \text{Class}=\text{Yes}) \]
  \[ = 1 \times 0 \times 1.2 \times 10^{-9} = 0 \]

Since \( P(X|\text{No})P(\text{No}) > P(X|\text{Yes})P(\text{Yes}) \)
Therefore \( P(\text{No}|X) > P(\text{Yes}|X) \)
\[ \Rightarrow \text{Class} = \text{No} \]
Naive Bayes Classifier

- If one of the conditional probability is zero, then the entire expression becomes zero
- Probability estimation:

  Original: \( P(A_i \mid C) = \frac{N_{ic}}{N_c} \)

  Laplace: \( P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c} \)  
  \( c \): number of classes

  m-estimate: \( P(A_i \mid C) = \frac{N_{ic} + mp}{N_c + m} \)  
  \( p \): prior probability  
  \( m \): parameter
### Example of Naive Bayes Classifier

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

**A:** attributes

**M:** mammals

**N:** non-mammals

\[
P(A | M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06
\]

\[
P(A | N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042
\]

\[
P(A | M)P(M) = 0.06 \times \frac{7}{20} = 0.021
\]

\[
P(A | N)P(N) = 0.004 \times \frac{13}{20} = 0.0027
\]

\[
P(A | M)P(M) > P(A | N)P(N)
\]

=> Mammals
Naive Bayes Classifier Summary

- Robust to isolated noise points
- Handle missing values by ignoring the instance during probability estimate calculations
- Robust to irrelevant attributes
- Independence assumption may not hold for some attributes
  - Use other techniques such as Bayesian Belief Networks (BBN)
Support Vector Machines
A hyperplane in $p$ dimensions is a flat affine subspace of dimension $p - 1$.

In general the equation for a hyperplane has the form

$$\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p = 0$$

In $p = 2$ dimensions a hyperplane is a line.

If $\beta_0 = 0$, the hyperplane goes through the origin, otherwise not.

The vector $\beta = (\beta_1, \beta_2, \ldots, \beta_p)$ is called the normal vector — it points in a direction orthogonal to the surface of a hyperplane.
HyperPlane in 2D

\[ \beta_1 X_1 + \beta_2 X_2 - 6 = 0 \]

\[ \beta_1 X_1 + \beta_2 X_2 - 6 = 1.6 \]

\[ \beta_1 X_1 + \beta_2 X_2 - 6 = -4 \]

\[ \beta_1 = 0.8 \]
\[ \beta_2 = 0.6 \]
Find a linear hyperplane (decision boundary) that will separate the data
Support Vector Machines

One Possible Solution
Support Vector Machines

Another possible solution
Support Vector Machines
Support Vector Machines

Which one is better? B1 or B2?
How do you define better?
Find hyperplane **maximizes** the margin => B1 is better than B2
Support Vector Machines

$$\vec{w} \cdot \vec{x} + b = 0$$

$$\vec{w} \cdot \vec{x} + b = -1$$

$$\vec{w} \cdot \vec{x} + b = +1$$

$$f(\vec{x}) = \begin{cases} 
1 & \text{if } \vec{w} \cdot \vec{x} + b \geq 1 \\
-1 & \text{if } \vec{w} \cdot \vec{x} + b \leq -1 
\end{cases}$$

Margin = \frac{2}{\|\vec{w}\|^2}$
Support Vector Machines

- We want to maximize: \( \text{Margin} = \frac{2}{\| \mathbf{w} \|^2} \)
  - Which is equivalent to minimizing: \( L(w) = \frac{\| \mathbf{w} \|^2}{2} \)
  - But subjected to the following constraints:

\[
    f(\mathbf{x}_i) = \begin{cases} 
    1 & \text{if } \mathbf{w} \cdot \mathbf{x}_i + b \geq 1 \\
    -1 & \text{if } \mathbf{w} \cdot \mathbf{x}_i + b \leq -1 
\end{cases}
\]

- This is a constrained optimization problem
  - Numerical approaches to solve it (e.g., quadratic programming)
What if the problem is not linearly separable?
What if the problem is not linearly separable?

- Introduce slack variables
  - Need to minimize:
    \[ L(w) = \frac{||\bar{w}||^2}{2} + C \sum_{i=1}^{N} \xi_i \]
  - Subject to:
    \[
    f(\bar{x}_i) = \begin{cases} 
    1 & \text{if } \bar{w} \cdot \bar{x}_i + b \geq 1 - \xi_i \\
    -1 & \text{if } \bar{w} \cdot \bar{x}_i + b \leq -1 + \xi_i 
    \end{cases}
    \]
What if decision boundary is not linear?
Nonlinear Support Vector Machines

- Transform data into higher dimensional space
Kernels

- Often we want to **capture nonlinear patterns** in the data
  - Nonlinear Regression: Input-output relationship may not be linear
  - Nonlinear Classification: Classes may not be separable by a linear boundary
- Linear models (e.g., linear regression, linear SVM) are not just rich enough

**Kernels:** Make linear models work in nonlinear settings
- By **mapping data to higher dimensions** where it exhibits linear patterns
- Apply the linear model in the new input space
- Mapping $\equiv$ changing the feature representation
Classifying non-linearly separable data

- Consider this binary classification problem
  
  - Each example represented by a **single feature** $x$
  - No linear separator exists for this data
  
  - Now map each example as $x \rightarrow \{x, x^2\}$
    
    - Each example now has **two features** (“derived” from the old representation)
    
    - Data now becomes linearly separable in the new representation
Classifying non-linearly separable data

- Let's look at another example:

- Each example defined by a **two features** $\mathbf{x} = \{x_1, x_2\}$
- No linear separator exists for this data
- Now map each example as $\mathbf{x} = \{x_1, x_2\} \rightarrow \mathbf{z} = \{x_1^2, \sqrt{2}x_1x_2, x_2^2\}$
  - Each example now has **three features** (“derived” from the old representation)
- Data now becomes linearly separable in the new representation
Feature Mapping

- Consider the following mapping $\phi$ for an example $\mathbf{x} = \{x_1, \ldots, x_D\}$

$$
\phi : \mathbf{x} \rightarrow \{x_1^2, x_2^2, \ldots, x_D^2, x_1 x_2, x_1 x_2, \ldots, x_1 x_D, \ldots, \ldots, x_{D-1} x_D\}
$$

- It's an example of a quadratic mapping
  - Each new feature uses a pair of the original features

- **Problem:** Mapping usually leads to the number of features blow up!
  - Computing the mapping itself can be inefficient in such cases
  - Moreover, *using* the mapped representation could be inefficient too
    - e.g., imagine computing the similarity between two examples: $\phi(\mathbf{x})^T \phi(\mathbf{z})$

- Thankfully, Kernels help us avoid both these issues!
  - The mapping doesn't have to be explicitly computed
  - Computations with the mapped features remain efficient
Consider two examples \( x = \{x_1, x_2\} \) and \( z = \{z_1, z_2\} \)

Let’s assume we are given a function \( k \) (kernel) that takes as inputs \( x \) and \( z \)

\[
\begin{align*}
k(x, z) &= (x^\top z)^2 \\
      &= (x_1 z_1 + x_2 z_2)^2 \\
      &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2 \\
      &= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^\top (z_1^2, \sqrt{2}z_1 z_2, z_2^2) \\
      &= \phi(x)^\top \phi(z)
\end{align*}
\]

The above \( k \) implicitly defines a mapping \( \phi \) to a higher dimensional space

\[
\phi(x) = \{x_1^2, \sqrt{2}x_1 x_2, x_2^2\}
\]

Note that we didn’t have to define/compute this mapping.

Simply defining the kernel a certain way gives a higher dim. mapping \( \phi \)

Moreover the kernel \( k(x, z) \) also computes the dot product \( \phi(x)^\top \phi(z) \)

\( \phi(x)^\top \phi(z) \) would otherwise be much more expensive to compute explicitly

All kernel functions have these properties
Kernels: Formally Defined

- Recall: Each kernel $k$ has an associated feature mapping $\phi$
- $\phi$ takes input $x \in \mathcal{X}$ (input space) and maps it to $\mathcal{F}$ (“feature space”)
- Kernel $k(x, z)$ takes two inputs and gives their similarity in $\mathcal{F}$ space
  
  \[
  \phi : \mathcal{X} \rightarrow \mathcal{F} \\
  k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}, \quad k(x, z) = \phi(x)^\top \phi(z)
  \]
- $\mathcal{F}$ needs to be a vector space with a dot product defined on it
  - Also called a Hilbert Space
- Can just any function be used as a kernel function?
  - No. It must satisfy Mercer’s Condition
Popular Kernels

The following are the most popular kernels for real-valued vector inputs:

- **Linear (trivial) Kernel:**
  $$k(x, z) = x^\top z$$ (mapping function $\phi$ is identity - no mapping)

- **Quadratic Kernel:**
  $$k(x, z) = (x^\top z)^2 \quad \text{or} \quad (1 + x^\top z)^2$$

- **Polynomial Kernel (of degree $d$):**
  $$k(x, z) = (x^\top z)^d \quad \text{or} \quad (1 + x^\top z)^d$$

- **Radial Basis Function (RBF) Kernel:**
  $$k(x, z) = \exp[-\gamma \| x - z \|^2]$$
  - $\gamma$ is a hyperparameter (also called the kernel bandwidth)
  - The RBF kernel corresponds to an infinite dimensional feature space $\mathcal{F}$ (i.e., you can’t actually write down the vector $\phi(x)$)
Kernels can turn a linear model into a nonlinear one.

Recall: Kernel $k(x, z)$ represents a dot product in some high dimensional feature space $\mathcal{F}$.

Any learning algorithm in which examples only appear as dot products $(x_i^T x_j)$ can be kernelized (i.e., non-linearized).

.. by replacing the $x_i^T x_j$ terms by $\phi(x_i)^T \phi(x_j) = k(x_i, x_j)$.

Most learning algorithms are like that.

- Perceptron, SVM, linear regression, etc.

- Many of the unsupervised learning algorithms too can be kernelized (e.g., $K$-means clustering, Principal Component Analysis, etc.)
Summary

Major Concepts:

- Probabilistic interpretation of Classification
- Bayesian Classifiers
- Naive Bayes Classifier
- Support Vector Machines (SVM)
- Kernels
Slide Material References

- Slides from TSK Book, Chapter 5
- Slides from Piyush Rai
- See also the footnotes