Classification

KNN Classifier, Naive Bayesian Classifier
Outline

1. KNN Classifier
2. Naive Bayesian Classifier
**k-Nearest-Neighbor classifiers**

- First introduced in early 1950s
- Distance metrics
  - Generally Euclidean distance is used when the values are continuous
  - Hamming distance for categorical/nominal values
Algorithm idea

- Let $k$ be the number of nearest neighbors and $D$ be the set of training examples
- **for** each test example $z = (x', y')$, do
  - Compute $d(x, x')$, the distance between $z$ and every example $(x, y) \in D$
  - Select $D_z \subseteq D$, the set of $k$ closest training examples to $z$
  - $y' = \underset{v}{\arg\max} \sum_{(x_i, y_i) \in D_z} I(v == y_i)$ //Majority voting
- **end for**

$v$: a class label

$y_i$: the class label for one of the NNs

$I$: indicate function that returns value 1 if its argument is true and 0 otherwise.
Computation can be **costly** if the number of training examples is large.

- **Efficient indexing techniques** are available to find the nearest neighbors of a test example.

- **Sensitive $k$:** reduce the impact of $k$ is to weight the influence of a NN $x_i$: $w_i = \frac{1}{d(x', x_i)^2}$

Distance-weighted voting:

$$y' = \arg\max_v \sum_{(x_i, y_i) \in D_z} w_i \times I(v == y_i)$$
Characteristics

- **Instance-based** learning: (1) require a proximity measure (2) make predictions without having to maintain a model.

- **Lazy learner**: do not require model building.

- Predictions are made based on *local information*. Thus, it is quite susceptible to noise.

- Can produce *arbitrarily shaped decision boundaries*. More flexible compared with the decision tree approach.
References

kNN classifier – R

> install.packages('class')
> library(class)
> help(knn)

https://cran.r-project.org/web/packages/class/class.pdf
Naive Bayesian Classifier

- **Non-deterministic** relationship between attribute set and the class label.
  - Reason 1: noisy data
  - Reason 2: complex factors
- **Probabilistic relationships** between attribute set and the class label.
- Bayesian classifiers: a probabilistic framework for solving classification problems.
Bayes Theorem

- 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|S) = P(S|M)P(M) / P(S)
```

```
P(M|S) = (1/2) · (1/50,000) / (1/20) = 20/100,000 = 0.0002
```
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?

- **Solve this problem:**
  - $M$: a patient has meningitis
  - $S$: a patient has stiff neck
  - $P(S|M) = 50\%, P(M) = 1/50,000$, $P(S) = 1/20$, $P(M|S) =$?

  $$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{(1/2) \cdot (1/50,000)}{1/20} = \frac{20}{100,000} = 0.0002$$
Bayesian Classifiers

- Consider each attribute and class label as random variables.
- Given a record with attributes $(A_1, A_2, \cdots, A_n)$
  - Goal is to predict class $C$.
  - Specifically, we want to find the value of $C$ that maximizes $P(C | A_1, A_2, \cdots, A_n)$.
- Can we estimate $P(C | A_1, A_2, \cdots, A_n)$ directly from data?
Bayesian Classifiers – approach

- Compute the posterior probability $P(C|A_1, A_2, \cdots, A_n)$ for all values of $C$ using the Bayes theorem

$$P(C|A_1, A_2, \cdots, A_n) = \frac{P(A_1, A_2, \cdots, A_n|C) P(C)}{P(A_1, A_2, \cdots, A_n)}$$

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

$$\equiv \text{maximize } P(A_1, A_2, \cdots, A_n|C) P(C)$$

- How to estimate $P(A_1, A_2, \cdots, A_n|C)$?
Naive Bayes Classifier

- **Assume independence** among attributes $A_i$ when class is given

$$P(A_1, A_2, \cdots, A_n|C_j) = P(A_1|C_j)P(A_2|C_j) \cdots P(A_n|C_j)$$

- 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.
## Estimate Probabilities from Data

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</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>

- **Class:** \( P(C = C_i) = \frac{N_i}{N} \)
- **Discrete attributes:** \( P(A_i = a_i|C = C_j) = \frac{N_{ij}}{N_j} \)
  - \( N_{ij} \): # of instances having attribute \( A_i = a_i \) and belonging to class \( C_j \)
  - \( N_j \): # of instances belonging to class \( C_j \)
- **P(No) = 0.7, P(Yes) = 0.3**
- **\( P(Status = Married|No) = \frac{4}{7} \) P(Refund = Yes|Yes) = 0**
Estimate Probabilities from Data – continuous attributes

- **Discretize** the range into bins
  - one ordinal attribute per bin
- **Two-way split**: \((A < v)\) or \((A \geq v)\)
  - 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 \(\mu\) and standard deviation \(\sigma\))
  - Once probability distribution is known, can use it to estimate the conditional probability

\[
P(A_i | c_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} \exp \left( -\frac{(A_i - \mu_{ij})^2}{2\sigma^2_{ij}} \right)
\]

- \(\mu_{ij}\): estimated as sample mean of \(A_i\) for all training records with class \(c_j\).
- \(\sigma_{ij}\): estimated as sample variance \(s^2\) of all training records with class \(c_j\).
Estimate Probabilities from Data – Example

<table>
<thead>
<tr>
<th>Tid</th>
<th>Refund</th>
<th>Marital Status</th>
<th>Taxable Income</th>
<th>Cheat</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>

\[ P(A_i | c_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} \exp\left(-\frac{(A_i - \mu_{ij})^2}{2\sigma_{ij}^2}\right) \]

For (Income, Class=No)
Sample mean: \( \frac{125 + 100 + 70 + 120 + 60 + 220 + 75}{7} = 110 \)
Sample variance: 2975

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

Given a test record: X=(Refund=No, Married, Income=120K)

\[ P(\text{Refund} = \text{Yes}|\text{No}) = \frac{3}{7} \]
\[ P(\text{Refund} = \text{No}|\text{No}) = \frac{4}{7} \]
\[ P(\text{Refund} = \text{Yes}|\text{Yes}) = 0 \]
\[ P(\text{Marital Status} = \text{Single}|\text{No}) = \frac{2}{7} \]
\[ P(\text{Marital Status} = \text{Divorced}|\text{No}) = \frac{1}{7} \]
\[ P(\text{Marital Status} = \text{Married}|\text{No}) = \frac{4}{7} \]
\[ P(\text{Marital Status} = \text{Single}|\text{Yes}) = \frac{2}{3} \]
\[ P(\text{Marital Status} = \text{Divorced}|\text{Yes}) = \frac{1}{3} \]
\[ P(\text{Marital Status} = \text{Married}|\text{Yes}) = 0 \]

\[ P(\text{Class} = \text{Yes}) = 0.3 \]
\[ P(\text{Class} = \text{No}) = 0.7 \]

For taxable income

If class=No: Sample mean=110, variance=2975

If class=Yes: Sample mean=90, variance=25
Example 1 (cont.)

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

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

Since \( P(X|\text{No}) \times P(\text{No}) > P(X|\text{Yes}) \times P(\text{Yes}) \)
\( P(\text{No}|X) > P(\text{Yes}|X) \rightarrow \text{class} = \text{No} \)

\( P(\text{Married}|\text{Yes}) = 0 \). If one of the conditional probability is zero, then the entire expression becomes zero.
Naive Bayes Classifier – Probability estimation for Zero conditional probability

- **Original**: \( P(A_i = a_i | C = C_j) = \frac{N_{ij}}{N_j} \)
- **Laplace**: \( P(A_i = a_i | C = C_j) = \frac{N_{ij} + 1}{N_j + c} \)
- **m-estimate**: \( P(A_i = a_i | C = C_j) = \frac{N_{ij} + mp}{N_j + m} \)

- \( N_{ij} \): the number of instances with attribute value \( a_i \) and belonging to a class \( C_j \)
- \( N_j \): the number of instances belonging to a class \( C_j \)
- \( c \): total number of classes
- \( p \): prior probability for the positive class \( C_j \)
- \( m \): parameter

E.g., use m-estimate: \( m = 3 \) and \( p = 0.3 \) (prior for positive class Yes)
\[
P(\text{Marital Status} = \text{Married} | \text{Yes}) = \frac{0 + 0.3 \times 3}{3 + 3} = \frac{0.9}{6}
\]
Probability estimation - Zero probability in R package e1071

- R package e1071, method naiveBayes assumes
  - independence of the predictor variables, and
  - Gaussian distribution (given the target class) of metric predictors.
  - For attributes with missing values, the corresponding table entries are omitted for prediction.

- Laplace and threshold are used to adjust the attribute likelihood $P(A_i = a_i | C = C_j)$.

- Laplace is a positive number controlling Laplace smoothing, which adjusts the attribute likelihood $P(A_i = a_i | C = C_j)$ to be
  $$
  \frac{N_{ij} + \text{laplace}}{N_j + |A_i| \times \text{laplace}},
  $$
  where
  - $|A_i|$ is the number of different values for attribute $A_i$.
  - This parameter is only used to process attributes with categorical values.
  - Default laplace value is zero.

- The threshold parameter is used to replace the zero attribute likelihood $P(A_i = a_i | C = C_j)$. 
### Example of Naive Bayes Classifier – Example 2

<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>no</td>
<td>no</td>
<td>no</td>
<td>non-mammals</td>
</tr>
<tr>
<td>salmon</td>
<td>no</td>
<td>no</td>
<td>yes</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>no</td>
<td>sometimes</td>
<td>yes</td>
<td>non-mammals</td>
</tr>
<tr>
<td>komodo</td>
<td>no</td>
<td>no</td>
<td>no</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>no</td>
<td>yes</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>no</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>sometimes</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>
Example 2 (cont.)

Give test case:

<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></td>
<td>yes</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>?</td>
</tr>
</tbody>
</table>

- A: attributes
- M: mammals
- N: non-mammals
Example 2 (cont.)

- $P(M) = \frac{7}{20}, P(N) = \frac{13}{20}$
- $P(\text{GB} = Y|M) = \frac{6}{7}, P(\text{GB} = N|M) = \frac{1}{7}$
- $P(\text{Can Fly} = Y|M) = \frac{1}{7}, P(\text{Can Fly} = N|M) = \frac{6}{7}$
- $P(\text{Live in Water} = Y|M) = \frac{2}{7}, P(\text{Live in Water} = N|M) = \frac{5}{7}$
- $P(\text{Have Legs} = Y|M) = \frac{5}{7}, P(\text{Have Legs} = N|M) = \frac{2}{7}$

\[
P(X|M) = P(\text{GB} = Y|M) \times P(\text{Can Fly} = N|M) \times P(\text{Live in Water} = Y|M) \times P(\text{Have Legs} = N|M) \\
= \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06
\]

\[
P(X|N) = P(\text{GB} = Y|N) \times P(\text{Can Fly} = N|N) \times P(\text{Live in Water} = Y|N) \times P(\text{Have Legs} = N|N) \\
= \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042
\]

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

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

Since $P(X|M) \times P(M) > P(X|N) \times P(N)$, class = Mammals
Naive Bayesian classifier – R

```r
> install.packages('e1071', dependencies = TRUE)
> library(e1071)
> help(naiveBayes)

https://cran.r-project.org/web/packages/e1071/e1071.pdf
Naive Bayesian classifier – R

Sometimes, we may run into the following mistake:

```r
model_nb <- naiveBayes(trn.classlabel ~ ., data = trn.data)
predict(model_nb, tst.data, type=c("class"))
#factor(0)
#Levels:
```

- One possible reason is that your `trn.classlabel` are treated as numerical values.
- They should be categorical class labels as shown in `naiveBayes` method description.
- *Computes the conditional a-posterior probabilities of a categorical class variable given independent predictor variables using the Bayes rule.*

You may want to run this to see whether the class labels are categorical values.

```r
is.factor(trn.classlabel)
```

If they are not categorical class labels, the following change should work

```r
model_nb <- naiveBayes(as.factor(trn.classlabel) ~ ., data = trn.data)
predict(model_nb, tst.data, type=c("class"))
```
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)