Naïve Bayes, Support Vector Machines

Danna Gurari
University of Texas at Austin
Spring 2021

https://www.ischool.utexas.edu/~dannag/Courses/IntroToMachineLearning/CourseContent.html
Review

• Last week:
  • Multiclass classification applications and evaluating models
  • Motivation for new ML era: need non-linear models
  • Nearest neighbor classification
  • Decision tree classification
  • Parametric versus non-parametric models

• Assignments (Canvas)
  • Problem set 3 due tonight
  • Problem set 4 out and due in two weeks (after spring break)
  • Lab assignment 2 out and due in three weeks

• Questions?
Today’s Topics

• Evaluating Machine Learning Models Using Cross-Validation

• Naïve Bayes

• Support Vector Machines
Today’s Topics

• Evaluating Machine Learning Models Using Cross-Validation

• Naïve Bayes

• Support Vector Machines
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

<table>
<thead>
<tr>
<th>Input:</th>
<th>Label:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Cat" /></td>
<td>Hairy</td>
</tr>
<tr>
<td><img src="image2" alt="Dog" /></td>
<td>Hairy</td>
</tr>
<tr>
<td><img src="image3" alt="Glass" /></td>
<td>Not Hairy</td>
</tr>
<tr>
<td><img src="image4" alt="Cat" /></td>
<td>Hairy</td>
</tr>
</tbody>
</table>

- **Hairy**
- **Not Hairy**
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

Classifier predicts well when test data matches training data. Lucky?
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

Classifier *predicts poorly* when test data does not match training data. Unlucky?
Goal: Design Models that **Generalize** Well to New, Previously Unseen Examples

How to know if good/bad evaluation scores happen from good/bad luck?
Evaluation of Classification Model

Cross-validation: limit influence of chosen dataset split
Evaluation of Classification Model

e.g., 3-fold cross-validation

Input:

<table>
<thead>
<tr>
<th>Hairy</th>
<th>Hairy</th>
<th>Not Hairy</th>
<th>Not Hairy</th>
</tr>
</thead>
</table>

Label:

**Cross-validation**
Evaluation of Classification Model

e.g., 3-fold cross-validation

Fold 1:
- train on $k-1$ partitions
- test on $k$ partitions

Fold 2:
- train on $k-1$ partitions
- test on $k$ partitions

Fold 3:
- train on $k-1$ partitions
- test on $k$ partitions
Evaluation of Classification Model

e.g., 3-fold cross-validation

Classifier accuracy: prediction accuracy across all folds of test data
Evaluation of Classification Model

Input:

Label:

Hairy  Hairy  Not Hairy  Not Hairy

1/5  1/5  1/5  1/5  1/5

e.g., 5-fold cross-validation

How many partitions of the data to create?
Evaluation of Classification Model

e.g., 5-fold cross-validation

**Input:**
- Cat
- Dog
- Glass
- Spider

**Label:**
- Hairy
- Hairy
- Not Hairy
- Not Hairy

Iter 1: test data  Iter 2: test data  Iter 3: test data  Iter 4: test data  Iter 5: test data

How many iterations of train & test to run?
Evaluation of Classification Model

How many partitions of the data to create?

e.g., 10-fold cross-validation

Input:

Label:

Hairy  Hairy  Not Hairy  Not Hairy

1/10  1/10  1/10  1/10  1/10  1/10  1/10  1/10  1/10  1/10
Evaluation of Classification Model

e.g., 10-fold cross-validation

Input:

Label:

How many iterations of train & test to run?
Evaluation of Classification Model

e.g., k-fold cross-validation

What are the (dis)advantages of using larger values for “k”?
K-Fold Cross-Validation: How to Partition Data?

• e.g., 3-fold cross validation?
Stratified k-fold Cross Validation

- e.g., 3-fold cross validation? Preserve class proportions in each fold to represent proportions in the whole dataset

```
In [4]: iris.target

Out[4]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
          0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
          1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
          1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
          2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
          2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
          2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])
```
Stratified k-fold Cross Validation

https://github.com/amueller/introduction_to_ml_with_python/blob/master/05-model-evaluation-and-improvement.ipynb
Stratified k-fold Cross Validation

https://github.com/amueller/introduction_to_ml_with_python/blob/master/05-model-evaluation-and-improvement.ipynb
Group Discussion: Cross Validation

• Why would you choose cross validation over percentage split (train/val/test split)?

• Why would you choose percentage split (train/val/test split) over cross validation?

• What does high variance of test accuracy between different folds tell you?

• Does cross validation build a final model for use on new data?
Today’s Topics

• Evaluating Machine Learning Models Using Cross-Validation

• Naïve Bayes

• Support Vector Machines
Historical Context of ML Models

Human “Computers”

1613

1945
First programmable machine

1950
Turing Test

1956
AI

1957
Perceptron

1959
Machine Learning

1962
Decision Trees

1961
Naïve Bayes


1rst AI Winter

2nd AI Winter

Early 1800

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression

Linear regression
Naïve Bayes

- Learns a model of the joint probability of the input features and each class, and then picks the most probable class.
Background: Probability

• Joint probability: $P(A, B)$
  - i.e., probability of two events occurring simultaneously

• How to calculate joint probability?

• Can use **chain rule**: $P(A, B) = P(B | A) \times P(A)$
  - $P(A | B)$ is probability of an event occurring in the presence of a second event; i.e., conditional probability
Background: Probability Example

• Calculating joint probability using chain rule
  • \( P(A, B) = P(B|A) \times P(A) \)

• e.g., Let:
  • \( A \) be the event of choosing from the blue bowl
  • \( B \) the event of choosing a Snickers

  • \( P(B|A) = ? \)

  • \( P(A, B) = \frac{2}{3} \times \frac{1}{2} = \frac{1}{3} \)
Naïve Bayes

• Learns a model of the joint probability of the input features and each class, and then picks the most probable class
Background: Bayes’ Theorem

• Given chain rule:
  - \( P(A, B) = P(A|B) \times P(B) \)
  - \( P(A, B) = P(B|A) \times P(A) \)

• We can get:
  - \( P(A|B) \times P(B) = P(B|A) \times P(A) \)

• Rearranging:
  - \( P(A|B) = \frac{P(B|A) \times P(A)}{P(B)} \)

• Rewriting:

\[
P(C_i|\text{features}) = \frac{P(\text{features}|C_i) \times P(C_i)}{P(\text{features})} \]

Want to find class with the largest probability

Constant for all classes... so can ignore this!
Naïve Bayes

• Learns a model of the joint probability of the input features and each class, and then picks the most probable class.
Naïve Bayes: Naively Assumes Features Are Class Conditionally Independent

\[
P(C_i | \text{features}) = P(\text{features} | C_i) \times P(C_i)
\]

\[
P(\text{features} | C_i) = \prod_{j=1}^{m} P(x_j | C_i)
\]

\[
P(\text{features} | C_i) = P(x_1 | C_i) \times P(x_2 | C_i) \times \ldots \times P(x_m | C_i)
\]

\[
P(C_i | \text{features}) = P(x_1 | C_i) \times P(x_2 | C_i) \times \ldots \times P(x_m | C_i) \times P(C_i)
\]

If we assume independence then

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

However, in many cases such an assumption maybe too strong (more later in the class)
Naïve Bayes: Different Generative Models Can Yield the Observed Features

Recall: Want to find class with the largest probability

Key Decision: How to compute probability of each feature given the class?

\[ P(C_i|\text{features}) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_m|C_i) \times P(C_i) \]
Naïve Bayes: Different Generative Models Can Yield the Observed Features

- **Gaussian** Naïve Bayes (typically used for “continuous”-valued features)
  - Assume data drawn from a Gaussian distribution: mean + standard deviation

\[
P(C_i | \text{features}) = P(x_1 | C_i) \times P(x_2 | C_i) \times \cdots \times P(x_m | C_i) \times P(C_i)
\]
Naïve Bayes: Different Generative Models Can Yield the Observed Features

- **Multinomial** Naïve Bayes (typically used for “discrete”-valued features)
  - Assume count data and computes fraction of entries belonging to the category

<table>
<thead>
<tr>
<th></th>
<th>Movie</th>
<th>Type</th>
<th>Length</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>Comedy</td>
<td>Short</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>m2</td>
<td>Drama</td>
<td>Medium</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>m3</td>
<td>Comedy</td>
<td>Medium</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>m4</td>
<td>Drama</td>
<td>Long</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>m5</td>
<td>Drama</td>
<td>Medium</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>m6</td>
<td>Drama</td>
<td>Short</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>m7</td>
<td>Comedy</td>
<td>Short</td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>m8</td>
<td>Drama</td>
<td>Medium</td>
<td></td>
<td>Yes</td>
</tr>
</tbody>
</table>

\[
P(C_i|\text{features}) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_m|C_i) \times P(C_i)
\]
Gaussian Naïve Bayes: Example

\[
P(C_i|\text{features}) = P(x_1|C_i) \times P(C_i)
\]

<table>
<thead>
<tr>
<th>IMDb Rating</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Yes</td>
</tr>
<tr>
<td>9.3</td>
<td>Yes</td>
</tr>
<tr>
<td>5.1</td>
<td>No</td>
</tr>
<tr>
<td>6.9</td>
<td>No</td>
</tr>
<tr>
<td><strong>8.3</strong></td>
<td>Yes</td>
</tr>
<tr>
<td>4.5</td>
<td>No</td>
</tr>
<tr>
<td>8.0</td>
<td>Yes</td>
</tr>
<tr>
<td>7.5</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- \( P(\text{Liked}) = ? \)
- \( \frac{5}{8} = 0.625 \)
Gaussian Naïve Bayes: Example

- \( P(\text{Liked}) = \frac{5}{8} = 0.625 \)
- \( P(\text{Not Liked}) = \frac{3}{8} = 0.375 \)

<table>
<thead>
<tr>
<th>( X_1 )</th>
<th>IMDb Rating</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>7.2</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>9.3</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>5.1</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>6.9</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>8.3</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>4.5</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>8.0</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>7.5</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\[
P(C_i|\text{features}) = P(x_1|C_i) \times P(C_i)
\]
Gaussian Naïve Bayes: Example

\[ x_1 \]

<table>
<thead>
<tr>
<th>IMDb Rating</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Yes</td>
</tr>
<tr>
<td>9.3</td>
<td>Yes</td>
</tr>
<tr>
<td>5.1</td>
<td>No</td>
</tr>
<tr>
<td>6.9</td>
<td>No</td>
</tr>
<tr>
<td>8.3</td>
<td>Yes</td>
</tr>
<tr>
<td>4.5</td>
<td>No</td>
</tr>
<tr>
<td>8.0</td>
<td>Yes</td>
</tr>
<tr>
<td>7.5</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- P(Liked) = 5/8 = 0.625
- P(Not Liked) = 3/8 = 0.375
- P(IMDb Rating | Liked): Mean and Standard Deviation?
  - Mean = 8.06
  - Standard Deviation = 0.81

\[ P(C_i | features) = P(x_1 | C_i) \times P(C_i) \]
Gaussian Naïve Bayes: Example

• \( P(\text{Liked}) = \frac{5}{8} = 0.625 \)
• \( P(\text{Not Liked}) = \frac{3}{8} = 0.375 \)
• \( P(\text{IMDb Rating} \mid \text{Liked}) \)
  • Mean = 8.06
  • Standard Deviation = 0.81
• \( P(\text{IMDb Rating} \mid \text{Not Liked}) \): Mean and Standard Deviation?
  • Mean = 5.5
  • Standard Deviation = 1.25

<table>
<thead>
<tr>
<th>IMDb Rating</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Yes</td>
</tr>
<tr>
<td>9.3</td>
<td>Yes</td>
</tr>
<tr>
<td>5.1</td>
<td>No</td>
</tr>
<tr>
<td>6.9</td>
<td>No</td>
</tr>
<tr>
<td>8.3</td>
<td>Yes</td>
</tr>
<tr>
<td>4.5</td>
<td>No</td>
</tr>
<tr>
<td>8.0</td>
<td>Yes</td>
</tr>
<tr>
<td>7.5</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Gaussian Naïve Bayes: Example

- **P(Liked) = 5/8 = 0.625**
- **P(Not Liked) = 3/8 = 0.375**
- **P(IMDb Rating | Liked)**
  - Mean = 8.06
  - Standard Deviation = 0.81
- **P(IMDb Rating | Not Liked)**
  - Mean = 5.5
  - Standard Deviation = 1.25

<table>
<thead>
<tr>
<th>IMDb Rating</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Yes</td>
</tr>
<tr>
<td>9.3</td>
<td>Yes</td>
</tr>
<tr>
<td>5.1</td>
<td>No</td>
</tr>
<tr>
<td>6.9</td>
<td>No</td>
</tr>
<tr>
<td>8.3</td>
<td>Yes</td>
</tr>
<tr>
<td>4.5</td>
<td>No</td>
</tr>
<tr>
<td>8.0</td>
<td>Yes</td>
</tr>
<tr>
<td>7.5</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Test Example

- IMDb Rating: 6.4
  - P(Liked | Features)

(Can Use: https://planetcalc.com/4986/)
Gaussian Naïve Bayes: Example

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>IMDb Rating</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>9.3</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>6.9</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>8.3</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>8.0</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>7.5</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

- $P(\text{Liked}) = \frac{5}{8} = 0.625$
- $P(\text{Not Liked}) = \frac{3}{8} = 0.375$
- $P(\text{IMDb Rating} | \text{Liked})$
  - Mean = 8.06
  - Standard Deviation = 0.81
- $P(\text{IMDb Rating} | \text{Not Liked})$
  - Mean = 5.5
  - Standard Deviation = 1.25

Test Example

IMDb Rating: 6.4

- $P(\text{Liked} | \text{Features})$
  - $= 0.06 \times 0.625$

\[
P(C_i | \text{features}) = P(x_1 | C_i) \times P(C_i)
\]
Gaussian Naïve Bayes: Example

\[ P(C_i | features) = P(x_1 | C_i) \times P(C_i) \]

- \( P(\text{Liked}) = 5/8 = 0.625 \)
- \( P(\text{Not Liked}) = 3/8 = 0.375 \)
- \( P(\text{IMDb Rating} | \text{Liked}) \)
  - Mean = 8.06
  - Standard Deviation = 0.81
- \( P(\text{IMDb Rating} | \text{Not Liked}) \)
  - Mean = 5.5
  - Standard Deviation = 1.25

Test Example

IMDb Rating: 6.4

- \( P(\text{Liked} \mid \text{Features}) \)
  - \( = 0.06 \times 0.625 \)
  - \( = 0.0375 \)
- \( P(\text{Not Liked} \mid \text{Features}) \)
  - \( = 0.25 \times 0.375 \)
  - \( = 0.09 \)

Which class is the most probable?
Multinomial Naïve Bayes: Example

- \( P(\text{Liked}) = \frac{5}{8} = 0.625 \)
- \( P(\text{Not Liked}) = \frac{3}{8} = 0.375 \)
- \( P(\text{Comedy} \mid \text{Liked}) = \frac{2}{5} = 0.4 \)
- \( P(\text{Comedy} \mid \text{Not Liked}) = \frac{1}{3} = 0.333 \)
- \( P(\text{Drama} \mid \text{Liked}) = \frac{3}{5} = 0.6 \)
- \( P(\text{Drama} \mid \text{Not Liked}) = \frac{2}{3} = 0.666 \)

\[
P(C_i \mid \text{features}) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times P(C_i)
\]
## Multinomial Naïve Bayes: Example

### Table

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>Comedy</td>
<td>Short</td>
<td>Yes</td>
</tr>
<tr>
<td>m2</td>
<td>Drama</td>
<td>Medium</td>
<td>Yes</td>
</tr>
<tr>
<td>m3</td>
<td>Comedy</td>
<td>Medium</td>
<td>No</td>
</tr>
<tr>
<td>m4</td>
<td>Drama</td>
<td>Long</td>
<td>No</td>
</tr>
<tr>
<td>m5</td>
<td>Drama</td>
<td>Medium</td>
<td>Yes</td>
</tr>
<tr>
<td>m6</td>
<td>Drama</td>
<td>Short</td>
<td>No</td>
</tr>
<tr>
<td>m7</td>
<td>Comedy</td>
<td>Short</td>
<td>Yes</td>
</tr>
<tr>
<td>m8</td>
<td>Drama</td>
<td>Medium</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- $P(\text{Short} \mid \text{Liked}) = ?$
  - $2/5 = 0.4$
- $P(\text{Short} \mid \text{Not Liked}) = ?$
  - $1/3 = 0.333$
- $P(\text{Medium} \mid \text{Liked}) = ?$
  - $3/5 = 0.6$
- $P(\text{Medium} \mid \text{Not Liked}) = ?$
  - $1/3 = 0.333$
- $P(\text{Long} \mid \text{Liked}) = ?$
  - $0/5 = 0$
- $P(\text{Long} \mid \text{Not Liked}) = ?$
  - $1/3 = 0.333$

\[
P(C_i \mid \text{features}) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times P(C_i)
\]
Multinomial Naïve Bayes: Example

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movie</td>
<td>Type</td>
<td>Length</td>
</tr>
<tr>
<td>m1</td>
<td>Comedy</td>
<td>Short</td>
</tr>
<tr>
<td>m2</td>
<td>Drama</td>
<td>Medium</td>
</tr>
<tr>
<td>m3</td>
<td>Comedy</td>
<td>Medium</td>
</tr>
<tr>
<td>m4</td>
<td>Drama</td>
<td>Long</td>
</tr>
<tr>
<td>m5</td>
<td>Drama</td>
<td>Medium</td>
</tr>
<tr>
<td>m6</td>
<td>Drama</td>
<td>Short</td>
</tr>
<tr>
<td>m7</td>
<td>Comedy</td>
<td>Short</td>
</tr>
<tr>
<td>m8</td>
<td>Drama</td>
<td>Medium</td>
</tr>
</tbody>
</table>

Which class is the most probable?

$P(\text{Liked}) = 0.63$

- $P(\text{Not Liked}) = 0.38$
- $P(\text{Comedy} \mid \text{Liked}) = 0.4$
- $P(\text{Comedy} \mid \text{Not Liked}) = 0.33$
- $P(\text{Drama} \mid \text{Liked}) = 0.6$
- $P(\text{Drama} \mid \text{Not Liked}) = 0.67$

$P(\text{Short} \mid \text{Liked}) = 0.4$

- $P(\text{Short} \mid \text{Not Liked}) = 0.33$
- $P(\text{Medium} \mid \text{Liked}) = 0.6$
- $P(\text{Medium} \mid \text{Not Liked}) = 0.33$
- $P(\text{Long} \mid \text{Liked}) = 0$
- $P(\text{Long} \mid \text{Not Liked}) = 0.33$

$P(\text{Liked} \mid \text{Features}) = 0.4 \times 0.6 \times 0.63 = 0.15$

$P(\text{Not Liked} \mid \text{Features}) = 0.33 \times 0.33 \times 0.38 = 0.04$
### Multinomial Naïve Bayes: Example

#### Test Example

- **Type:** Comedy
- **Length:** Long

Which class is the most probable?

<table>
<thead>
<tr>
<th>Movie</th>
<th>Type</th>
<th>Length</th>
<th>Liked?</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1</td>
<td>Comedy</td>
<td>Short</td>
<td>Yes</td>
</tr>
<tr>
<td>m2</td>
<td>Drama</td>
<td>Medium</td>
<td>Yes</td>
</tr>
<tr>
<td>m3</td>
<td>Comedy</td>
<td>Medium</td>
<td>No</td>
</tr>
<tr>
<td>m4</td>
<td>Drama</td>
<td>Long</td>
<td>No</td>
</tr>
<tr>
<td>m5</td>
<td>Drama</td>
<td>Medium</td>
<td>Yes</td>
</tr>
<tr>
<td>m6</td>
<td>Drama</td>
<td>Short</td>
<td>No</td>
</tr>
<tr>
<td>m7</td>
<td>Comedy</td>
<td>Short</td>
<td>Yes</td>
</tr>
<tr>
<td>m8</td>
<td>Drama</td>
<td>Medium</td>
<td>Yes</td>
</tr>
</tbody>
</table>

- $P(\text{Liked}) = 0.63$
- $P(\text{Not Liked}) = 0.38$
- $P(\text{Comedy} \mid \text{Liked}) = 0.4$
- $P(\text{Comedy} \mid \text{Not Liked}) = 0.33$
- $P(\text{Drama} \mid \text{Liked}) = 0.6$
- $P(\text{Drama} \mid \text{Not Liked}) = 0.67$
- $P(\text{Short} \mid \text{Liked}) = 0.4$
- $P(\text{Short} \mid \text{Not Liked}) = 0.33$
- $P(\text{Medium} \mid \text{Liked}) = 0.6$
- $P(\text{Medium} \mid \text{Not Liked}) = 0.33$
- $P(\text{Long} \mid \text{Liked}) = 0$
- $P(\text{Long} \mid \text{Not Liked}) = 0.33$

To avoid zero, assume training data is so large that adding one to each count makes a negligible difference

$$P(C_i \mid \text{features}) = P(x_1 \mid C_i) \times P(x_2 \mid C_i) \times P(C_i)$$
What are Naïve Bayes’ Strengths

• Training is relatively fast
• Once trained, prediction is fast
• Can work well in the absence of a large dataset
• Requires little memory (a few computed statistics)
What are Naïve Bayes’ Weaknesses

• Makes a strong, often unrealistic assumption that the presence of a each feature is completely unrelated to the presence of other features
Today’s Topics

• Evaluating Machine Learning Models Using Cross-Validation

• Naïve Bayes

• Support Vector Machines
Historical Context of ML Models

1613
1945
1959
1962

Human “Computers”
Early 1800
Linear regression
First programmable machine
Turing Test
AI
Perceptron
K-nearest neighbors
Decision Trees
Machine Learning
Naïve Bayes
Support Vector Machines
1992

Support Vector Machine (SVM) Motivation

To which class would each hyperplane (decision boundary) assign the new data point?

https://en.wikipedia.org/wiki/Linear_separability
Support Vector Machine (SVM) Motivation

Which hyperplane would you choose to separate data?

https://en.wikipedia.org/wiki/Linear_separability
Support Vector Machine (SVM) Motivation

Idea: choose hyperplane that maximizes the “margin” width.

Margin: distance between the separating hyperplane and training samples (“support vectors”) closest to the hyperplane.

https://en.wikipedia.org/wiki/Linear_separability
Support Vector Machine (SVM) Motivation

When trying to maximize the margin, what happens to the choice of line when you add outliers to the dataset?

https://en.wikipedia.org/wiki/Linear_separability
Support Vector Machine (SVM): Formalizing Definition

Derivation of Margin Length

- Subtract two equations from each other:

\[ \mathbf{w}^T (\mathbf{x}_{pos} - \mathbf{x}_{neg}) = 2 \]

- Normalize by length of \( \mathbf{w} \) to compute margin length:

\[
\frac{\mathbf{w}^T (\mathbf{x}_{pos} - \mathbf{x}_{neg})}{\| \mathbf{w} \|} = \frac{2}{\| \mathbf{w} \|}
\]

where:

\[
\| \mathbf{w} \| = \sqrt{\sum_{j=1}^{m} w_j^2}
\]

Distance Between Parallel Lines Tutorial:
http://web.mit.edu/zoya/www/SVM.pdf
Support Vector Machine (SVM): Formalizing Definition

Same as finding parameters \((w, w_0)\) that maximizes margin:

\[
\min_{w,w_0} \frac{1}{2}||w||^2 \\
\text{s.t. } \forall i \left( y^{(i)} \left( w_0 + w^T x^{(i)} \right) \right) \geq 1,
\]

Decision boundary \(w^T x = 0\)

"negative" hyperplane \(w^T x = -1\)

"positive" hyperplane \(w^T x = 1\)

SVM: Maximize the margin

\[
w_0 x_0 + w_1 x_1 + \cdots + w_m x_m = \sum_{j=0}^{m} x_j w_j = w^T x
\]

Python Machine Learning; Raschkka & Mirjalili
Support Vector Machine (SVM): Training a Classifier

Same as finding parameters that maximizes margin:

\[ \begin{align*}
\min_{w, w_0} \frac{1}{2} ||w||^2 \\
\text{s.t. } \forall i \quad y^{(i)} \left( w_0 + w^T x^{(i)} \right) \geq 1,
\end{align*} \]

Can be solved with a quadratic programming solver... learn more about this at:
- “The Nature of Statistical Learning and Theory, by Vladimir Vapnik
- A Tutorial on Support Vector Machines for Pattern Recognition by Chris J. C. Burges’
What if the Decision Boundary is Not Linear?

Hard-Margin Classification

Soft-Margin Classification
Introduce “slack” variable:

\[
\begin{align*}
\mathbf{w}^T \mathbf{x}^{(i)} &\geq 1 - \xi^{(i)} \quad & y^{(i)} = 1 \\
\mathbf{w}^T \mathbf{x}^{(i)} &\leq -1 + \xi^{(i)} \quad & y^{(i)} = -1
\end{align*}
\]

\[
\min \frac{1}{2} \|\mathbf{w}\|^2 - \lambda \sum_{i=1}^{N} \xi_i
\]

s.t. \( \xi_i \geq 0; \quad \forall i \), \( t^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \xi_i = 0 \)
Soft-Margin Classification

Controls how much slack is allowed:

\[
\min \frac{1}{2} \|w\|^2 + \lambda \sum_{i=1}^{N} \xi_i \\
\text{s.t. } \xi_i \geq 0; \ \forall i \ \ t^{(i)}(w^T x^{(i)}) \geq 1 - \xi_i = 0
\]
Soft-Margin Classification

\[
\min \frac{1}{2} \|\mathbf{w}\|^2 - \lambda \sum_{i=1}^{N} \xi_i \\
\text{s.t.} \quad \xi_i \geq 0; \quad \forall i \quad t^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)}) \geq 1 - \xi_i = 0
\]

Which plot shows when the slack variable is larger?

(A) Increases priority placed on minimizing error

Python Machine Learning; Raschkka & Mirjalili
Soft-Margin Classification

\[ \min \frac{1}{2} \|w\|^2 - \lambda \sum_{i=1}^{N} \xi_i \]

s.t. \( \xi_i \geq 0; \quad \forall i \quad t^{(i)}(w^T x^{(i)}) \geq 1 - \xi_i = 0 \)

Which plot shows when the slack variable is smaller?

A

B

(Increases priority placed on maximizing margin)

Python Machine Learning; Raschkka & Mirjalili
What if the Decision Boundary is Not Linear?

Hard-Margin Classification

Soft-Margin Classification

Python Machine Learning; Raschkka & Mirjalili
Kernelized Support Vector Machines

- Recall polynomial regression?
  - Project features to higher order space

- Kernels efficiently project features to higher order spaces

- What conversion to use? e.g.,
  - Polynomial kernel
  - Gaussian Radial Basis Function kernel
What are SVM’s Strengths

• Insensitive to outliers (only relies on support vectors to choose dividing line)
• Once trained, prediction is fast
• Requires little memory (rely on a few support vectors)
• Work well with high-dimensional data
What are SVM’s Weaknesses

• Prohibitive computational costs for large datasets
• Performance heavily dependent on soft margin value for non-linear classification
• Does not have a direct probabilistic interpretation
Today’s Topics

• Evaluating Machine Learning Models Using Cross-Validation

• Naïve Bayes

• Support Vector Machines