Decision Trees and SVMs

Jordan Boyd-Graber
University of Colorado Boulder
1. DECEMBER 2014

Slides adapted from Tom Mitchell, Eric Xing, and Lauren Hannah
Roadmap

- Classification: machines labeling data for us
- Last time: naïve Bayes and logistic regression
- This time:
  - Decision Trees
    - Simple, nonlinear, interpretable
  - SVMs
    - (another) example of linear classifier
    - State-of-the-art classification
  - Examples in Rattle (Logistic, SVM, Trees)
  - **Discussion:** Which classifier should I use for my problem?
Plan

Decision Trees

Learning Decision Trees

Vector space classification

Linear Classifiers

Support Vector Machines

Recap
Trees

Suppose that we want to construct a set of rules to represent the data

- can represent data as a series of if-then statements
- here, “if” splits inputs into two categories
- “then” assigns value
- when “if” statements are nested, structure is called a tree

![Decision Tree Diagram](image)
Ex: data \((X_1, X_2, X_3, Y)\) with \(X_1, X_2, X_3\) are real, \(Y\) Boolean

First, see if \(X_1 > 5\):
- if TRUE, see if \(X_1 > 8\)
  - if TRUE, return FALSE
  - if FALSE, return TRUE
- if FALSE, see if \(X_2 < -2\)
  - if TRUE, see if \(X_3 > 0\)
    - if TRUE, return TRUE
    - if FALSE, return FALSE
  - if FALSE, return TRUE
Example 1: \((X_1, X_2, X_3) = (1, 1, 1)\)

Example 2: \((X_1, X_2, X_3) = (10, -3, 0)\)
Example 1: \((X_1, X_2, X_3) = (1, 1, 1) \rightarrow TRUE\)

Example 2: \((X_1, X_2, X_3) = (10, -3, 0)\)
Example 1: \((X_1, X_2, X_3) = (1, 1, 1) \rightarrow \text{TRUE}\)

Example 2: \((X_1, X_2, X_3) = (10, -3, 0) \rightarrow \text{FALSE}\)
Trees

Terminology:

• branches: one side of a split
• leaves: terminal nodes that return values

Why trees?

• trees can be used for regression or classification
  ○ regression: returned value is a real number
  ○ classification: returned value is a class
• unlike linear regression, SVMs, naive Bayes, etc, trees fit *local models*
  ○ in large spaces, global models may be hard to fit
  ○ results may be hard to interpret
• fast, interpretable predictions
Example: Predicting Electoral Results

2008 Democratic primary:
- Hillary Clinton
- Barack Obama

Given historical data, how will a count vote?
- can extrapolate to state level data
- might give regions to focus on increasing voter turnout
- would like to know how variables interact
Example: Predicting Electoral Results

Decision Tree: The Obama-Clinton Divide

In the nominating contests so far, Senator Barack Obama has won the vast majority of counties with large black or highly educated populations. Senator Hillary Rodham Clinton has a commanding lead in less-educated counties dominated by whites. Follow the arrows for a more detailed split.

And is the high school graduation rate higher than 75 percent?

NO: This is a county with less-educated voters.
YES: This is a county with more educated voters.

Clinton wins these counties 704 to 89.
Obama wins these counties 383 to 70.

And where is the county?
North or South
West or Midwest

Clinton wins these counties 182 to 79.

In 2000, were many households poor?

YES: At least 47% earned less than $30,000.
NO: At least 53% earned more than $30,000.

Clinton wins these counties 52 to 25.

What's the population density?

Very rural

Clinton wins these counties 48 to 13.

In 2004, did Bush beat Kerry badly? (By more than 16.5 percentage points)

YES: 4.5 people per sq mile
NO: Very Republican

Clinton wins these counties 56 to 35.

Note: Chart excludes Florida and Rhode Island. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine, Texas, counties are excluded; likewise, one for primary voters and one for caucus participants.

Example: Predicting Electoral Results

Decision Tree: The Obama-Clinton Divide

In the nominating contests so far, Senator Barack Obama has won the vast majority of counties with large black or highly educated populations. Senator Hillary Rodham Clinton has a commanding lead in less-educated counties dominated by whites. Follow the arrows for a more detailed split.

- Is a county more than 20 percent black?
  - NO: There are not many African-Americans in this county.
  - YES: This county has a large African-American population.

- And is the high school graduation rate higher than 75 percent?
  - NO: This is a county with less-educated voters.
  - YES: This is a county with more educated voters.

Obama wins these counties 383 to 70.
Example: Predicting Electoral Results

Figure 1: Classification tree for county-level outcomes in the 2008 Democratic Party primary (as of April 16), by Amanda Cox for the New York Times.

- NO: This is a county with less-educated voters.
- YES: This is a county with more educated voters.

And is the high school graduation rate higher than 87 percent?

- NO: 78 to 87 percent have a diploma.
- YES: This is a highly educated county.

And where is the county?

- Northeast or South
- West or Midwest

- Clinton wins these counties 704 to 89.

In 2000, were many households poor?

- YES: At least 47% earned less than $30,000.
- NO: At least 53% earned more than $30,000.

- Clinton wins these counties 52 to 25.

- Obama wins these counties 383 to 70.

What's the population density?

- Very rural
- >61.5 people
**Decision Trees**

Example: Predicting Electoral Results

Figure 1: Classification tree for county-level outcomes in the 2008 Democratic Party primary (as of April 16), by Amanda Cox for the New York Times.

- Clinton wins these counties 52 to 25.
- What’s the population density?
  - Very rural
  - > 61.5 people per sq. mile

In 2004, did Bush beat Kerry badly? (by more than 16.5 percentage points)

- Yes
  - Very Republican
  - Clinton wins these counties 48 to 13.
- No
  - Obama wins these counties 201 to 83.

Note: Chart excludes Florida and Michigan. County-level results are not available in Alaska, Hawaii, Kansas, Nebraska, New Mexico, North Dakota or Maine. Texas counties are included here as once for primary voters and once for caucus participants.

Sources: Election results via The Associated Press; Census Bureau; Dave Leip’s Atlas of U.S. Presidential Elections
Decision Trees

Decision tree representation:

- Each internal node tests an attribute
- Each branch corresponds to attribute value
- Each leaf node assigns a classification

How would we represent as a function of $X, Y$:

- $X \text{ AND } Y$ (both must be true)
- $X \text{ OR } Y$ (either can be true)
- $X \text{ XOR } Y$ (one and only one is true)
When to Consider Decision Trees

- Instances describable by attribute-value pairs
- Target function is discrete valued
- Disjunctive hypothesis may be required
- Possibly noisy training data

Examples:
- Equipment or medical diagnosis
- Credit risk analysis
- Modeling calendar scheduling preferences
Plan

Decision Trees

Learning Decision Trees

Vector space classification

Linear Classifiers

Support Vector Machines

Recap
Top-Down Induction of Decision Trees

Main loop:
1. \( A \leftarrow \) the “best” decision attribute for next node
2. Assign \( A \) as decision attribute for node
3. For each value of \( A \), create new descendant of node
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?

\[ [29+, 35-] \leftarrow A1=? \]

\[ [21+, 5-] \]

\[ [8+, 30-] \]

\[ [29+, 35-] \leftarrow A2=? \]

\[ [18+, 33-] \]

\[ [11+, 2-] \]
Entropy: Reminder

- $S$ is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in $S$
- $p_\ominus$ is the proportion of negative examples in $S$
- Entropy measures the impurity of $S$

$$Entropy(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$
# Entropy

How spread out is the distribution of $S$:

$$p_\oplus(-\log_2 p_\oplus) + p_\ominus(-\log_2 p_\ominus)$$

$$Entropy(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$
Information Gain

Which feature $A$ would be a more useful rule in our decision tree?

$Gain(S, A) = \text{expected reduction in entropy due to sorting on } A$

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

[29+, 35-]  A1=?

[21+, 5-]   [8+, 30-]

[29+, 35-]  A2=?

[18+, 33-]   [11+, 2-]
\[ H(S) = -\frac{29}{54} \log \left( \frac{29}{54} \right) - \frac{35}{64} \log \left( \frac{35}{64} \right) = \]
\[ H(S) = -\frac{29}{54} \log \left( \frac{29}{54} \right) - \frac{35}{64} \log \left( \frac{35}{64} \right) \]

= 0.96
\[
H(S) = -\frac{29}{54} \lg \left( \frac{29}{54} \right) - \frac{35}{64} \lg \left( \frac{35}{64} \right)
= 0.96
\]

\[
\text{Gain}(S, A_1) = 0.96 - \frac{26}{64} \left[ -\frac{5}{26} \lg \left( \frac{5}{26} \right) - \frac{21}{26} \lg \left( \frac{21}{26} \right) \right]
- \frac{38}{64} \left[ -\frac{8}{38} \lg \left( \frac{8}{38} \right) - \frac{30}{38} \lg \left( \frac{30}{38} \right) \right]
= \]

\[
H(S) = -\frac{29}{54} \log \left(\frac{29}{54}\right) - \frac{35}{64} \log \left(\frac{35}{64}\right) \\
= 0.96
\]

\[
\text{Gain}(S, A_1) = 0.96 - \frac{26}{64} \left[ -\frac{5}{26} \log \left(\frac{5}{26}\right) - \frac{21}{26} \log \left(\frac{21}{26}\right) \right] \\
- \frac{38}{64} \left[ -\frac{8}{38} \log \left(\frac{8}{38}\right) - \frac{30}{38} \log \left(\frac{30}{38}\right) \right] \\
= 0.96 - 0.28 - 0.44 = 0.24
\]
Learning Decision Trees

\[ H(S) = -\frac{29}{54} \log \left( \frac{29}{54} \right) - \frac{35}{64} \log \left( \frac{35}{64} \right) \]

\[ = 0.96 \]

\[
\text{Gain}(S, A_1) = 0.96 - \frac{26}{64} \left[ -\frac{5}{26} \log \left( \frac{5}{26} \right) - \frac{21}{26} \log \left( \frac{21}{26} \right) \right] \\
- \frac{38}{64} \left[ -\frac{8}{38} \log \left( \frac{8}{38} \right) - \frac{30}{38} \log \left( \frac{30}{38} \right) \right] \\
= 0.96 - 0.28 - 0.44 = 0.24
\]

\[
\text{Gain}(S, A_2) = 0.96 - \frac{51}{64} \left[ -\frac{18}{51} \log \left( \frac{18}{51} \right) - \frac{33}{51} \log \left( \frac{33}{51} \right) \right] \\
- \frac{13}{64} \left[ -\frac{11}{13} \log \left( \frac{11}{13} \right) - \frac{2}{13} \log \left( \frac{2}{13} \right) \right] \\
= \]

\[
H(S) = -\frac{29}{54} \lg \left( \frac{29}{54} \right) - \frac{35}{64} \lg \left( \frac{35}{64} \right) \\
= 0.96
\]

\[
\text{Gain}(S, A_1) = 0.96 - \frac{26}{64} \left[ -\frac{5}{26} \lg \left( \frac{5}{26} \right) - \frac{21}{26} \lg \left( \frac{21}{26} \right) \right] \\
- \frac{38}{64} \left[ -\frac{8}{38} \lg \left( \frac{8}{38} \right) - \frac{30}{38} \lg \left( \frac{30}{38} \right) \right] \\
= 0.96 - 0.28 - 0.44 = 0.24
\]

\[
\text{Gain}(S, A_2) = 0.96 - \frac{51}{64} \left[ -\frac{18}{51} \lg \left( \frac{18}{51} \right) - \frac{33}{51} \lg \left( \frac{33}{51} \right) \right] \\
- \frac{13}{64} \left[ -\frac{11}{13} \lg \left( \frac{11}{13} \right) - \frac{2}{13} \lg \left( \frac{2}{13} \right) \right] \\
= 0.96 - 0.75 - 0.13 = 0.08
\]
## Training Examples

<table>
<thead>
<tr>
<th>Day</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
<th>PlayTennis</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D2</td>
<td>Sunny</td>
<td>Hot</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D3</td>
<td>Overcast</td>
<td>Hot</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D4</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D5</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D6</td>
<td>Rain</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>No</td>
</tr>
<tr>
<td>D7</td>
<td>Overcast</td>
<td>Cool</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D8</td>
<td>Sunny</td>
<td>Mild</td>
<td>High</td>
<td>Weak</td>
<td>No</td>
</tr>
<tr>
<td>D9</td>
<td>Sunny</td>
<td>Cool</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D10</td>
<td>Rain</td>
<td>Mild</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D11</td>
<td>Sunny</td>
<td>Mild</td>
<td>Normal</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D12</td>
<td>Overcast</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>Yes</td>
</tr>
<tr>
<td>D13</td>
<td>Overcast</td>
<td>Hot</td>
<td>Normal</td>
<td>Weak</td>
<td>Yes</td>
</tr>
<tr>
<td>D14</td>
<td>Rain</td>
<td>Mild</td>
<td>High</td>
<td>Strong</td>
<td>No</td>
</tr>
</tbody>
</table>
Which attribute is the best classifier?

Gain (S, Humidity )
\[
= 0.940 - (7/14) \cdot 0.985 - (7/14) \cdot 0.592 \\
= 0.151
\]

Gain (S, Wind)
\[
= 0.940 - (8/14) \cdot 0.811 - (6/14) \cdot 1.0 \\
= 0.048
\]
ID3 Algorithm

- Start at root, look for best attribute
- Repeat for subtrees at each attribute outcome
- Stop when information gain is below a threshold
- Bias: prefers shorter trees (Occam’s Razor)
  - a short hyp that fits data unlikely to be coincidence
  - a long hyp that fits data might be coincidence
  - Prevents overfitting (more later)
Plan

Decision Trees

Learning Decision Trees

Vector space classification

Linear Classifiers

Support Vector Machines

Recap
Suppose you have two classes: vacations and sports
Suppose you have four documents

<table>
<thead>
<tr>
<th>Sports</th>
<th>Vacations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doc₁: {ball, ball, ball, travel}</td>
<td>Doc₃: {travel, ball, travel}</td>
</tr>
<tr>
<td>Doc₂: {ball, ball}</td>
<td>Doc₄: {travel}</td>
</tr>
</tbody>
</table>

What does this look like in vector space?
Vector space classification

Put the documents in vector space

Travel

Ball
Vector space representation of documents

- Each document is a vector, one component for each term.
- Terms are axes.
- High dimensionality: 10,000s of dimensions and more
- How can we do classification in this space?
Vector space classification

- As before, the training set is a set of documents, each labeled with its class.
- In vector space classification, this set corresponds to a labeled set of points or vectors in the vector space.
- Premise 1: Documents in the same class form a contiguous region.
- Premise 2: Documents from different classes don’t overlap.
- We define lines, surfaces, hypersurfaces to divide regions.
Classes in the vector space

China

UK

Kenya

Should the document be assigned to China, UK or Kenya?
Classes in the vector space

Should the document ✭ be assigned to China, UK or Kenya?
Classes in the vector space

Find separators between the classes
Classes in the vector space

Find separators between the classes
Classes in the vector space

Based on these separators: * should be assigned to China
Vector space classification

Classes in the vector space

How do we find separators that do a good job at classifying new documents like ∗? – Main topic of today
Plan

Decision Trees

Learning Decision Trees

Vector space classification

Linear Classifiers

Support Vector Machines

Recap
Linear Classifiers

Linear classifiers

• Definition:
  ○ A linear classifier computes a linear combination or weighted sum \( \sum_i w_i x_i \) of the feature values.
  ○ Classification decision: \( \sum_i w_i x_i > \theta \)?
  ○ \( \ldots \) where \( \theta \) (the threshold) is a parameter.

• (First, we only consider binary classifiers.)

• Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).

• We call this the separator or decision boundary.

• We find the separator based on training set.

• Methods for finding separator: logistic regression, naïve Bayes, linear SVM

• Assumption: The classes are linearly separable.
Linear classifiers

- **Definition:**
  - A linear classifier computes a linear combination or weighted sum $\sum_i w_i x_i$ of the feature values.
  - Classification decision: $\sum_i w_i x_i > \theta$?
  - ... where $\theta$ (the threshold) is a parameter.

- (First, we only consider binary classifiers.)

- Geometrically, this corresponds to a line (2D), a plane (3D) or a hyperplane (higher dimensionalities).

- We call this the **separator** or **decision boundary**.

- We find the separator based on training set.

- Methods for finding separator: logistic regression, naïve Bayes, linear SVM

- Assumption: The classes are **linearly separable**.

- Before, we just talked about equations. What’s the geometric intuition?
A linear classifier in 1D

- A linear classifier in 1D is a point \( x \) described by the equation \( w_1 d_1 = \theta \).
A linear classifier in 1D

- A linear classifier in 1D is a point \( x \) described by the equation \( w_1 d_1 = \theta \)
- \( x = \theta / w_1 \)
A linear classifier in 1D is a point $x$ described by the equation $w_1 d_1 = \theta$

- $x = \theta / w_1$
- Points $(d_1)$ with $w_1 d_1 \geq \theta$ are in the class $c$. 

A linear classifier in 1D is a point $x$ described by the equation $w_1 d_1 = \theta$

- $x = \theta / w_1$
- Points $(d_1)$ with $w_1 d_1 \geq \theta$ are in the class $c$. 

A linear classifier in 1D
A linear classifier in 1D is a point \( x \) described by the equation \( w_1 d_1 = \theta \)

- \( x = \theta / w_1 \)
- Points \((d_1)\) with \( w_1 d_1 \geq \theta \) are in the class \( c \).
- Points \((d_1)\) with \( w_1 d_1 < \theta \) are in the complement class \( \overline{c} \).
A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$. Points $(d_1, d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class $c$, and points $(d_1, d_2)$ with $w_1 d_1 + w_2 d_2 < \theta$ are in the complement class $\bar{c}$. 
A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$.

Example for a 2D linear classifier.
A linear classifier in 2D is a line described by the equation $w_1 d_1 + w_2 d_2 = \theta$.

Example for a 2D linear classifier:

Points $(d_1, d_2)$ with $w_1 d_1 + w_2 d_2 \geq \theta$ are in the class $c$. 
A linear classifier in 2D

- A linear classifier in 2D is a line described by the equation \( w_1 d_1 + w_2 d_2 = \theta \).
- Example for a 2D linear classifier.
- Points \((d_1, d_2)\) with \( w_1 d_1 + w_2 d_2 \geq \theta \) are in the class \( c \).
- Points \((d_1, d_2)\) with \( w_1 d_1 + w_2 d_2 < \theta \) are in the complement class \( \overline{c} \).
A linear classifier in 3D

- A linear classifier in 3D is a plane described by the equation
  \[ w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta \]
A linear classifier in 3D

- A linear classifier in 3D is a plane described by the equation:
  \[ w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta \]
- Example for a 3D linear classifier
A linear classifier in 3D

- A linear classifier in 3D is a plane described by the equation
  \[ w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta \]
- Example for a 3D linear classifier
- Points \((d_1\ d_2\ d_3)\) with \(w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta\) are in the class \(c\).
A linear classifier in 3D

- A linear classifier in 3D is a plane described by the equation:
  \[ w_1 d_1 + w_2 d_2 + w_3 d_3 = \theta \]
- Example for a 3D linear classifier:
- Points \((d_1 \ d_2 \ d_3)\) with \(w_1 d_1 + w_2 d_2 + w_3 d_3 \geq \theta\) are in the class \(c\).
- Points \((d_1 \ d_2 \ d_3)\) with \(w_1 d_1 + w_2 d_2 + w_3 d_3 < \theta\) are in the complement class \(\bar{c}\).
Naive Bayes and Logistic Regression as linear classifiers

Multinomial Naive Bayes is a linear classifier (in log space) defined by:

\[ \sum_{i=1}^{M} w_i d_i = \theta \]

where \( w_i = \log[\hat{P}(t_i|c)/\hat{P}(t_i|\bar{c})] \), \( d_i = \) number of occurrences of \( t_i \) in \( d \), and \( \theta = -\log[\hat{P}(c)/\hat{P}(\bar{c})] \). Here, the index \( i \), \( 1 \leq i \leq M \), refers to terms of the vocabulary.

Logistic regression is the same (we only put it into the logistic function to turn it into a probability).
Naïve Bayes and Logistic Regression as linear classifiers

Multinomial Naïve Bayes is a linear classifier (in log space) defined by:

\[ \sum_{i=1}^{M} w_i d_i = \theta \]

where \( w_i = \log[\hat{P}(t_i|c)/\hat{P}(t_i|\bar{c})] \), \( d_i = \) number of occurrences of \( t_i \) in \( d \), and \( \theta = -\log[\hat{P}(c)/\hat{P}(\bar{c})] \). Here, the index \( i \), \( 1 \leq i \leq M \), refers to terms of the vocabulary.

Logistic regression is the same (we only put it into the logistic function to turn it into a probability).

**Takeway**

Naïve Bayes, logistic regression and SVM are all linear methods. They choose their hyperplanes based on different objectives: joint likelihood (NB), conditional likelihood (LR), and the margin (SVM).
Which hyperplane?
Which hyperplane?

- For linearly separable training sets: there are infinitely many separating hyperplanes.
- They all separate the training set perfectly . . .
- . . . but they behave differently on test data.
- Error rates on new data are low for some, high for others.
- How do we find a low-error separator?
Plan

Decision Trees

Learning Decision Trees

Vector space classification

Linear Classifiers

Support Vector Machines

Recap
• Machine-learning research in the last two decades has improved classifier effectiveness.

• New generation of state-of-the-art classifiers: support vector machines (SVMs), boosted decision trees, regularized logistic regression, neural networks, and random forests

• Applications to IR problems, particularly text classification
Support Vector Machines

- 2-class training data
Support Vector Machines

- 2-class training data
- decision boundary → linear separator
Support Vector Machines

- 2-class training data
- decision boundary \(\rightarrow\) **linear separator**
- criterion: being maximally far away from any data point \(\rightarrow\) determines classifier **margin**

Margin is maximized
Support Vector Machines

- 2-class training data
- decision boundary → **linear separator**
- criterion: being maximally far away from any data point → determines classifier **margin**
- linear separator position defined by **support vectors**

```
Support vectors
Maximum margin
decision hyperplane
```

Margin is maximized

Why maximize the margin?
Points near decision surface are uncertain classification decisions. A classifier with a large margin makes no low certainty classification decisions. Provides classification safety margin w.r.t slight errors in measurement or document variation.
Why maximize the margin?

- Points near decision surface → uncertain classification decisions
- A classifier with a large margin is always confident
- Gives classification safety margin (measurement or variation)
Why maximize the margin?

- SVM classifier: large margin around decision boundary
- compare to decision hyperplane: place fat separator between classes
  - unique solution
- decreased memory capacity
- increased ability to correctly generalize to test data
Walkthrough example: building an SVM over the data shown

Working geometrically:

- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between (1, 1) and (2, 3), giving a weight vector of (1, 2).
- The optimal decision surface is orthogonal to that line and intersects it at the halfway point. Therefore, it passes through (1, 1.5).
- The SVM decision boundary is:
  \[ 0 = 1 \cdot 2 \cdot x + y \cdot 114, \]
  \[ 0 = 2 \cdot 5 \cdot x + 4 \cdot 5 \cdot y \cdot 115 \]
Walkthrough example: building an SVM over the data shown

Working geometrically:

- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between (1, 1) and (2, 3), giving a weight vector of (1, 2).
Walkthrough example: building an SVM over the data shown

Working geometrically:

- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between (1, 1) and (2, 3), giving a weight vector of (1, 2).
- The optimal decision surface is orthogonal to that line and intersects it at the halfway point. Therefore, it passes through (1.5, 2).
Walkthrough example: building an SVM over the data shown

Working geometrically:

- The maximum margin weight vector will be parallel to the shortest line connecting points of the two classes, that is, the line between (1, 1) and (2, 3), giving a weight vector of (1, 2).
- The optimal decision surface is orthogonal to that line and intersects it at the halfway point. Therefore, it passes through (1.5, 2).
- The SVM decision boundary is:

\[
0 = \frac{1}{2}x + y - \frac{11}{4} \iff 0 = \frac{2}{5}x + \frac{4}{5}y - \frac{11}{5}
\]
SVM extensions

- Slack variables: not perfect line
- Kernels: different geometries

- Loss functions: Different penalties for getting the answer wrong
Recap

Plan

Decision Trees
Learning Decision Trees
Vector space classification
Linear Classifiers
Support Vector Machines
Recap
Text classification

- Many commercial applications
- There are many applications of text classification for corporate Intranets, government departments, and Internet publishers.
- Often greater performance gains from exploiting domain-specific text features than from changing from one machine learning method to another. (Homework 2)
Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None?
- Very little?
- A fair amount?
- A huge amount
Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little?
- A fair amount?
- A huge amount
Choosing what kind of classifier to use

When building a text classifier, first question: how much training data is there currently available?

• None? Hand write rules or use active learning
• Very little? Naïve Bayes
• A fair amount?
• A huge amount
Choosing what kind of classifier to use

When building a text classifier, first question: how much training data is there currently available?

- None? Hand write rules or use active learning
- Very little? Naïve Bayes
- A fair amount? SVM
- A huge amount
Choosing what kind of classifier to use

When building a text classifier, first question: **how much training data is there currently available?**

- None? **Hand write rules or use active learning**
- Very little? **Naïve Bayes**
- A fair amount? **SVM**
- A huge amount **Doesn’t matter, use whatever works**
Recap

• Is there a learning method that is optimal for all text classification problems?
• No, because there is a tradeoff between bias and variance.
• Factors to take into account:
  ○ How much training data is available?
  ○ How simple/complex is the problem? (linear vs. nonlinear decision boundary)
  ○ How noisy is the problem?
  ○ How stable is the problem over time?
    • For an unstable problem, it’s better to use a simple and robust classifier.
    • You’ll be investigating the role of features in HW2!