Road Map

- Basic concepts
- Decision tree induction
- Evaluation of classifiers
- Rule induction
- Classification using association rules
- Naïve Bayesian classification
- Naïve Bayes for text classification
- Support vector machines
- K-nearest neighbor
- Ensemble methods: Bagging and Boosting
- Summary
Road Map

- Basic concepts
- Decision tree induction
- Evaluation of classifiers
- Rule induction
- Classification using association rules
- Naïve Bayesian classification
- Naïve Bayes for text classification
- Support vector machines
- K-nearest neighbor
- Ensemble methods: Bagging and Boosting
- Summary
Introduction

- Support vector machines were invented by V. Vapnik and his co-workers in 1970s in Russia and became known to the West in 1992.
- SVMs are linear classifiers that find a hyperplane to separate two class of data, positive and negative.
- Kernel functions are used for nonlinear separation.
- SVM not only has a rigorous theoretical foundation, but also performs classification more accurately than most other methods in applications, especially for high dimensional data.
- It is perhaps the best classifier for text classification.
Basic concepts

Let the set of training examples $D$ be

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_r, y_r)\},$$

where $x_i = (x_1, x_2, \ldots, x_n)$ is an input vector in a real-valued space $X \subseteq \mathbb{R}^n$ and $y_i$ is its class label (output value), $y_i \in \{1, -1\}$.

1: positive class and -1: negative class.

SVM finds a linear function of the form ($w$: weight vector)

$$f(x) = \langle w \cdot x \rangle + b$$

$$y_i = \begin{cases} 
1 & \text{if} \langle w \cdot x_i \rangle + b \geq 0 \\
-1 & \text{if} \langle w \cdot x_i \rangle + b < 0 
\end{cases}$$
The hyperplane

- The hyperplane that separates positive and negative training data is
  \[ \langle w \cdot x \rangle + b = 0 \]
- It is also called the decision boundary (surface).
- So many possible hyperplanes, which one to choose?
Maximal margin hyperplane

- SVM looks for the separating hyperplane with the largest margin.
- Machine learning theory says this hyperplane minimizes the error bound.
Linear SVM: separable case

- Assume the data are linearly separable.
- Consider a positive data point \((x^+, 1)\) and a negative \((x^-, -1)\) that are closest to the hyperplane
  \[<w \cdot x> + b = 0.\]
- We define two parallel hyperplanes, \(H_+\) and \(H_-\), that pass through \(x^+\) and \(x^-\) respectively. \(H_+\) and \(H_-\) are also parallel to \(<w \cdot x> + b = 0.\)

\[
\begin{align*}
H_+ & : <w \cdot x^+> + b = 1 \\
H_- & : <w \cdot x^-> + b = -1 \\
\text{such that} & : <w \cdot x_i> + b \geq 1 \quad \text{if } y_i = 1 \\
& : <w \cdot x_i> + b \leq -1 \quad \text{if } y_i = -1,
\end{align*}
\]
Compute the margin

- Now let us compute the distance between the two margin hyperplanes $H_+$ and $H_-$. Their distance is the **margin** ($d_+ + d_-$ in the figure).

- Recall from vector space in algebra that the (perpendicular) **distance** from a point $x_i$ to the hyperplane $\langle w \cdot x \rangle + b = 0$ is:

$$\frac{|\langle w \cdot x_i \rangle + b|}{||w||}$$

where $||w||$ is the norm of $w$,

$$||w|| = \sqrt{\langle w \cdot w \rangle} = \sqrt{w_1^2 + w_2^2 + \ldots + w_n^2}$$
Linear SVM: Non-separable case

- Linear separable case is the ideal situation.
- Real-life data may have noise or errors.
  - Class label incorrect or randomness in the application domain.
- With noisy data, the constraints may not be satisfied. Then, no solution!
Geometric interpretation

- Two error data points $x_a$ and $x_b$ (circled) in wrong regions
How to deal with nonlinear separation?

- The SVM formulations require linear separation.
- Real-life data sets may need nonlinear separation.
- To deal with nonlinear separation, the same formulation and techniques as for the linear case are still used.
- We only transform the input data into another space (usually of a much higher dimension) so that
  - a linear decision boundary can separate positive and negative examples in the transformed space,
- The transformed space is called the feature space. The original data space is called the input space.
Geometric interpretation

In this example, the transformed space is also 2-D. But usually, the number of dimensions in the feature space is much higher than that in the input space.
Some other issues in SVM

- SVM works only in a real-valued space.
  - For a categorical attribute, we need to convert its categorical values to numeric values.

- SVM does only two-class classification.
  - For multi-class problems, some strategies can be applied, e.g., one-against-rest, and error-correcting output coding.

- The hyperplane produced by SVM is hard to understand by human users.
  - SVM is commonly used in applications that do not required human understanding.
Road Map

- Basic concepts
- Decision tree induction
- Evaluation of classifiers
- Rule induction
- Classification using association rules
- Naïve Bayesian classification
- Naïve Bayes for text classification
- Support vector machines
- K-nearest neighbor
- Ensemble methods: Bagging and Boosting
- Summary
k-Nearest Neighbor Classification

- Unlike all the previous learning methods, kNN does not build model from the training data.
- To classify a test instance $d$, define $k$-neighborhood $P$ as $k$ nearest neighbors of $d$
- Count number $n$ of training instances in $P$ that belong to class $c_j$
- Estimate $\Pr(c_j|d)$ as $n/k$
- No training is needed. Classification time is linear in training set size for each test case.
**kNN Algorithm**

Algorithm $\text{kNN}(D, d, k)$

1. Compute the distance between $d$ and every example in $D$;
2. Choose the $k$ examples in $D$ that are nearest to $d$, denote the set by $P (\subseteq D)$;
3. Assign $d$ the class that is the most frequent class in $P$ (or the majority class);

- $k$ is usually chosen empirically via a validation set or cross-validation by trying a range of $k$ values.

- **Distance function** is crucial, but depends on applications.
Example: k=6 (6NN)

- Government
- Science
- Arts

A new point
Pr(science| )?
Discussions

- kNN can deal with complex and arbitrary decision boundaries.
- Despite its simplicity, researchers have shown that the classification accuracy of kNN can be quite strong and in many cases as accurate as those elaborated methods.
- kNN is slow at the classification time
- kNN does not produce an understandable model
Road Map

- Basic concepts
- Decision tree induction
- Evaluation of classifiers
- Rule induction
- Classification using association rules
- Naïve Bayesian classification
- Naïve Bayes for text classification
- Support vector machines
- K-nearest neighbor
- Ensemble methods: Bagging and Boosting
- Summary
Combining classifiers

- So far, we have only discussed individual classifiers, i.e., how to build them and use them.
- Can we combine multiple classifiers to produce a better classifier?
- Yes, sometimes
- We discuss two main algorithms:
  - Bagging
  - Boosting
Bagging

- Breiman, 1996

**Bootstrap Aggregating** = Bagging

- Application of bootstrap sampling
  - **Given:** set $D$ containing $m$ training examples
  - Create a sample $S[i]$ of $D$ by drawing $m$ examples at random *with replacement* from $D$
  - $S[i]$ of size $m$: expected to leave out 0.37 of examples from $D$
Bagging (cont…)

- **Training**
  - Create \( k \) bootstrap samples \( S[1], S[2], \ldots, S[k] \)
  - Build a distinct classifier on each \( S[i] \) to produce \( k \) classifiers, using the same learning algorithm.

- **Testing**
  - Classify each new instance by voting of the \( k \) classifiers (equal weights)
### Bagging Example

<table>
<thead>
<tr>
<th>Original</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set 1</td>
<td>2</td>
<td>7</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Training set 2</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>Training set 3</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>7</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Training set 4</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>8</td>
</tr>
</tbody>
</table>
Bagging (cont ...)

- When does it help?
  - When learner is unstable
    - Small change to training set causes large change in the output classifier
    - True for decision trees, neural networks; not true for $k$-nearest neighbor, naïve Bayesian, class association rules
  - Experimentally, bagging can help substantially for unstable learners, may somewhat degrade results for stable learners

Bagging Predictors, Leo Breiman, 1996
Boosting

- A family of methods:
  - We only study **AdaBoost** (Freund & Schapire, 1996)

- **Training**
  - Produce a sequence of classifiers (the same base learner)
  - Each classifier is dependent on the previous one, and focuses on the previous one’s errors
  - Examples that are incorrectly predicted in previous classifiers are given higher weights

- **Testing**
  - For a test case, the results of the series of classifiers are combined to determine the final class of the test case.
AdaBoost

**Weighted training set**

\[(x_1, y_1, w_1)\]
\[(x_2, y_2, w_2)\]
\[
\cdots
\]
\[(x_n, y_n, w_n)\]

Non-negative weights sum to 1

Change weights

**called a weaker classifier**

- Build a classifier \( h_t \)
  whose accuracy on training set \( > \frac{1}{2} \)
  (better than random)
AdaBoost algorithm

Algorithm AdaBoost.M1

Input: sequence of $m$ examples $\{(x_1, y_1), \ldots, (x_m, y_m)\}$
with labels $y_e \in Y = \{1, \ldots, k\}$
weak learning algorithm WeakLearn
integer $T$ specifying number of iterations

Initialize $D_1(x) = 1/m$ for all $i$.

Do for $t = 1, 2, \ldots, T$:

1. Call WeakLearn, providing it with the distribution $D_t$.
2. Get back a hypothesis $h_t : X \rightarrow Y$.
3. Calculate the error of $h_t$: $\epsilon_t = \sum_{x : h_t(x) \neq y} D_t(x)$.

If $\epsilon_t > 1/2$, then set $T = t - 1$ and abort loop.

4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$.
5. Update distribution $D_t$:

$$D_{t+1}(x) = \frac{D_t(x)}{Z_t} \times \begin{cases} 
\beta_t & \text{if } h_t(x) = y \\
1 & \text{otherwise}
\end{cases}$$

where $Z_t$ is a normalization constant (chosen so that $D_{t+1}$ will be a distribution).

Output the final hypothesis:

$$h_{\text{final}}(x) = \arg \max_{y \in Y} \sum_{t : h_t(x) = y} \log \frac{1}{\beta_t}.$$
Bagging, Boosting and C4.5

**C4.5’s mean error rate over the 10 cross-validation.**

**Bagged C4.5 vs. C4.5.**

**Boosted C4.5 vs. C4.5.**

**Boosting vs. Bagging**

<table>
<thead>
<tr>
<th></th>
<th>C4.5</th>
<th>Bagged C4.5 vs C4.5</th>
<th>Boosted C4.5 vs C4.5</th>
<th>Boosting vs Bagging</th>
</tr>
</thead>
<tbody>
<tr>
<td>err (%)</td>
<td></td>
<td>w-l</td>
<td>ratio</td>
<td>w-l</td>
</tr>
<tr>
<td>anneal</td>
<td>7.67</td>
<td>6.25</td>
<td>.814</td>
<td>4.73</td>
</tr>
<tr>
<td>audiology</td>
<td>22.12</td>
<td>19.29</td>
<td>.872</td>
<td>15.71</td>
</tr>
<tr>
<td>auto</td>
<td>17.66</td>
<td>19.66</td>
<td>1.113</td>
<td>15.22</td>
</tr>
<tr>
<td>breast-w</td>
<td>5.28</td>
<td>4.23</td>
<td>.802</td>
<td>4.09</td>
</tr>
<tr>
<td>chess</td>
<td>8.55</td>
<td>8.33</td>
<td>.975</td>
<td>4.59</td>
</tr>
<tr>
<td>colic</td>
<td>14.92</td>
<td>15.19</td>
<td>1.018</td>
<td>18.83</td>
</tr>
<tr>
<td>credit-a</td>
<td>14.70</td>
<td>14.13</td>
<td>.962</td>
<td>15.64</td>
</tr>
<tr>
<td>credit-g</td>
<td>28.84</td>
<td>25.81</td>
<td>.908</td>
<td>29.14</td>
</tr>
<tr>
<td>diabetes</td>
<td>25.39</td>
<td>23.63</td>
<td>.931</td>
<td>28.18</td>
</tr>
<tr>
<td>glass</td>
<td>32.48</td>
<td>27.01</td>
<td>.832</td>
<td>23.55</td>
</tr>
<tr>
<td>heart-c</td>
<td>22.94</td>
<td>21.52</td>
<td>.938</td>
<td>21.39</td>
</tr>
<tr>
<td>heart-h</td>
<td>21.53</td>
<td>20.31</td>
<td>.943</td>
<td>21.05</td>
</tr>
<tr>
<td>hepatitis</td>
<td>20.39</td>
<td>18.52</td>
<td>.908</td>
<td>17.68</td>
</tr>
<tr>
<td>hypo</td>
<td>.48</td>
<td>.45</td>
<td>.928</td>
<td>.36</td>
</tr>
<tr>
<td>iris</td>
<td>4.80</td>
<td>5.13</td>
<td>1.069</td>
<td>6.53</td>
</tr>
<tr>
<td>labor</td>
<td>19.12</td>
<td>14.39</td>
<td>.752</td>
<td>13.86</td>
</tr>
<tr>
<td>letter</td>
<td>11.99</td>
<td>7.51</td>
<td>.626</td>
<td>4.66</td>
</tr>
<tr>
<td>lymphography</td>
<td>21.69</td>
<td>20.41</td>
<td>.941</td>
<td>17.43</td>
</tr>
<tr>
<td>phoneme</td>
<td>19.44</td>
<td>18.73</td>
<td>.964</td>
<td>16.36</td>
</tr>
<tr>
<td>segment</td>
<td>3.21</td>
<td>2.74</td>
<td>.853</td>
<td>1.87</td>
</tr>
<tr>
<td>sick</td>
<td>1.34</td>
<td>1.22</td>
<td>.907</td>
<td>1.05</td>
</tr>
<tr>
<td>sonar</td>
<td>25.62</td>
<td>23.80</td>
<td>.929</td>
<td>19.62</td>
</tr>
<tr>
<td>soybean</td>
<td>7.73</td>
<td>7.58</td>
<td>.981</td>
<td>7.16</td>
</tr>
<tr>
<td>splice</td>
<td>5.91</td>
<td>5.58</td>
<td>.943</td>
<td>5.43</td>
</tr>
<tr>
<td>vehicle</td>
<td>27.09</td>
<td>25.54</td>
<td>.943</td>
<td>22.72</td>
</tr>
<tr>
<td>vote</td>
<td>5.06</td>
<td>4.37</td>
<td>.864</td>
<td>5.29</td>
</tr>
<tr>
<td>waveform</td>
<td>27.33</td>
<td>19.77</td>
<td>.723</td>
<td>18.53</td>
</tr>
</tbody>
</table>

**average**

<table>
<thead>
<tr>
<th>C4.5</th>
<th>Bagged C4.5 vs C4.5</th>
<th>Boosted C4.5 vs C4.5</th>
<th>Boosting vs Bagging</th>
</tr>
</thead>
<tbody>
<tr>
<td>15.66</td>
<td>14.11</td>
<td>13.36</td>
<td>.847</td>
</tr>
</tbody>
</table>

---

Spring 2008  Web Mining Seminar  156
Does AdaBoost always work?

- The actual performance of boosting depends on the data and the base learner.
  - It requires the base learner to be unstable as bagging.
- Boosting seems to be susceptible to noise.
  - When the number of outliers is very large, the emphasis placed on the hard examples can hurt the performance.
Road Map

- Basic concepts
- Decision tree induction
- Evaluation of classifiers
- Rule induction
- Classification using association rules
- Naïve Bayesian classification
- Naïve Bayes for text classification
- Support vector machines
- K-nearest neighbor
- Ensemble methods: Bagging and Boosting

Summary
Summary

- Applications of supervised learning are in almost any field or domain.
- We studied many classification techniques.
- There are still many other methods, e.g.,
  - Bayesian networks
  - Neural networks
  - Genetic algorithms
  - Fuzzy classification
  This large number of methods also show the importance of classification and its wide applicability.
- It remains an active research area.