Instance-Based Classifiers

Set of Stored Cases

<table>
<thead>
<tr>
<th>Atri</th>
<th>ArtN</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

- Store the training records
- Use training records to predict the class label of unseen cases

Instance-Based Classifiers

Examples:
- Rote-learner
  - Memorizes entire training data and performs classification only if attributes of record match one of the training examples exactly
- Nearest neighbor
  - Uses k "closest" points (nearest neighbors) for performing classification

Nearest Neighbor Classifiers

Basic idea:
- If it walks like a duck, quacks like a duck, then it's probably a duck

Nearest-Neighbor Classifiers

Requires three things
- The set of stored records
- Distance Metric to compute distance between records
- The value of k, the number of nearest neighbors to retrieve

To classify an unknown record:
- Compute distance to other training records
- Identify k nearest neighbors
- Use class labels of nearest neighbors to determine the class label of unknown record (e.g., by taking majority vote)

Definition of Nearest Neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x
1 nearest-neighbor

Voronoi Diagram

Nearest Neighbor Classification

- Compute distance between two points:
  - Euclidean distance
  \[ d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \]
  - Determine the class from nearest neighbor list
    - take the majority vote of class labels among the k-nearest neighbors
    - Weigh the vote according to distance
      - weight factor, \( w = \frac{1}{d^2} \)

Choosing the value of k:
- If k is too small, sensitive to noise points
- If k is too large, neighborhood may include points from other classes

Scaling issues
- Attributes may have to be scaled to prevent distance measures from being dominated by one of the attributes
- Example:
  - height of a person may vary from 1.5m to 1.8m
  - weight of a person may vary from 90lb to 300lb
  - income of a person may vary from $10K to $1M

How to find the right distance measure for a particular application? May require domain knowledge.
- Recent research on learning the distance measure from data.
- k is "complexity parameter" of kNN: select via cross-validation.
- Beware of the curse of dimensionality: in high dimensional space, the nearest neighbour is very far away.

k-NN classifiers are lazy learners
- It does not build models explicitly
- Unlike eager learners such as decision tree induction and rule-based systems
- Classifying unknown records is relatively expensive
Example: PEBLS

- PEBLS: Parallel Examplar-Based Learning System (Cost & Salzberg)
  - Works with both continuous and nominal features
    - For nominal features, distance between two nominal values is computed using modified value difference metric (MVDM)
  - Each record is assigned a weight factor
  - Number of nearest neighbor, \( k = 1 \)

Modified value difference metric for nominal attributes:

The distance between two attribute values is determined by the difference between their conditional class distributions.

Distance between record \( x \) and record \( y \):

\[
\Delta(x, y) = w_x w_y \sum_{i=1}^{k} d(x_i, y_i)^2
\]

where:

\( w_x = \frac{\text{number of times } x \text{ is used for prediction}}{\text{number of times } x \text{ predicts correctly}} \)

\( w_x = 1 \) if \( X \) makes accurate prediction most of the time

\( w_x > 1 \) if \( X \) is not reliable for making predictions

Example of distance calculation

\( x = \text{(yes, single, 125K)} \quad y = \text{(no, married, 100K)} \)

\[
d(x, y)^2 = d(\text{yes, no})^2 = \left( \frac{6}{7} \right)^2 = 36/49.
\]

\[
d(x, y)^2 = d(\text{single, married})^2 = 1^2 = 1.
\]

\[
d(x, y)^2 = d(125, 100)^2 = 25^2 = 625.
\]

Income dominates distance between objects.

Divide income by its standard deviation 45.63.

\[
d(x, y)^2 = d(2.74, 2.19)^2 = 0.55^2 = 0.3025.
\]
kNN regression

- kNN can also be used for regression problems where we want to predict a numeric variable rather than a class label.
- Rather than taking the majority vote, we take the average $y$ value of the $k$ nearest neighbors as the predicted value for a new case.
- Direct attempt to estimate $E(y \mid x)$, the average $y$ value at $x$. This average would be the optimal prediction (minimizes mean squared error).

kNN regression: Example

In a clinical study of risk-factors for cardiovascular disease,
- the independent variable $x$ is a patient’s waist circumference
- the dependent variable $y$ is a patient’s deep abdominal adipose tissue (more complicated to measure).

The researchers want to predict the amount of deep abdominal adipose tissue from a simple measurement of waist circumference.

Adipose Tissue Data

Data of 109 men between 18 and 42.

Example: data points

Example: 8 consecutive points from the original data set.

Subset for example

<table>
<thead>
<tr>
<th>$i$</th>
<th>$(x_i, y_i)$</th>
<th>$i$</th>
<th>$(x_i, y_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(68.85, 55.78)</td>
<td>5</td>
<td>(73.10, 38.21)</td>
</tr>
<tr>
<td>2</td>
<td>(71.85, 21.68)</td>
<td>6</td>
<td>(73.20, 32.22)</td>
</tr>
<tr>
<td>3</td>
<td>(71.90, 28.32)</td>
<td>7</td>
<td>(73.80, 43.35)</td>
</tr>
<tr>
<td>4</td>
<td>(72.60, 25.89)</td>
<td>8</td>
<td>(74.15, 33.41)</td>
</tr>
</tbody>
</table>

For $k=2$ ($k=5$) and $x=73$ we get the prediction:

$\hat{y}_2(x = 73) = \frac{38.21 + 32.22}{2} = 35.215$

$\hat{y}_5(x = 73) = \frac{38.21 + 32.22 + 43.35 + 25.89 + 28.32}{5} = 33.598$

kNN regression with $k=1$

With $k=1$ the regression function shows rather “wild” fluctuations.
kNN regression with $k=20$

With $k=20$ the regression function is much more "smooth".

kNN regression with $k=109$

Ensemble Methods

- Construct a set of classifiers from the training data
- Predict class label of previously unseen records by aggregating predictions made by multiple classifiers

General Idea

Step 1: Create Multiple Data Sets
Step 2: Build Multiple Classifiers
Step 3: Combine Classifiers

Why could it work?

- Consider a binary classification problem.
- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - Assume classifiers are independent
  - Take majority vote as ensemble prediction.
  - Probability that 13 out of 25 classifiers make a wrong prediction (binomial distribution):
    \[
    \binom{25}{13} 0.35^{13}0.65^{12} \approx 0.035
    \]

Why could it work?

The ensemble makes an error when the majority of the base classifiers is wrong, so probability of error of ensemble is:

\[
\sum_{i=13}^{25} \binom{25}{i} 0.35^i0.65^{25-i} \approx 0.06
\]

In practice it won’t work this well, because typically base classifiers are correlated.
Bias and Variance components of error

The expected error of a classifier can be decomposed into three sources of error:

1. Bias: is model "correct" on average?
2. Variance: how sensitive is the model to the particular training sample?

Here the average and variance is taken with respect to repeatedly drawing training samples (of some fixed size) from the population and constructing a classifier on that training sample.

Error decomposition

\[ \hat{P}(y = 1 | x) \]

Unavoidable error: 0.2.
Bias: 0.7 - 0.8 = -0.1.

Illustration of bias

Black: true optimal decision boundary.
Blue: average decision boundary of a linear classifier.
Red: average decision boundary of some more "flexible" classifier.

Bias and Variance Components of Error

- Usually classifiers with low bias have high variance and vice versa: there is a trade-off (bias-variance dilemma). E.g. a classifier that creates a linear decision boundary may have high bias, but has low variance.
- As the size of the training sample increases
  - Variance goes down
  - Bias is unaffected
- So with larger training samples, we can “afford” more complex models (bias component of error starts to dominate).

Examples of Ensemble Methods

- How to generate an ensemble of classifiers?
  - Bagging (bootstrap aggregating).
  - Boosting
Bagging

- Sample N times from data with replacement
- Build classifier on each bootstrap sample
- Each data point has probability $1 - (1 - 1/N)^N$ of being selected
- Mimics drawing repeated samples from the population.

Bagging: Example

<table>
<thead>
<tr>
<th>Original data</th>
<th>Bootstrap sample 1</th>
<th>Bootstrap sample 2</th>
<th>Bootstrap sample 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1.76,1)</td>
<td>(1.76,1)</td>
<td>(1.76,1)</td>
<td>(1.73,0)</td>
</tr>
<tr>
<td>(1.84,1)</td>
<td>(1.76,1)</td>
<td>(2.01,1)</td>
<td>(1.69,0)</td>
</tr>
<tr>
<td>(1.69,0)</td>
<td>(2.01,1)</td>
<td>(1.76,1)</td>
<td>(1.73,0)</td>
</tr>
<tr>
<td>(1.82,1)</td>
<td>(1.82,1)</td>
<td>(1.76,1)</td>
<td>(1.73,0)</td>
</tr>
<tr>
<td>(2.01,1)</td>
<td>(2.01,1)</td>
<td>(1.69,0)</td>
<td>(1.82,1)</td>
</tr>
<tr>
<td>(1.73,0)</td>
<td>(1.76,1)</td>
<td>(1.82,1)</td>
<td>(1.69,0)</td>
</tr>
</tbody>
</table>

Use 1-nearest neighbor as the base classifier.

Bagging: example

Example classifications:
- $x = 1.77$: $C_1(x) = 1$, $C_2(x) = 1$ and $C_3(x) = 0$. $C'(x) = 1$.
- $x = 1.69$: $C_1(x) = 1$, $C_2(x) = 0$ and $C_3(x) = 0$. $C'(x) = 0$.
- $x = 1.90$: $C_1(x) = 1$, $C_2(x) = 1$ and $C_3(x) = 1$. $C'(x) = 1$.

$C'(x)$ takes the majority vote of the base classifiers.

Bagging

- Is mainly a variance reduction technique.
- Hence, it can reduce the error of high variance classifiers, such as decision trees or rule-based classifiers.
- For biased classifiers with low variance bagging may even be harmful.
- Combining base classifiers will reduce the interpretability of the model (“we take a majority vote of these ten trees”).

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

Example 4 is hard to classify
- Its weight is increased; therefore it is more likely to be chosen again in subsequent rounds
In boosting the base classifier can be a very simple one. For example: decision stump, a decision tree that only makes one split. Even if the base classifier is slightly better than random guessing, boosting can drive the training error to zero.

Example: AdaBoost

Draw a sample of size N using weights (initially each data point has weight 1/N) to get data set D_i. Use D_i to train classifier C_i.

Weighted error of base classifier C_i:

\[ \varepsilon_i = \frac{1}{N} \sum_{j=1}^{N} \delta(C_i(x_j) \neq y_j) \]

Importance of classifier C_i:

\[ \alpha_i = \ln \left( \frac{1 - \varepsilon_i}{\varepsilon_i} \right) \]

Illustrating Adaboost

Original data (sorted on x):

We have 10 data points, so each data point gets initial weight 1/10. Suppose we sample the following six (to simplify computations) points:

The optimal decision stump is:
Illustrating Adaboost

The updated weights are:

\[
\begin{array}{ccccccccccc}
+ & + & + & - & - & - & + & + \\
0.0625 & 0.25 \\
\end{array}
\]

Suppose the new sample is:

\[
\begin{array}{ccccccccccc}
+ & + & + & - & - & - & - & - & + & + \\
- & + & - & + & - & - & + & - & + & -
\end{array}
\]

Now we make 3 errors, each with weight 0.0625, so

\[
\epsilon_2 = 3 \times 0.0625 = 0.1875
\]

Hence

\[
\alpha_2 = \ln \left( \frac{1 - \epsilon_2}{\epsilon_2} \right) = \ln 4.33 = 1.47
\]

The new weights are:

\[
\begin{array}{cccc}
+ & + & + & - & - & - & + & + \\
0.17 & 0.039 & 0.15 \\
\end{array}
\]

Illustrating Adaboost

New sample and decision stump:

\[
\begin{array}{ccccccccccc}
+ & + & + & - & - & - & - & - & + & + \\
+ & + & + & - & - & - & - & - & + & + \\
\end{array}
\]

\[
\epsilon_3 = 5 \times 0.039 = 0.195
\]

\[
\alpha_3 = \ln \left( \frac{0.805}{0.195} \right) = \ln 4.13 = 1.42
\]

Combining the classifiers:

\[
\begin{align*}
C_1 : & \quad + + + - - - - - - & \quad \alpha_1 = 1.38 \\
C_2 : & \quad - - - - - - - + + + + & \quad \alpha_2 = 1.47 \\
C_3 : & \quad + + + + + + + + + + & \quad \alpha_3 = 1.42 \\
C^* : & \quad + + + - - - - - - & \quad \text{combined}
\end{align*}
\]