K Nearest Neighbors

3-16-16
Final AlphaGo update

Match result: AlphaGo 4 - Lee Sedol 1

Korea Baduk Association named AlphaGo an honorary 9-dan professional.

Google will donate the $1M prize to UNICEF and charities for STEM and go education.
k nearest neighbors: the setting

- Supervised learning (we know the correct output for each test point).
- Classification (outputs are discrete: class labels).
- Inputs can be continuous or discrete.
k nearest neighbors: the algorithm

Training:
- Store all of the test points and their labels.
  - Can use a data structure like a kd-tree that speeds up localized lookup.

Prediction:
- Find the k training inputs closest to the test input.
- Output the most common label among them.
KNN implementation decisions (and possible answers)

How should we measure distance?
● Euclidean distance between input vectors.

What if there’s a tie for the nearest points?
● Include all points that are tied.

What if there’s a tie for the most-common label?
● Remove the most-distant point until a plurality is achieved.

What if there’s a tie for both?
● We need some arbitrary tie-breaking rule.
Weighted nearest neighbors

Idea: closer points should matter more.

Solution: weight the vote by \[ \frac{1}{\text{distance} + c} \]

Instead of contributing one vote for its label, each neighbor contributes \[ \frac{1}{\text{distance} + c} \] votes for its label.
Why do we even need k neighbors?

Idea: if we’re weighting by distance, we can give all training points a vote.

- Points that are far away will just have really small weight.

Why might this be a bad idea?

- Slow: we have to sum over every point in the training set.
- If we’re using a kd-tree, we can get the neighbors quickly and sum over a small set.
We can use the same ideas for regression.

Locally-weighted regression setting:

- Supervised learning (we know the correct output for each test point).
- Regression (outputs are continuous).
Locally-weighted average

Instead of taking a majority vote, average the y-values.

We could average over the k nearest neighbors.

We could weight the average by distance.

\[ w_i \equiv \frac{1}{d(x_q, x_i)^2} \]

\[ \hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i} \]

Better yet, do both.
Locally-weighted (linear) regression

Least squares linear regression solves the following problem:

Select weights $w_0, \ldots, w_D$ for each dimension to minimize squared error:

$$E = \sum_{x \in \text{training set}} \left( \hat{f}(x) - f(x) \right)^2$$

$$\hat{f}(x) = w_0 + w_1 x_1 + \ldots + w_D x_D$$

Instead, we can minimize distance-weighted squared error:

$$E = \sum_{x \in \text{training set}} \frac{\left( \hat{f}(x) - f(x) \right)^2}{\text{distance}(x) + c}$$
The curse of dimensionality

Having more data attributes seems like it should help, but...

k nearest neighbors:
  ● Extraneous dimensions can increase the distance between relevant points.

naive Bayes:
  ● Each dimension adds training work.
  ● Each dimension thins the data set, making us rely more on the prior.
Exercise: label the test point.

<table>
<thead>
<tr>
<th></th>
<th>HI</th>
<th>CS</th>
<th>BR</th>
<th>FF</th>
<th>EA</th>
<th>RS</th>
<th>TB</th>
<th>AC</th>
<th>MX</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>R</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>D</td>
<td>?</td>
<td>y</td>
<td>y</td>
<td>?</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>D</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>?</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>D</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>D</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>D</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>R</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>R</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>D</td>
<td>y</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>y</td>
</tr>
<tr>
<td>R</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>R</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

k = 3

n = -1

y = +1

? = 0

Distance-weighted majority vote