Locally weighted regression

Relevant Readings: Section 8.3 in Mitchell

CS495 - Machine Learning, Fall 2009

Generalizing k-nearest neighbor to continuous outputs

- The version of k-nearest neighbors we have already seen works well for discrete outputs

Generalizing k-nearest neighbor to continuous outputs

- The version of k-nearest neighbors we have already seen works well for discrete outputs
- How would we generalize this to predict continuous outputs? Ideas?

Generalizing k-nearest neighbor to continuous outputs

- The version of k-nearest neighbors we have already seen works well for discrete outputs
- How would we generalize this to predict continuous outputs? Ideas?

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are
- *Regression* means approximating a function

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are
- *Regression* means approximating a function
- This is an instance-based learning method

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are
- *Regression* means approximating a function
- This is an instance-based learning method
- The idea: whenever you want to classify a sample:

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are
- *Regression* means approximating a function
- This is an instance-based learning method
- The idea: whenever you want to classify a sample:
  - Build a local model of the function (using a linear function, quadratic, neural network, etc.)

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are
- *Regression* means approximating a function
- This is an instance-based learning method
- The idea: whenever you want to classify a sample:
  - Build a local model of the function (using a linear function, quadratic, neural network, etc.)
  - Use the model to predict the output value

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are
- *Regression* means approximating a function
- This is an instance-based learning method
- The idea: whenever you want to classify a sample:
  - Build a local model of the function (using a linear function, quadratic, neural network, etc.)
  - Use the model to predict the output value
  - Throw the model away

Locally weighted regression

- *Local* means using nearby points (i.e. a nearest neighbors approach)
- *Weighted* means we value points based upon how far away they are
- *Regression* means approximating a function
- This is an instance-based learning method
- The idea: whenever you want to classify a sample:
  - Build a local model of the function (using a linear function, quadratic, neural network, etc.)
  - Use the model to predict the output value
  - Throw the model away

Locally weighted linear regression

- In the following,
  - x is an instance, D is the set of possible instances

Locally weighted linear regression

- In the following,
  - x is an instance, D is the set of possible instances
  - $a_i(x)$ is the value of the i^{th} attribute

Locally weighted linear regression

- In the following,
  - x is an instance, D is the set of possible instances
  - $a_i(x)$ is the value of the i^{th} attribute
  - The weights w_i form our hypothesis

Locally weighted linear regression

- In the following,
  - x is an instance, D is the set of possible instances
  - $a_i(x)$ is the value of the i^{th} attribute
  - The weights w_i form our hypothesis
  - f is the target function

Locally weighted linear regression

- In the following,
  - x is an instance, D is the set of possible instances
  - $a_i(x)$ is the value of the i^{th} attribute
  - The weights w_i form our hypothesis
  - f is the target function
  - $\hat{f}$ is our approximation to the target function

Locally weighted linear regression

- In the following,
  - x is an instance, D is the set of possible instances
  - $a_i(x)$ is the value of the i^{th} attribute
  - The weights w_i form our hypothesis
  - f is the target function
  - $\hat{f}$ is our approximation to the target function

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose
  - $\Delta w_j = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose
  - $\Delta w_j = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$
  - η is a small number (the learning rate)

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose
  - $\Delta w_j = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$
  - η is a small number (the learning rate)
- Now we adjust it to work with the present situation. Define the error for instance x_q:

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose
  - $\Delta w_j = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$
  - η is a small number (the learning rate)
- Now we adjust it to work with the present situation. Define the error for instance x_q:
  - $E(x_q) = (1/2) \sum_{x \in \text{KNN of } x_q} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2)\sum_{x \in D}(f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose
  - $\Delta w_j = \eta \sum_{x \in D}(f(x) - \hat{f}(x))a_j(x)$
  - η is a small number (the learning rate)
- Now we adjust it to work with the present situation. Define the error for instance x_q:
  - $E(x_q) = (1/2)\sum_{x \in \text{KNN of } x_q}(f(x) - \hat{f}(x))^2 K(d(x_q, x))$
- And the new version of the gradient descent becomes:

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose
  - $\Delta w_j = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$
  - η is a small number (the learning rate)
- Now we adjust it to work with the present situation. Define the error for instance x_q:
  - $E(x_q) = (1/2) \sum_{x \in \text{KNN of } x_q} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$
- And the new version of the gradient descent becomes:
  - $\Delta w_j = \eta \sum_{x \in \text{KNN of } x_q} K(d(x_q, x))(f(x) - \hat{f}(x)) a_j(x)$

Locally weighted linear regression

- In this case, we use a linear model to do the local approximation $\hat{f}$:
  - $\hat{f}(x) = w_0 + w_1 a_1(x) + \cdots + w_n a_n(x)$
- Suppose we aim to minimize the total squared error:
  - $E = (1/2) \sum_{x \in D} (f(x) - \hat{f}(x))^2$
- Recall the gradient descent we used in checkers for this purpose
  - $\Delta w_j = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$
  - η is a small number (the learning rate)
- Now we adjust it to work with the present situation. Define the error for instance x_q:
  - $E(x_q) = (1/2) \sum_{x \in \text{KNN of } x_q} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$
- And the new version of the gradient descent becomes:
  - $\Delta w_j = \eta \sum_{x \in \text{KNN of } x_q} K(d(x_q, x))(f(x) - \hat{f}(x)) a_j(x)$