Linear Regression

4/14/17

Hypothesis Space

Supervised learning

- For every input in the data set, we know the output

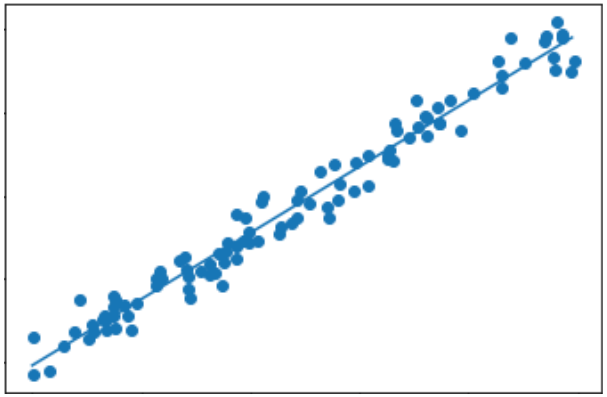
Regression

- Outputs are continuous
  - A number, not a category label

The learned model:

- A linear function mapping input to output
  - A weight for each feature (including bias)

Linear Models



In two dimensions: $f(x) = wx + b$

In *d* dimensions:

$$\vec{x} \equiv \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \qquad f(\vec{x}) = \begin{bmatrix} w_b \\ w_0 \\ \vdots \\ w_d \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_0 \\ \vdots \\ x_d \end{bmatrix}$$

We want to find the linear model that fits our data best.

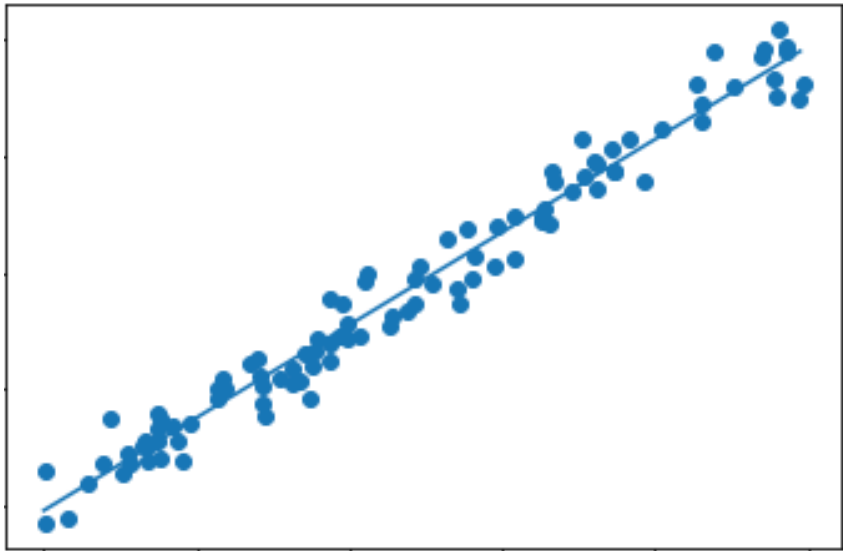
When have we seen a model like this before?

Linear Regression

We want to find the linear model that fits our data best.

Key idea: model data as linear model plus noise. Pick the weights to minimize noise magnitude.

$$f(\vec{x}) = \begin{bmatrix} w_b \\ w_0 \\ \vdots \\ w_d \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_0 \\ \vdots \\ x_d \end{bmatrix} + \epsilon$$



Squared Error

$$f(\vec{x}) = \begin{bmatrix} w_b \\ w_0 \\ \vdots \\ w_d \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_0 \\ \vdots \\ x_d \end{bmatrix} + \epsilon \qquad \hat{f}(\vec{x}) = \begin{bmatrix} w_b \\ w_0 \\ \vdots \\ w_d \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_0 \\ \vdots \\ x_d \end{bmatrix}$$

Define error for a data point to be the squared distance between correct output and predicted output:

$$\left(f(\vec{x}) - \hat{f}(\vec{x})\right)^2 = \epsilon^2$$

Error for the model is the sum of point errors:

$$\sum_{\vec{x} \in \text{data}} \left(y - \hat{f}(\vec{x})\right) = \sum_{\vec{x} \in \text{data}} \epsilon_{\vec{x}}^2$$

Minimizing Squared Error

Goal: pick weights that minimize squared error.

Approach #1: gradient descent

Your reading showed how to do this for 1D inputs:

$$\frac{\partial}{\partial \mathrm{m}} = \frac{2}{N}\sum_{i=1}^{N} - x_i(y_i - (mx_i + b))$$

$$\frac{\partial}{\partial \mathrm{b}} = \frac{2}{N}\sum_{i=1}^{N} - (y_i - (mx_i + b))$$

Minimizing Squared Error

Goal: pick weights that minimize squared error.

Approach #2 (the right way): analytical solution

- The gradient is 0 at the error minimum.
- There is generally a unique global minimum.

$$\vec{w} = \left(\boldsymbol{X}^T \boldsymbol{X}\right)^{-1} \boldsymbol{X}^T \vec{y}$$

$$\boldsymbol{X} \equiv \begin{bmatrix} \vec{x}_0 & \vec{x}_1 \ldots \vec{x}_n \end{bmatrix} \equiv \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_{00} & x_{01} & \ldots & x_{0n} \\ x_{10} & x_{11} & \ldots & x_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{d0} & x_{d1} & \ldots & x_{dn} \end{bmatrix}$$

Change of Basis

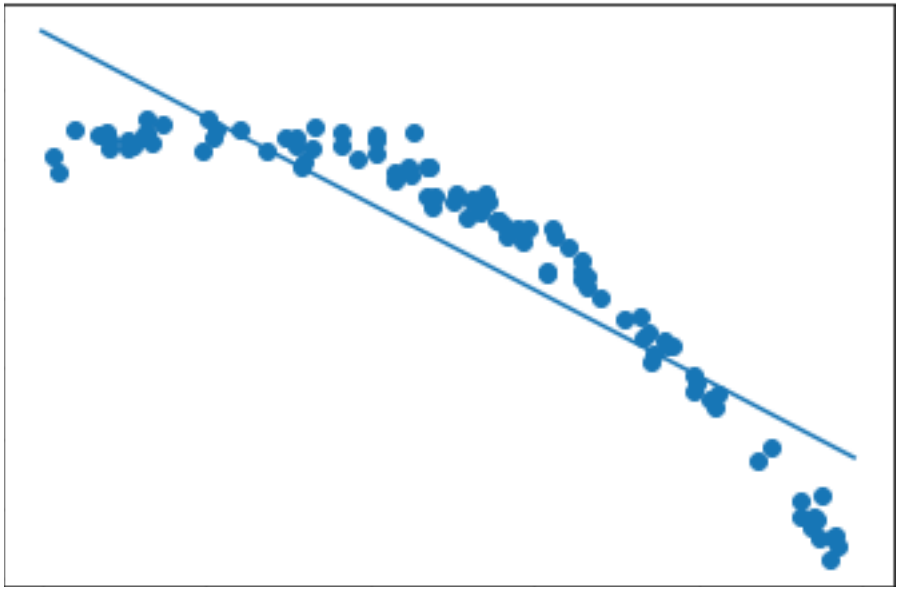
Polynomial regression is just linear regression with a change of basis.

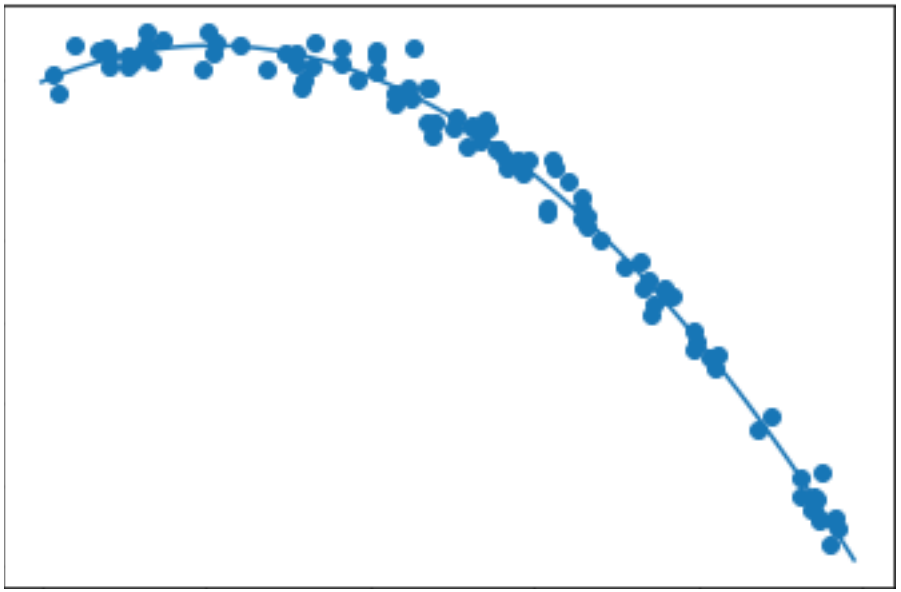
$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \longrightarrow \begin{bmatrix} x_0 \\ (x_0)^2 \\ x_1 \\ (x_1)^2 \\ \vdots \\ x_d \\ (x_d)^2 \end{bmatrix} \text{quadratic basis}$$

$$\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} \longrightarrow \begin{bmatrix} x_0 \\ (x_0)^2 \\ (x_0)^3 \\ x_1 \\ (x_1)^2 \\ (x_1)^3 \\ \vdots \\ x_d \\ (x_d)^2 \\ (x_d)^3 \end{bmatrix} \text{cubic basis}$$

Perform linear regression on the new representation.

Change of Basis Demo





Locally Weighted Regression

Recall from KNN: locally weighted averaging

We can apply the same idea here: points that are further away should contribute less to the estimate.

To estimate the value for a specific test point x_t compute a linear regression with error weighted by distance:

$$\sum_{\vec{x} \in \text{data}} \frac{\left(y - \hat{f}(\vec{x})\right)}{dist(\vec{x_t}, \vec{x})} = \sum_{\vec{x} \in \text{data}} \frac{\epsilon_{\vec{x}}^2}{||\vec{x_t} - \vec{x}||_2}$$

Exam Topics

Covers the machine learning portion of the class.

- Supervised learning
  - Regression
  - Classification
- Unsupervised learning
  - Clustering
  - Dimensionality reduction
- Semi-supervised learning
  - Reinforcement learning

Know the differences between these topics.
Know what algorithms apply to which problems.

Machine Learning Algorithms

- neural networks
  - perceptrons
  - backpropagation
  - auto-encoders
  - deep learning
- decision trees
- naive Bayes
- k-nearest neighbors
- support vector machines
- locally-weighted average
- linear regression
- EM
  - K-means
  - Gaussian mixtures
- hierarchical clustering
  - agglomerative
  - divisive
- principal component analysis
- growing neural gas
- Q-learning
- approximate Q-learning
- ensemble learning