Gradient Descent for Linear Regression

Dan Sheldon

November 18, 2014

Announcements

- Reading / slides posted
- HW0 due before fourth hour tomorrow
- HW1 posted tomorrow, due next Friday

Today

- Quick review
- Intuition about partial derivatives
- Gradient descent update rules for linear regression
- Linear algebra

Review: Supervised Learning

Observe list of training examples $(x^{(i)}, y^{(i)})$, want to find a function h such that $y^{(i)} \approx h(x^{(i)})$ for all i

Variations:

- Type of x (real number, image, etc.)
- Type of y (real number, 0/1, $\{0, 1, \ldots, k\}$)
- Type of h

Cost function paradigm

Define **parametric** function $h_\theta(x)$ with parameters $\theta_0, \dots, \theta_n$. E.g.:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Define **cost function** $J(\theta_0, \dots, \theta_n)$ to measure quality (lower is better) of different hypotheses. E.g.:

$$J(\theta_0, \theta_1) = \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Use a numerical **optimization algorithm** to find $\theta_0, \dots, \theta_n$ to minimize $J(\theta_0, \dots, \theta_n)$. E.g., gradient descent.

Gradient Descent

To minimize a function $J(\theta_0, \theta_1)$ of two variables

- Intialize θ_0, θ_1 arbitrarily
- Repeat until convergence

$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

  - $\alpha =$ step-size or **learning rate** (not too big)

Partial derivative intuition

Interpretation of partial derivative: $\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$ is the rate of change along the θ_j axis

Example: illustrate funciton with elliptical contours

- Sign of $\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$?
- Sign of $\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$?
- Which has larger absolute value?

Gradient descent intuition

$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

- Why does this move in the direction of steepest descent?
- What would we do if we wanted to maximize $J(\theta_0, \theta_1)$ instead?

Illustration: contours of linear functions, circle around current point

Gradient descent for linear regression

Algorithm

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \quad \text{for } j = 0, 1$$

Cost function

$$J(\theta_0, \theta_1) = \sum_{i=1}^{m} \frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

We need to calculate partial derivatives.

Linear regression partial derivatives

Let's first do this with a single training example (x, y):

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2} \big(h_\theta(x) - y \big)^2$$

Linear regression partial derivatives

Let's first do this with a single training example (x, y):

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \big(h_\theta(x) - y\big)^2 \\
&= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y)
\end{aligned}$$

Linear regression partial derivatives

Let's first do this with a single training example (x, y):

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \big(h_\theta(x) - y\big)^2 \\
&= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) \\
&= \big(h_\theta(x) - y\big) \cdot \frac{\partial}{\partial \theta_j} \Big(\theta_0 + \theta_1 x - y\Big)
\end{aligned}
$$

Linear regression partial derivatives

Let's first do this with a single training example (x, y):

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \big(h_\theta(x) - y\big)^2 \\
&= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) \\
&= \big(h_\theta(x) - y\big) \cdot \frac{\partial}{\partial \theta_j} \Big(\theta_0 + \theta_1 x - y\Big)
\end{aligned}$$

So we get

$$\begin{aligned}
\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) &= \big(h_\theta(x) - y\big) \\
\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) &= \big(h_\theta(x) - y\big) x
\end{aligned}$$

Linear regression partial derivatives

More generally, with many training examples (work this out):

$$\frac{\partial}{\partial\theta_0} J(\theta_0, \theta_1) = \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\frac{\partial}{\partial\theta_1} J(\theta_0, \theta_1) = \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

Linear regression partial derivatives

More generally, with many training examples (work this out):

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

So the algorithm is:

$$\theta_0 := \theta_0 - \alpha \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\theta_1 := \theta_1 - \alpha \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right) x^{(i)}$$

Demo: parameter space vs. hypotheses

Show MATLAB gradient descent demo

Gradient descent in higher dimensions

Straightforward generalization to minimize a function $J(\theta_0, \ldots, \theta_n)$ of many variables:

- Intialize θ_j arbitrarily for all j
- Repeat until convergence

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_0} J(\theta) \quad \text{for all } j$$

(simultaneuous updates)