Advanced Algorithms

Floriano Zini

Free University of Bozen-Bolzano
Faculty of Computer Science

Academic Year 2013-2014

Lecture 12 – Linear regression (cont.)

These slides are taken from Andrew Ng, Machine Learning on Coursera - https://class.coursera.org/ml-003/lecture/preview
Practice advice 1: Feature Scaling

Idea: Make sure features are on a similar scale so that gradient descent can converge more quickly.

E.g. $x_1 =$ size (0-2000 feet$^2$)
      $x_2 =$ number of bedrooms (1-5)

Replacement:

\[
\begin{align*}
\theta_1 & \leftarrow \frac{x_1 - \mu_1}{s_1} \\
\theta_2 & \leftarrow \frac{x_2 - \mu_2}{s_2}
\end{align*}
\]

Rule of thumb: get every feature into approximately a range $-1 \leq x_i \leq 1$

Practice advice 2: Mean Normalization

Replace $x_i$ with $x_i - \mu_i$ to make features have approximately zero mean (Do not apply to $x_0 = 1$).

E.g.

\[
\begin{align*}
\theta_1 & \leftarrow \frac{x_1 - \mu_1}{s_1} \\
\theta_2 & \leftarrow \frac{x_2 - \mu_2}{s_2}
\end{align*}
\]

Average size = 1000
Average # bedrooms = 2

\[
\begin{align*}
-0.5 \leq x_1 \leq 0.5, -0.5 \leq x_2 \leq 0.5
\end{align*}
\]

General rule:

\[
\begin{align*}
x_i & \leftarrow \frac{x_i - \mu_i}{s_i}
\end{align*}
\]

$\mu_i$ average value of $x_i$ in training set
$s_i$ range (max - min) or standard deviation of $x_i$
Suppose you are using a learning algorithm to estimate the price of houses in a city. You want one of your features $x_i$ to capture the age of the house. In your training set, all of your houses have an age between 30 and 50 years, with an average age of 38 years. Which of the following would you use as features, assuming you use feature scaling and mean normalization?

- $x_i = \text{age of house}$
- $x_i = \frac{\text{age of house}}{50}$
- $x_i = \frac{\text{age of house} - 38}{50}$
- $x_i = \frac{\text{age of house} - 38}{20}$
### Practice advice 3: Learning Rate

#### Gradient descent

\[
\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)
\]

- "Debugging": How to make sure gradient descent is working correctly
- How to choose learning rate \( \alpha \)

---

#### Making sure gradient descent is working correctly

\[
\min_\theta J(\theta)
\]

Example automatic convergence test:

Declare convergence if \( J(\theta) \) decreases by less than \( 10^{-3} \) in one iteration.

\( J(\theta) \) should decrease after every iteration
Practice advice 3: Learning Rate

Making sure gradient descent is working correctly.

For sufficiently small $\alpha$, $J(\theta)$ should decrease on every iteration.

But if $\alpha$ is too small, gradient descent can be slow to converge.
Practice advice 3: Learning Rate

Summary:

- If $\alpha$ is too small: slow convergence
- If $\alpha$ is too large: $J(\theta)$ may not decrease on every iteration; may not converge.

To choose $\alpha$, try

$0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, \ldots$