Notes on Andrew Ng's CS 229 Machine Learning Course

Tyler Neylon

331.2016

These are notes I'm taking as I review material from Andrew Ng's CS 229 course on machine learning. Specifically, I'm watching these videos and looking at the written notes and assignments posted here. These notes are available in two formats: html and pdf.

I'll organize these notes to correspond with the written notes from the class.

1 On lecture notes 1

The notes in this section are based on lecture notes 1.

1.1 Gradient descent in general

Given a cost function $J(\theta)$, the general form of an update is

$$\theta_j := \theta_j - \alpha \frac{\partial J}{\partial \theta_j}.$$

It bothers me that α is an arbitrary parameter. What is the best way to choose this parameter? Intuitively, it could be chosen based on some estimate or actual value of the second derivative of J. What can be theoretically guaranteed about the rate of convergence under appropriate conditions?

Why not use Newton's method? A general guess: the second derivative of J becomes cumbersome to work with.

It seems worthwhile to keep my eye open for opportunities to apply improved optimization algorithms in specific cases.

1.2 Gradient descent on linear regression

I realize this is a toy problem because linear regression in practice is not solve iteratively, but it seems worth understanding well. The general update equation is, for a single example i,

$$\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)}.$$

The delta makes sense in that it is proportional to the error $y - h_\theta$, and in that the sign of the product $(y - h_\theta)x$ guarantees moving in the right direction. However, my first guess would be that the expression $(y - h_\theta)/x$ would provide a better update.

For example, suppose we have a single data point (x, y) where $x \neq 0$, and a random value of θ. Then a great update would be

$$\theta_1 := \theta_0 + (y - \theta_0 x)/x,$$

since the next hypothesis value h_θ would then be

$$h_\theta = \theta_1 x = \theta_0 x + y - \theta_0 x = y,$$

which is good. Another intuitive perspective is that we should be making *bigger* changes to θ_j when x_j is *small*, since it's harder to influence h_θ for such x values.

This is not yet a solidified intuition. I'd be interested in revisiting this question if I have time.

1.3 Persistence of trace

The trace of a square matrix obeys the nice property that

$$\text{tr } AB = \text{tr } BA. \tag{1}$$

One way to see this is to note that

$$\text{tr } AB = a_{ij}b_{ji} = \text{tr } BA,$$

where I'm using the informal shorthand notation that a variable repeated within a single product implies that the sum is taken over all relevant values of that variable. Specifically,

$$a_{ij}b_{ji} \text{ means } \sum_{i,j} a_{ij}b_{ji}.$$

I wonder if there's a more elegant way to verify (1).

This notation will become more useful in a moment.

Ng gives other interesting trace-based equations, examined next.

$$\text{Goal:} \quad \nabla_A \mathrm{tr}\, AB = B^T.$$

Since

$$\mathrm{tr}\, AB = a_{ij}b_{ji},$$

we have that

$$(\nabla_A \mathrm{tr}\, AB)_{ij} = b_{ji},$$

verifying the goal.

$$\text{Goal:} \quad \nabla_{A^T} f(A) = (\nabla_A f(A))^T.$$

TODO continue from here