Online Learning

Ex. Shipping service website

chosen $y=1$

not $y=0$

$p(y=1|x;\theta)$ price logistic regression

Repeat forever {

Get (x,y) corresponding to user.

Update θ using (x,y):

$$\theta_j := \theta_j - \alpha(h_\theta(x)-y)x_j \qquad j=(0,\cdots,n)$$

}

- It can adapt to changing user tastes changes over time
- It allows us to learn from a continuous stream of data,

since we use each example once than no longer need to process it again

Map-reduce and Data parallelism

$m=400$, Batch gradient descent: $\theta_j := \theta_j - \alpha\frac{1}{400}\sum_{i=1}^{400}(h_\theta(x^{(i)})-y^{(i)})x_j^{(i)}$

Machine 1 : Use $(x^{(1)}, y^{(1)}), \dots, (x^{(100)}, y^{(100)})$ $temp_j^{(1)}$

Machine : Use $(x^{(101)}, y^{(101)}), \dots, (x^{(200)}, y^{(200)})$ $temp_j^{(2)}$

Machine : Use $(x^{(201)}, y^{(201)}), \dots, (x^{(300)}, y^{(300)})$ $temp_j^{(3)}$

Machine : Use $(x^{(301)}, y^{(301)}), \dots, (x^{(400)}, y^{(400)})$ $temp_j^{(4)}$

} Combine

Many learning algorithms can be expressed as computing sums of functions over the training set