MS&E 317/CS 263: Algorithms for Modern Data Models, Spring 2014
`http://msande317.stanford.edu`.
Instructors: Ashish Goel and Reza Zadeh, Stanford University.

Lecture 9, 9/26/2014. Scribed by Adrien Fallou.

Outline:

- The perceptron algorithm
- Stochastic gradient descent

This lecture starts the series on machine learning. We'll focus on supervised learning and classification.

10 The perceptron

10.1 Setting

We have a set of points $S \in \mathbb{R}^d$, $S = \{X_1, X_2, ...X_n\}$ with $\|X_i\|_2 = 1$. The points are coming as a stream, and each X_i is associated with a label y_i (+ or −).

We assume the points are separable, i.e. there exists a linear classifier that can correctly classify all points. We are looking to find this classifier. We define the margin γ as the shortest distance from any point to this classifier.

In order not to have to deal with an intercept term, we also assume there is a given dimension k for which $[X_i]_k = 1$ for all i.

10.2 Algorithm

$w_0 = 0$
    for each point X
        if $w_i X > 0$ predict $y_i = +$ else predict $y_i = -$
            if we make a mistake
                if true answer is + update $w_{i+1} = w_i + X$
                if true answer is − update $w_{i+1} = w_i - X$

10.3 Properties

M denotes the number of mistakes the algorithm makes, i.e. the total number of mistakes we'd get by running the algorithm an indefinite amount of time

Proposition 10.1 $M \leq \frac{1}{\gamma^2}$

Proof: The idea is to prove that $w_i \cdot w^*$ becomes bigger (i.e. both vectors become closer), while $\|w_i\|_2^2$ remains bounded.

- We claim $w_i \cdot w^* \geq w_i \cdot w^* + \gamma$ when we make a mistake.

  If we make a mistake:

  - If we predict $w_i x < 0$ and we make a mistake:

$$\begin{aligned} w_{i+1} \cdot w^* &= (w_i + X) \cdot w^* \\ &= w_i \cdot w^* + X \cdot w^* \\ &\geq w_i \cdot w^* + \gamma \text{ since we made a mistake, and by definition of } \gamma \end{aligned}$$

  - In the opposite case, we have: $w_{i+1} \cdot w^* = w_i \cdot w^* - w^* \cdot X \geq w_i \cdot w^* + \gamma$

- We'll also get a handle on the growth of $\|w_i\|_2^2$

  If we predict $w_i \cdot X > 0$ and make a mistake we have, since $\|X\|_2^2 = 1$

$$\begin{aligned} \|w_{i+1}\|_2^2 &= \|w_i + X\|_2^2 \\ &= \|w_i\|_2^2 + 2w_i \cdot X + \|X\|_2^2 \\ &\leq \|w_i\|_2^2 + 1 \end{aligned}$$

  And we easily get the same result for the opposite case

Thus after M mistakes we have:

- $w_{M+1} \cdot w^* \geq M\gamma$
- $\|w_{M+1}\|_2 \leq \sqrt{M}$

We can suppose $\|w^*\| = 1$, we now get

$$M\gamma \leq \|w_{M+1} \cdot w^*\|_2 \leq \|w_{M+1}\|_2 \leq \sqrt{M}$$

$$M \leq \frac{1}{\gamma^2}$$

■

Remark 10.1 *One of the problems we may have is that we need a nonzero margin for the perceptron to work, which we may not always have.*

11 Stochastic Gradient Descent (SGD)

We want to minimize

$$F(w) = \sum_{i=1}^{n} (w^T X_i - y_i)^2 = \sum_{i=1}^{n} Q_i(w)$$

If we minimize with regular ("batch") gradient descent, we have:

$$w_{i+1} = w_i - \alpha \nabla F$$

This approach requires going through the whole dataset to advance one timestep, which is not feasible when the data gets too big. Instead we can write:

$$\nabla F = \sum_{i=1}^{n} \nabla Q_i(w)$$

And the idea is to update our classifier w_i with each datapoint, as:

$$w_{i+1} = w_i - \alpha \nabla Q_i(w)$$

We go through the whole data in this way, shuffle the data and start again. This algorithm is called stochastic gradient descent.

Because we're only making one small update at a time, we'll need more iterations to get the same precision. The following table compares both algorithm:

	GD	2 GD	SGD	2 SGD
time per iteration	n	n	1	1
number of iterations to ε accuracy	$-\log \varepsilon$	$\log\log \frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$
total time	$-\log \varepsilon$	$n \log\log \frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$	$\frac{1}{\varepsilon}$

Remark 11.1 *Newton's method iterates as $w_{i+1} = w_i - \alpha H^{-1} \nabla F$ (we called it "2 GD" in the table). We can apply the same idea as for stochastic gradient descent and iterate as $w_{i+1} = w_i - \alpha H^{-1} \nabla Q_i$ ("2 SGD" in the table)*

Remark 11.2 *It seems that smallest total time we get is for Newton's method. However, this involves computing a Hessian on the whole data, which is not computationally feasible when the data gets large.*