梯度下降法的三种形式BGD、SGD以及MBGD

> https://zhuanlan.zhihu.com/p/25765735

1. 线性回归问题

线性回归函数的假设函数为：

$$h_\theta = \sum_{j=0}^{n} \theta_j x_j$$

对应的损失函数为：

$$J_{train}(\theta) = 1/(2m) \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

2. 批量梯度下降法BGD

- 目的是要**损失函数尽可能的小**；
- **不断反复的更新weights使得损失函数减小，**直到满足要求时停止；
- **每次参数更新的伪代码**如下：

repeat{

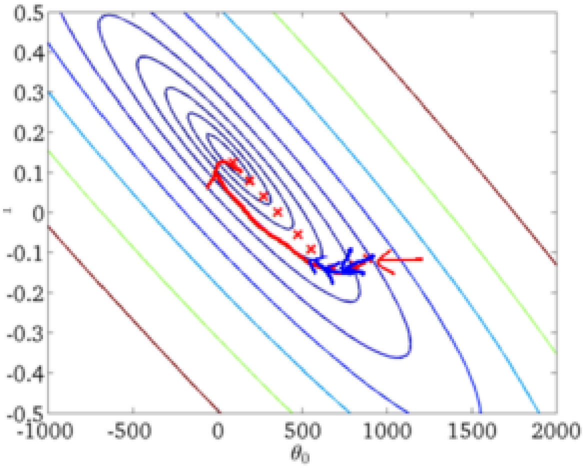
$$\theta_j' = \theta_j + \frac{1}{m}\sum_{i=1}^{m}(y^i - h_\theta(x^i))x_j^i$$

(for every j=0, ... , n)

}

我们每一次的参数更新都用到了所有的训练数据（比如有m个，就用到了m个），如果训练数据非常多的话，**是非常耗时的。**

- **下面给出批梯度下降的收敛图：**



从图中，我们可以得到BGD迭代的次数相对较少。

2. 随机梯度下降法SGD

- 利用**每个样本**的损失函数对θ求偏导得到对应的梯度，来更新θ：

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

- 更新过程如下：

1. Randomly shuffle dataset ;

2. **repeat**{

for i=1, ... , m{

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

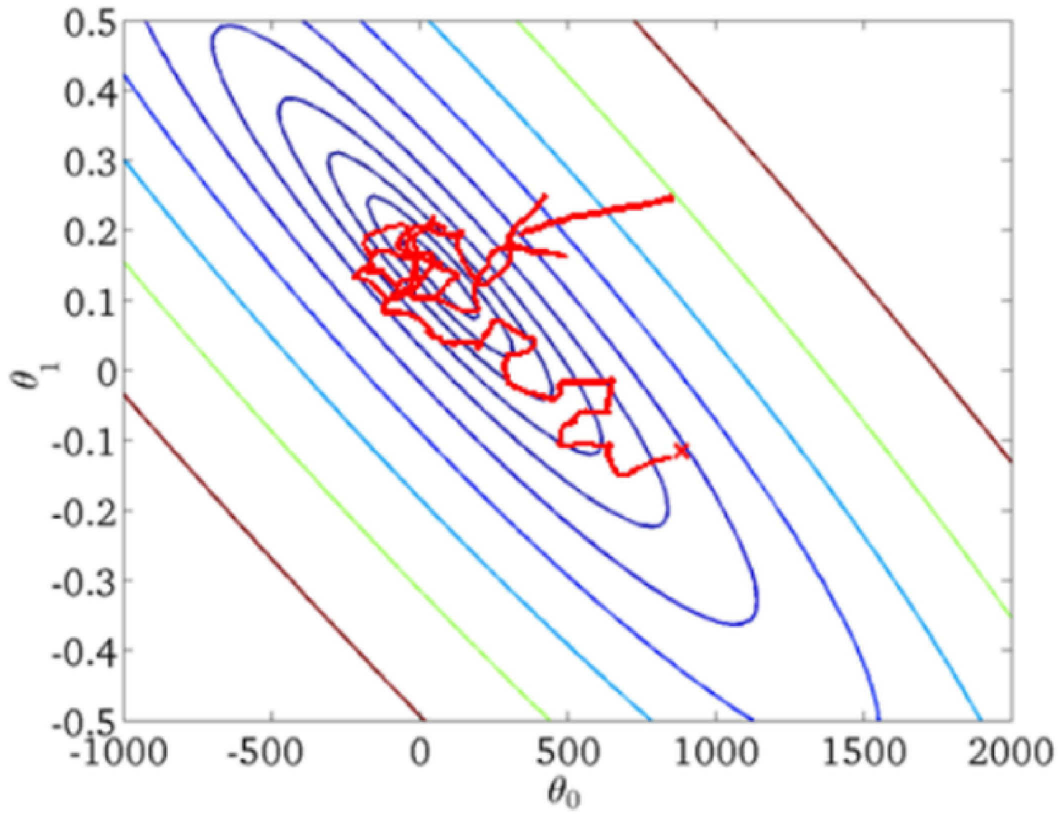
(for j=0, ... , n)

}

}

SGD并不是每次迭代都向着整体最优化方向。

- **随机梯度下降收敛图如下：**



我们可以从图中看出SGD迭代的次数较多，在解空间的搜索过程看起来很盲目。**但是大体上是往着最优值方向移动。**

3. mini-batch 小批量梯度下降法MBGD

我们从上面两种梯度下降法可以看出，其各自均有优缺点，那么能不能在两种方法的性能之间取得一个折衷呢？**即，算法的训练过程比较快，而且也要保证最终参数训练的准确率，**小批量梯度下降法（Mini-batch Gradient Descent，简称MBGD）

以10个样本作为一个mini-batch更新

更新伪代码如下：

Repeat{

for i=1, 11, 21, 31, ... , 991{

$$\theta_j := \theta_j - \alpha \frac{1}{10} \sum_{k=i}^{i+9} (h_\theta(x^{(k)}) - y^{(k)}) x_j^{(k)}$$

(for every j=0, ... , n)

}

}

4. 实例以及代码详解

线性拟合的实例表述