梯度下降（Gradient Descent）小结

在求解机器学习算法的模型参数，即无约束优化问题时，梯度下降（Gradient Descent）是最常采用的方法之一，另一种常用的方法是最小二乘法。这里就对梯度下降法做一个完整的总结。

1. 梯度

在微积分里面，对多元函数的参数求∂偏导数，把求得的各个参数的偏导数以向量的形式写出来，就是梯度。比如函数f(x,y), 分别对x,y求偏导数，求得的梯度向量就是$(\partial f/\partial x, \partial f/\partial y)^T$,简称grad f(x,y)或者$\nabla f(x,y)$。对于在点$(x_0,y_0)$的具体梯度向量就是$(\partial f/\partial x_0, \partial f/\partial y_0)^T$.或者$\nabla f(x_0,y_0)$，如果是3个参数的向量梯度，就是$(\partial f/\partial x, \partial f/\partial y, \partial f/\partial z)^T$,以此类推。

那么这个梯度向量求出来有什么意义呢？他的意义从几何意义上讲，就是函数变化增加最快的地方。具体来说，对于函数f(x,y),在点(x_0,y_0)，沿着梯度向量的方向就是$(\partial f/\partial x_0, \partial f/\partial y_0)^T$的方向是f(x,y)增加最快的地方。或者说，沿着梯度向量的方向，更加容易找到函数的最大值。反过来说，沿着梯度向量相反的方向，也就是 $-(\partial f/\partial x_0, \partial f/\partial y_0)^T$的方向，梯度减少最快，也就是更加容易找到函数的最小值。

2. 梯度下降与梯度上升

在机器学习算法中，在最小化损失函数时，可以通过梯度下降法来一步步的迭代求解，得到最小化的损失函数，和模型参数值。反过来，如果我们需要求解损失函数的最大值，这时就需要用梯度上升法来迭代了。

梯度下降法和梯度上升法是可以互相转化的。比如我们需要求解损失函数f(θ)的最小值，这时我们需要用梯度下降法来迭代求解。但是实际上，我们可以反过来求解损失函数 -f(θ)的最大值，这时梯度上升法就派上用场了。

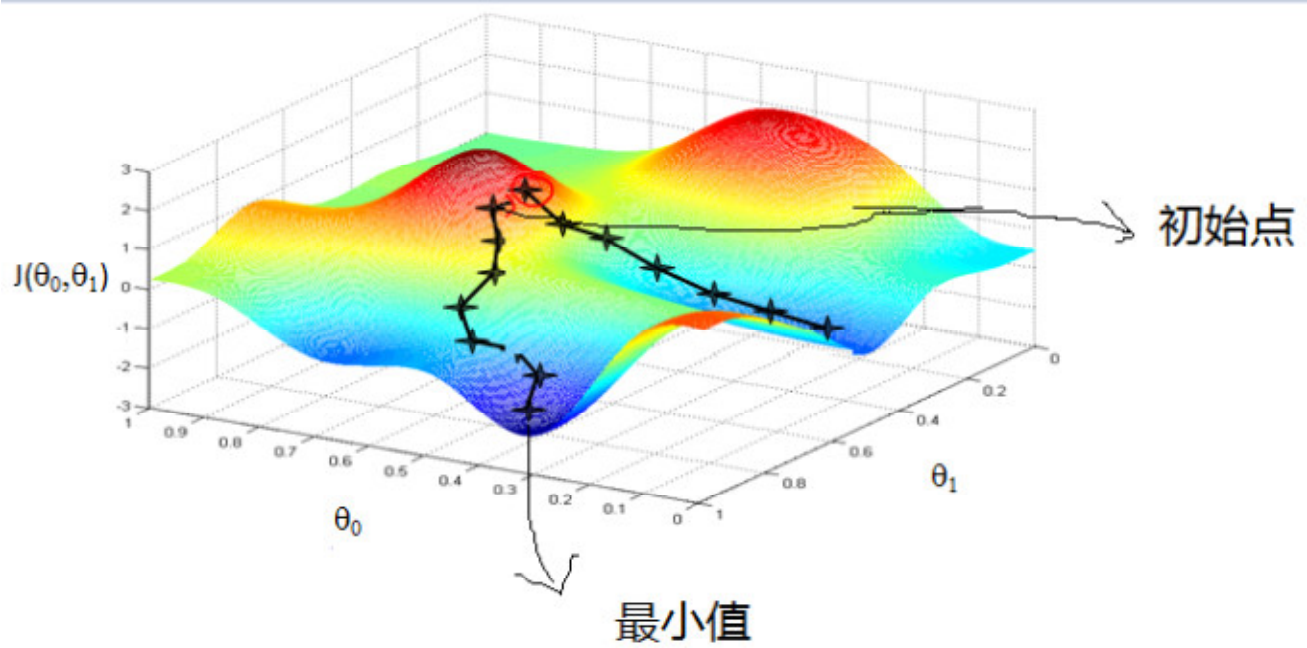
下面来详细总结下梯度下降法。

3. 梯度下降法算法详解

3.1 梯度下降的直观解释

首先来看看梯度下降的一个直观的解释。比如我们在一座大山上的某处位置，由于我们不知道怎么下山，于是决定走一步算一步，也就是在每走到一个位置的时候，求解当前位置的梯度，沿着梯度的负方向，也就是当前最陡峭的位置向下走一步，然后继续求解当前位置梯度，向这一步所在位置沿着最陡峭最易下山的位置走一步。这样一步步的走下去，一直走到觉得我们已经到了山脚。当然这样走下去，有可能我们不能走到山脚，而是到了某一个局部的山峰低处。

从上面的解释可以看出，梯度下降不一定能够找到全局的最优解，有可能是一个局部最优解。当然，如果损失函数是凸函数，梯度下降法得到的解就一定是全局最优解。



3.2 梯度下降的相关概念

在详细了解梯度下降的算法之前，我们先看看相关的一些概念。

1. 步长（Learning rate）：步长决定了在梯度下降迭代的过程中，每一步沿梯度负方向前进的长度。用上面下山的例子，步长就是在当前这一步所在位置沿着最陡峭最易下山的位置走的那一步的长度。

2.特征（feature）：指的是样本中输入部分，比如2个单特征的样本（$x^{(0)}, y^{(0)}$），（$x^{(1)}, y^{(1)}$），则第一个样本特征为$x^{(0)}$，第一个样本输出为$y^{(0)}$。

3. 假设函数（hypothesis function）：在监督学习中，为了拟合输入样本，而使用的假设函数，记为$h_\theta(x)$。比如对于单个特征的m个样本（$x^{(i)}, y^{(i)}$）$(i = 1, 2, \ldots m)$，可以采用拟合函数如下：$h_\theta(x) = \theta_0 + \theta_1 x$。

4. 损失函数（loss function）：为了评估模型拟合的好坏，通常用损失函数来度量拟合的程度。损失函数极小化，意味着拟合程度最好，对应的模型参数即为最优参数。在线性回归中，损失函数通常为样本输出和假设函数的差取平方。比如对于m个样本（x_i, y_i）$(i = 1, 2, \ldots m)$，采用线性回归，损失函数为：

$J(\quad,\quad)=\sum_{i=1}^{m}(\quad(\quad)-$

θ_0,θ_1 $h_\theta(x_i)$ $y_i)^2$

其中x_i表示第i个样本特征，y_i表示第i个样本对应的输出，$h_\theta(x_i)$为假设函数。

3.3 梯度下降的详细算法

梯度下降法的算法可以有代数法和矩阵法（也称向量法）两种表示，如果对矩阵分析不熟悉，则代数法更加容易理解。不过矩阵法更加的简洁，且由于使用了矩阵，实现逻辑更加的一目了然。这里先介绍代数法，后介绍矩阵法。

3.3.1 梯度下降法的代数方式描述

1. 先决条件： 确认优化模型的假设函数和损失函数。

比如对于线性回归，假设函数表示为 $h_\theta(x_1,x_2,...x_n)=\theta_0+\theta_1x_1+...+\theta_nx_n$ ，其中θ_i (i = 0,1,2... n)为模型参数，x_i (i = 0,1,2... n)为每个样本的n个特征值。这个表示可以简化，我们增加一个特征$x_0=1$ ，这样$h_\theta(x_0,x_1,...x_n)=\sum_{i=0}^{n}\theta_ix_i$ 。

同样是线性回归，对应于上面的假设函数，损失函数为：

$$J(\theta_0,\theta_1...,\theta_n)=\frac{1}{2m}\sum_{j=0}^{m}(h_\theta(x_0^{(j)},x_1^{(j)},...x_n^{(j)})-y_j)^2$$

2. 算法相关参数初始化：主要是初始化$\theta_0,\theta_1...,\theta_n$,算法终止距离ε 以及步长α 。在没有任何先验知识的时候，我喜欢将所有的θ 初始化为0， 将步长初始化为1。在调优的时候再 优化。

3. 算法过程：

1）确定当前位置的损失函数的梯度，对于θ_i ,其梯度表达式如下：

$$\frac{\partial}{\partial\theta_i}J(\theta_0,\theta_1...,\theta_n)$$

2）用步长乘以损失函数的梯度，得到当前位置下降的距离，即 $\alpha\frac{\partial}{\partial\theta_i}J(\theta_0,\theta_1,\ldots,\theta_n)$ 对应于前面登山例子中的某一步。

3）确定是否所有的θ_i ,梯度下降的距离都小于ε ，如果小于ε 则算法终止，当前所有的θ_i (i=0,1,...n)即为最终结果。否则进入步骤4.

4）更新所有的θ ，对于θ_i ，其更新表达式如下。更新完毕后继续转入步骤1.

$$\theta_i=\theta_i-\alpha\frac{\partial}{\partial\theta_i}J(\theta_0,\theta_1\ldots,\theta_n)$$

下面用线性回归的例子来具体描述梯度下降。假设我们的样本是 $(x_1^{(0)},x_2^{(0)},\ldots x_n^{(0)},y_0),(x_1^{(1)},x_2^{(1)},\ldots x_n^{(1)},y_1),\ldots(x_1^{(m)},x_2^{(m)},\ldots x_n^{(m)},y_m)$,损失函数如前面先决条件所述：

$$J(\theta_0,\theta_1\ldots,\theta_n)=\frac{1}{2m}\sum_{j=0}^{m}(h_\theta(x_0^{(j)},x_1^{(j)},\ldots x_n^{(j)})-y_j)^2$$ 。

则在算法过程步骤1中对于θ_i 的偏导数计算如下：

$$\frac{\partial}{\partial\theta_i}J(\theta_0,\theta_1\ldots,\theta_n)=\frac{1}{m}\sum_{j=0}^{m}(h_\theta(x_0^{(j)},x_1^{(j)},\ldots x_n^{(j)})-y_j)x_i^{(j)}$$

由于样本中没有x_0 上式中令所有的x_0^j 为1.

步骤4中θ_i 的更新表达式如下：

$$\theta_i=\theta_i-\alpha\frac{1}{m}\sum_{j=0}^{m}(h_\theta(x_0^{(j)},x_1^{(j)},\ldots x_n^{j})-y_j)x_i^{(j)}$$

从这个例子可以看出当前点的梯度方向是由所有的样本决定的，加$\frac{1}{m}$ 是为了好理解。由于步长也为常数，他们的乘机也为常数，所以这里$\alpha\frac{1}{m}$ 可以用一个常数表示。

在下面第4节会详细讲到的梯度下降法的变种，他们主要的区别就是对样本的采用方法不同。这里我们采用的是用所有样本。

3.3.2 梯度下降法的矩阵方式描述

这一部分主要讲解梯度下降法的矩阵方式表述，相对于3.3.1的代数法，要求有一定的矩阵分析的基础知识，尤其是矩阵求导的知识。

1. 先决条件： 和3.3.1类似， 需要确认优化模型的假设函数和损失函数。对于线性回归，假设函数 $h_\theta(x_1, x_2, ...x_n) = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n$ 的矩阵表达方式为：

$h_\theta(\mathbf{x}) = \mathbf{X}\theta$ ，其中， 假设函数$h_\theta(\mathbf{X})$ 为mx1的向量,θ 为(n+1)x1的向量，里面有n个代数法的模型参数。$\mathbf{X}$ 为mx(n+1)维的矩阵。m代表样本的个数，n+1代表样本的特征数。

损失函数的表达式为：$J(\theta) = \frac{1}{2}(\mathbf{X}\theta - \mathbf{Y})^T(\mathbf{X}\theta - \mathbf{Y})$ ，其中$\mathbf{Y}$ 是样本的输出向量，维度为mx1.

2. 算法相关参数初始化 θ 向量可以初始化为默认值，或者调优后的值。算法终止距离ε ，步长α 和3.3.1比没有变化。

3. 算法过程：

1）确定当前位置的损失函数的梯度，对于θ 向量,其梯度表达式如下：

$$\frac{\partial}{\partial\theta}J(\theta)$$

2）用步长乘以损失函数的梯度，得到当前位置下降的距离，即$\alpha\frac{\partial}{\partial\theta}J(\theta)$ 对应于前面登山例子中的某一步。

3）确定θ 向量里面的每个值,梯度下降的距离都小于ε ，如果小于ε 则算法终止，当前θ 向量即为最终结果。否则进入步骤4.

4）更新θ 向量，其更新表达式如下。更新完毕后继续转入步骤1.

$$\theta = \theta - \alpha\frac{\partial}{\partial\theta}J(\theta)$$

还是用线性回归的例子来描述具体的算法过程。

损失函数对于θ 向量的偏导数计算如下：

$$\frac{\partial}{\partial\theta}J(\theta) = \mathbf{X}^T(\mathbf{X}\theta - \mathbf{Y})$$

步骤4中θ 向量的更新表达式如下：

$$\theta = \theta - \alpha \quad (\mathbf{X}\theta - \mathbf{Y})\mathbf{X}_T$$

对于3.3.1的代数法，可以看到矩阵法要简洁很多。这里面用到了矩阵求导链式法则，和两个矩阵求导的公式。

公式1：$\frac{\partial}{\partial \mathbf{X}}(\mathbf{X}\mathbf{X}^{\mathbf{T}}) = 2\mathbf{X}$

公式2：$\frac{\partial}{\partial \theta}(\mathbf{X}\theta) = \mathbf{X}^{\mathbf{T}}$

如果需要熟悉矩阵求导建议参考张贤达的《矩阵分析与应用》一书。

3.4 梯度下降的算法调优

在使用梯度下降时，需要进行调优。哪些地方需要调优呢?

1. 算法的步长选择。在前面的算法描述中，我提到取步长为1，但是实际上取值取决于数据样本，可以多取一些值，从大到小，分别运行算法，看看迭代效果，如果损失函数在变小，说明取值有效，否则要增大步长。前面说了。步长太大，会导致迭代过快，甚至有可能错过最优解。步长太小，迭代速度太慢，很长时间算法都不能结束。所以算法的步长需要多次运行后才能得到一个较为优的值。

2. 算法参数的初始值选择。 初始值不同，获得的最小值也有可能不同，因此梯度下降求得的只是局部最小值；当然如果损失函数是凸函数则一定是最优解。由于有局部最优解的风险，需要多次用不同初始值运行算法，关键损失函数的最小值，选择损失函数最小化的初值。

3.归一化。由于样本不同特征的取值范围不一样，可能导致迭代很慢，为了减少特征取值的影响，可以对特征数据归一化，也就是对于每个特征x，求出它的期望$\overline{x}$ 和标准差std(x)，然后转化为：

$$\frac{x-\overline{x}}{std(x)}$$

这样特征的新期望为0，新方差为1，迭代次数可以大大加快。

4. 梯度下降法大家族（BGD，SGD，MBGD）

4.1 批量梯度下降法（Batch Gradient Descent)

批量梯度下降法，是梯度下降法最常用的形式，具体做法也就是在更新参数时使用所有的样本来进行更新，这个方法对应于前面3.3.1的线性回归的梯度下降算法，也就是说3.3.1的梯度下降算法就是批量梯度下降法。

$$\theta_i = \theta_i - \alpha \sum_{j=0}^{m} (h_\theta(x_0^{(j)}, x_1^{(j)}, \dots x_n^{(j)}) - y_j) x_i^{(j)}$$

由于我们有m个样本，这里求梯度的时候就用了所有m个样本的梯度数据。

4.2 随机梯度下降法（Stochastic Gradient Descent)

随机梯度下降法，其实和批量梯度下降法原理类似，区别在与求梯度时没有用所有的m个样本的数据，而是仅仅选取一个样本 来求梯度。对应的更新公式是：

$$\theta_i = \theta_i - \alpha (h_\theta(x_0^{(j)}, x_1^{(j)}, \dots x_n^{(j)}) - y_j) x_i^{(j)}$$

随机梯度下降法，和4.1的批量梯度下降法是两个极端，一个采用所有数据来梯度下降，一个用一个样本来梯度下降。自然各自的优缺点都非常突出。对于训练速度来说，随机梯度下降法由于每次仅仅采用一个样本来迭代，训练速度很快，而批量梯度下降法在样本量很大的时候，训练速度不能让人满意。对于准确度来说，随机梯度下降法用于仅仅用一个样本决定梯度方向，导致解很有可能不是最优。对于收敛速度来说，由于随机梯度下降法一次迭代一个样本，导致迭代方向变化很大，不能很快的收敛到局部最优解。

那么，有没有一个中庸的办法能够结合两种方法的优点呢？有！这就是4.3的小批量梯度下降法。

4.3 小批量梯度下降法（Mini-batch Gradient Descent)

小批量梯度下降法是批量梯度下降法和随机梯度下降法的折衷，也就是对于m个样本，我们采用x个样子来迭代，1 x m。一般可以取x=10，当然根据样本的数据，可以调整这个x的值。对应的更新公式是：

$$\theta_i = \theta_i - \alpha \sum_{j=t}^{t+x-1} (h_\theta(x_0^{(j)}, x_1^{(j)}, \dots x_n^{(j)}) - y_j) x_i^{(j)}$$

5. 梯度下降法和其他无约束优化算法的比较

在机器学习中的无约束优化算法，除了梯度下降以外，还有前面提到的最小二乘法，此外还有牛顿法和拟牛顿法。

梯度下降法和最小二乘法相比，梯度下降法需要选择步长，而最小二乘法不需要。梯度下降法是迭代求解，最小二乘法是计算解析解。如果样本量不算很大，且存在解析解，最小二乘法比起梯度下降法要有优势，计算速度很快。但是如果样本量很大，用最小二乘法由于需要求一个超级大的逆矩阵，这时就很难或者很慢才能求解解析解了，使用迭代的梯度下降法比较有优势。

梯度下降法和牛顿法/拟牛顿法相比，两者都是迭代求解，不过梯度下降法是梯度求解，而牛顿法/拟牛顿法是用二阶的海森矩阵的逆矩阵或伪逆矩阵求解。相对而言，使用牛顿法/拟牛顿法收敛更快。但是每次迭代的时间比梯度下降法长。

分类 0081. 机器学习

标签 梯度下降

刘建平Pinard
关注 - 14
粉丝 - 1988
+加关注

57 1

下一篇：最小二乘法小结

posted 2016-10-17 22 49 刘建平Pinard 阅读(153060) 评论(97) 编辑 收藏

评论列表

51楼 楼主 2018-03-02 10 44 刘建平Pinard

__ liuhuisen
你好，的确是写错了，已经改正，感谢指出错误。

支持(0) 反对(0)

52楼 2018-03-05 09 10 钱大爷

你好,请教一下,为什么采用Logistic egression是,对分类型数据要采用One ot编码方式,而不能采用Label ncoder.谢谢.

支持(0) 反对(0)

53楼 2018-03-13 10 45 卡卡麓

你好，写的很棒，看完受益匪浅。
在 3.4 梯度下降的算法调优中，第三点，归一化中，有点小看法。减去均值再除以标准差，应该叫标准化。归一化则又称正则化，是对样本的行做L1或L2正则的操作。

支持(2) 反对(0)

54楼 楼主 2018-03-13 11 29 刘建平Pinard

__ 卡卡麓
你好，现在标准化和归一化这两个词经常混用。直接说标准化或者归一化还不能确定真正的方法。
比如最常用的z-score标准化，也就是我上面讲到了转化为均值0，方差1.
还有max-min归一化，用所有值减去min然后除以（max-min）。
当然还有你说的L1，L2正则化的标准化方法。

对应sklearn的库：
max-min归一化 preprocessing.MinMaxScaler
z-score标准化 preprocessing.StandardScaler
L1/L2归一化 preprocessing.Normalizer (可以参数选择L1还是L2)

支持(1) 反对(0)

55楼 2018-03-13 13 17 sonderxixing

受教了

支持(0) 反对(0)

56楼 2018-03-14 10 16 卡卡麓

__ 刘建平Pinard
感谢回复，受教了，这几个概念确实混乱，在不同的书上有不同表述。

支持(0) 反对(0)

57楼 2018-03-15 18 05 小溏

写的真好！赞一万个！

支持(0) 反对(0)

58楼 2018-03-16 11 42 骑猪去流浪

支持支持 学习了

支持(0) 反对(0)

59楼 2018-03-17 09 15 ock CG

您好，看了您的文章收获很大，感谢您。

支持(0) 反对(0)

60楼 2018-03-20 14 43 TinyLaughing

博主，写的真赞，学习了！
之前一直有关于损失函数前面系数不一致的疑惑，看了这篇文章后，明白了。
谢谢博主！

支持(0) 反对(0)

61楼 2018-03-23 11 32 楊義Sunshine

$(x_1^{(m)}, x_2^{(m)}, \dots x_n^{(m)}, y_n)$
代数法最后一个y的下标应该是m。

支持(0) 反对(0)

62楼 楼主 2018-03-25 18 37 刘建平Pinard

__ 楊義Sunshine
你好，是写错啦，感谢指正，已经修改了。

支持(0) 反对(0)

63楼 2018-04-03 09 48 永不息的舞步

楼主,请教一下,z-socre标准化,max-min归一化,和l1,l2标准化的使用场合

支持(0) 反对(0)

64楼 楼主 2018-04-03 11 27 刘建平Pinard

__ 永不息的舞步
你好，这看你的问题需要。这里只举一个例子：
比如我们要做线性回归，我们期望所有的数据度量衡一致，那么我们可以用z-score标准化或者max-min归一化。但是如果我们是要做PCA，那么就不能用z-score标准化了，为什么呢？因为我们需要方差的差异来选择合适的降维的对应维度，如果都是方差1，那么就没法弄了。

支持(0) 反对(1)

65楼 2018-04-03 11 30 永不息的舞步

__ 刘建平Pinard
好的,谢谢您的指导

支持(0) 反对(0)

66楼 2018-04-11 14 15 拉卡拉卡

看了楼主很多文章！写的太棒了！反复看感觉都不一样
想打印下来，楼主有没有导出过pdf版本呀，求分享
谢谢！

支持(1) 反对(0)

67楼 楼主 2018-04-11 14 32 刘建平Pinard

__ 拉卡拉卡
你好，那倒没有，主要原因是LaTex是直接手写的。不过web版的想打印应该也可以的。你可以试一试。

支持(0) 反对(0)

68楼 2018-04-11 14 43 拉卡拉卡

__ 刘建平Pinard
好的，谢谢回复

支持(0) 反对(0)

69楼 2018-04-11 15 24 水来浪去

总结的很好，收益了!

支持(0) 反对(0)

70楼 2018-04-11 16 05 訫无杂念

博主你好，有个问题需要向你请教一下，原文中说，用步长乘以损失函数的梯度，得到当前位置下降的距离，步长就是斜边距离，损失函数梯度可以理解为斜率，下降距离也就是垂直距离，不知道怎么算出来的。望指教，谢谢

支持(0) 反对(0)

71楼 楼主 2018-04-11 16 17 刘建平Pinard

__ 訫无杂念
你好，损失函数梯度是一个关于样本特征的函数，而步长是一个定值。对于每一个样本，我们都可以计算它对应于损失函数梯度的取值。而对于所有的训练样本，我们可以得到所有的训练样本损失函数梯度的取值。
将这些取值加起来取平均值，就是我们需要计算的梯度值。

支持(0) 反对(0)

72楼 2018-04-11 16 23 訫无杂念

谢谢。我有点不明白步长 损失函数梯度=当前位置下降距离，我推了一下没有推出来

支持(0) 反对(0)

73楼 2018-04-13 09 50 wensss

博主您好，有个问题想请教你一下，在梯度下降法中用梯度下降的距离小于阈值来跳出循环，那么，每次得到的梯度是个二维矩阵，这时候它的下降距离应该怎么衡量呢?

支持(0) 反对(0)

74楼 楼主 2018-04-14 23 19 刘建平Pinard

__ wensss
你好，对于向量来说，那么如果这个向量的各个维度的标量改变都小于阈值，则跳出。
对于矩阵来说，可以看所有的位置的值改变是否小于阈值。

支持(0) 反对(0)

75楼 2018-04-19 11 31 lu ianhan

andrew ng的视频搬过来的

支持(0) 反对(0)

76楼 2018-05-01 19 17 thinkingbear

3.3.2中1.先决条件里面损失函数里面是不是少一个1/m的系数?

支持(0) 反对(0)

77楼 2018-05-01 19 21 thinkingbear

3.3.2中1.先决条件里面x矩阵是不是应该在最后加一列1变成m（n+1)的矩阵，theta是从theta0到thetan，是n+1维的?

支持(0) 反对(0)

78楼 楼主 2018-05-01 23 31 刘建平Pinard

__ thinkingbear
你好。
3.3.1，那里加不加1/m都不影响取极值的时候对应的模型参数的值，仅仅影响损失函数的极值，而损失函数的极值不是我们关心的，我们仅仅关心取极值的时候对应的模型参数的值，因此加不加都没有关系。

3.3.2 那里的确有错误，感谢指出，修改了。

支持(0) 反对(0)

79楼 2018-05-02 20 34 shadow3002

受教了，非常感谢!

支持(0) 反对(0)

80楼 2018-05-10 19 22 kylin0228

博主您好
对于步长 我看李航老师那本书上写的用一维搜索，
是不是每次迭代的时候，我都要去遍找一下最优的步长？这样不会计算很大吗？谢谢

支持(0) 反对(0)

81楼 楼主 2018-05-10 22 01 刘建平Pinard

__ kylin0228
你好，步长是一个需要调的参数，但是在每次训练的时候，步长一般是给定的（特殊的步长变化的也有，不过不是大多数的情况），这样在该次训练的每次迭代的时候步长不变。不同次训练则可以选择不同的步长。

支持(0) 反对(0)

82楼 2018-05-13 10 25 多一点

您好，PCA降维很多的情况下，是需要将数据进行标准化的

支持(0) 反对(0)

83楼 2018-05-21 20 46 ack47

__ kylin0228
如果步长选取不当，可能导致得不到最优解。Andrew NG在coursera machine learning课程(免费的）上解释的很清楚，感兴趣的可以看看

支持(0) 反对(0)

84楼 2018-05-21 20 48 ack47

博主写的很赞!
是否可以在文末加一些 eference，把写作过程中参考的资料列出来，比如书籍或者网页?

支持(0) 反对(0)

85楼 楼主 2018-05-22 19 56 刘建平Pinard

__ ack47
16年到17年那批文章都是基于我15年到16年的学习笔记写的，当时的参考文献没有记，所以后面就没法写了。感谢你的建议。一般比较新的比如深度学习的文章我都是写了主要参考资料的。后面会注意。

支持(0) 反对(0)

86楼 2018-06-12 14 20 卢杰

博主，您好，第三段的内容不是很理解，然后去知乎搜了一下，明白了。分析一下。
梯度下降法(gradient descent)是一种常用的一阶(first-order)优化方法，是求解无约束优化问题最简单、最经典的方法之一。我们来考虑一个无约束优化问题 min x f(x) , 其中 f(x) 为连续可微函数，如果我们能够构造一个序列 x 0, x 1, x 2, ... ，并能够满足:

f(x t+1) f(x t), t=0,1,2,...
那么问题就是如何找到下一个点 x t+1 ,并保证 f(x t+1) f(x t) 呢? 假设我们当前的函数 f(x) 的形式是上图的形状，现在我们随机找了一个初始的点 x 1 ，对于一元函数来说，函数值只会随着 x 的变化而变化，那么我们就设计下一个 x t+1 是从上一个 x t 沿着某一方向走一小步 Delta x 得到的。此处的关键问题就是：这一小步的方向是朝向哪里?

对于一元函数来说， x 是会存在两个方向：要么是正方向(Delta x 0)，要么是负方向 （ Delta x 0 ），如何选择每一步的方向，就需要用到大名鼎鼎的泰勒公式，先看一下下面这个泰勒展式:

f(x+ Delta x) sime f(x) + Delta x triangledown f(x)

左边就是当前的 x 移动一小步 Delta x 之后的下一个点位，它近似等于右边。前面我们说了关键问题是找到一个方向，使得 f(x+ Delta x) f(x) ，那么根据上面的泰勒展式，显然我们需要保证:

Delta x triangledown f(x) 0

可选择令:

Delta x = - alpha triangledown f(x)，(alpha 0)

其中步长 alpha 是一个较小的正数，从而： Delta x triangledown f(x) = - alpha (triangledown f(x)) 2 .

由于任何不为0的数的平方均大于0因此保证了 Delta x triangledown f(x) 0 .

从而，设定:

f(x+ Delta x) = f(x - alpha triangledown f(x)) ,

则可保证:

f(x+ Delta x) f(x)

那么更新 x 的计算方式就很简单了，可按如下公式更新 x

x leftarrow x - alpha triangledown f(x)

这就是所谓的沿负梯度方向走一小步。

到此为止，这就是梯度下降的全部原理。

支持(0) 反对(0)

87楼 楼主 2018-06-12 22 37 刘建平Pinard

__ 卢杰
写的挺好啊，我这里没写这些因为这是高数的基本知识了。
你回复里面的公式 都用两个 符号包起来就可以正常显示了。

支持(0) 反对(0)

88楼 2018-06-17 15 29 aaronwang123

你好博主，请教个小问题呢
在用小批量k个样本，或者单个(k=1)样本来训练的时候，

for steps1
for (i= 0, i m/k，i++)；m为样本总数
for steps2
w-=dw
end
end
end

这个思路对吗？也就是在每小批量多次迭代后尽量拟合好这个小批量样本后，在拟合下一小批量，把所有小批量都拟合完一篇。。。。然后再整体重复这个轮回。？

支持(0) 反对(0)

89楼 楼主 2018-06-17 16 33 刘建平Pinard

__ aaronwang123
你好！你的思路没问题。很多时候，我们一般是把所有小批量都拟合完一遍后就停止了。当然如果觉得收敛的不好，还可以继续整体重复轮回，甚至再进行不同的小样本划分后，在重复轮回。

支持(0) 反对(0)

90楼 2018-07-25 10 25 TimCNBlog

你好，3.3.2的公式1是不是采用的分母布局?
$\frac{\partial(\mathbf{X}\mathbf{X})}{\partial\mathbf{X}} = \frac{\partial\mathbf{X}^{\mathrm{T}}\mathbf{X}}{\partial\mathbf{X}} = 2\mathbf{X}$
$J(\theta) = \frac{1}{2}(\mathbf{X}\theta - \mathbf{Y})^{T}(\mathbf{X}\theta - \mathbf{Y})$ 是根据矩阵的意义写成这样的，还是根据上面公式分子的转换写的?
(ps 评论除了可以显示公式外，还可以显示颜色等其他latex操作嘛)

支持(0) 反对(0)

91楼 2018-07-25 11 35 TimCNBlog

__ 卢杰
引用
博主，您好，第三段的内容不是很理解，然后去知乎搜了一下，明白了。分析一下。
梯度下降法(gradient descent)是一种常用的一阶(first-order)优化方法，是求解无约束优化问题最简单、最经典的方法之一。我们来考虑一个无约束优化问题 min x f(x) , 其中 f(x) 为连续可微函数，如果我们能够构造一个序列 x

> 0, x 1, x 2, ... , 并能够满足:
>
> f(x t+1) f(x t), t=0,1,2,...
> 那么问题就是如何找到下一个点 x t+1 ,并保证 f(x t+1) f(x t) 呢? 假设我们当前的函数 f(x) ...

你好，这个评论里公式没有被编辑出来，看得很吃力，请问有原文链接吗?

支持(0) 反对(0)

92楼 2018-07-30 16 46 grace甜

老师您好:

如何理解梯度下降的代数形式的目标函数要求均值，但是最小二乘的代数形式的目标函数不求均值。但是它们的矩阵形式的目标函数却是一样的?

支持(0) 反对(0)

93楼 楼主 2018-07-31 10 41 刘建平Pinard

__ grace甜
你好，最小二乘法可以直接求全局最优解，自然也不需要求什么均值了。
但是梯度下降法是一种迭代的求解局部最优解的方法，为了控制梯度的方向是向我们希望的最小值方向，那么取样本的均值来计算是最合理的。当然样本量可以根据需要来修改。

目标函数，是我们优化的目标，在这里梯度下降法和最小二乘法目标是一致的，都是希望求得使目标函数最小化时候的模型参数。但是两者优化求模型参数的方法不同。

支持(0) 反对(0)

94楼 2018-08-08 10 00 会长

总结的很到位。那张图看着眼熟，貌似是达爷的图吧。请教楼主一个问题：步长和初始值的设定全凭反复试验和经验吗?有没有可以遵循的法则? 谢谢

支持(0) 反对(0)

95楼 楼主 2018-08-08 11 07 刘建平Pinard

__ 会长
你好，的确是Andrew Ng讲义里的图。

步长和初始值的设定基本靠经验。

对于步长一般可以先定一个固定值。发现算法效果不好再来尝试优化。有些类库比如sklearn，它自己会有自己的步长策略，所以没有这个参数给你调的。Tensor low之类的库则有这个参数给你调。

对于初始化，不同的算法一般有自己的较优的初始化策略。比如 -Means。这些需要具体算法具体分析。

支持(0) 反对(0)

96楼 2018-08-08 11 24 会长

__ 刘建平Pinard
谢谢回复!

支持(0) 反对(0)

97楼 2018-08-08 23 31 木易晚成

这个写得也太好了吧

支持(0) 反对(0)

最新IT新闻

自如C O熊林回应指责：房租涨幅远低于市场水平
关于小鹏汽车的五个重要问题，我们找到了答案
被质疑的红芯： 我们一点都不担心
微软为印度应用 uuh推出专属网站 无需第三方账号也能聊天
58速运 改名 快狗 司机集体到公司讨尊严：这是骂谁呢
更多新闻...

最新知识库文章

被踢出去的用户
成为一个有目标的学习者
历史转折中的 杭派工程师
如何提高代码质量?
在腾讯的八年，我的职业思考
更多知识库文章...