



华南理工大学

South China University of Technology

《机器学习》课程实验报告

学 院 软件学院

专 业 软件工程

组 员 张毓嘉

学 号 201530613689

邮 箱 3290196208@qq.com

指导教师 吴庆耀

提交日期 2017 年 12 月 14 日

1. 实验题目：逻辑回归、线性分类与随机梯度下降

2. 实验时间：2017 年 12 月 09 日

3. 报告人:张毓嘉

4. 实验目的:

1. 对比理解梯度下降和随机梯度下降的区别与联系。
2. 对比理解逻辑回归和线性分类的区别与联系。
3. 进一步理解 SVM 的原理并在较大数据上实践。

5. 数据集以及数据分析:

实验使用的是 LIBSVM Data 的中的 a9a 数据，包含 32561 / 16281(testing)个样本，每个样本有 123/123 (testing)个属性。

6. 实验步骤:

逻辑回归与随机梯度下降

1. 读取实验训练集和验证集。
2. 逻辑回归模型参数初始化，可以考虑全零初始化，随机初始化或者正态分布初始化。
3. 选择 Loss 函数及对其求导，过程详见课件 ppt。
4. 求得部分样本对 Loss 函数的梯度。
5. 使用不同的优化方法更新模型参数（NAG，RMSPProp，AdaDelta 和 Adam）。

6. 选择合适的阈值，将验证集中计算结果大于阈值的标记为正类，反之为负类。在验证集上测试并得到不同优化方法的 Loss 函数值， J_{train} ， J_{val} 和 J_{avg} 。
 7. 重复步骤 4–6 若干次，画出 J_{train} ， J_{val} 和 J_{avg} 随迭代次数的变化图。
-

线性分类与随机梯度下降

1. 读取实验训练集和验证集。
2. 支持向量机模型参数初始化，可以考虑全零初始化，随机初始化或者正态分布初始化。
3. 选择 Loss 函数及对其求导，过程详见课件 ppt。
4. 求得部分样本对 Loss 函数的梯度。
5. 使用不同的优化方法更新模型参数（NAG，RMSPProp，AdaDelta 和 Adam）。
6. 选择合适的阈值，将验证集中计算结果大于阈值的标记为正类，反之为负类。在验证集上测试并得到不同优化方法的 Loss 函数值， J_{train} ， J_{val} 和 J_{avg} 。
7. 重复步骤 4–6 若干次，画出 J_{train} ， J_{val} 和 J_{avg} 随迭代次数的变化图。

7. 代码内容：

（针对逻辑回归和线性分类分别填写 8-11 内容）

8. 模型参数的初始化方法：

随机初始化

9.选择的 loss 函数及其导数:

逻辑回归和随机梯度下降:

```
loss = -sum([y_train[i]*math.log(f(x_train[i, :], w, b))+(1-y_train[i])*math.log(1-f(x_train[i, :], w, b))for i in range(x_train_len)])/x_train_len
```

```
gradient_averange =  
(f(x_train[k, :], w, b)-y_train[k])*x_train[k, i]
```

线性分类和随机梯度下降:

```
loss = a*math.sqrt(sum([w[i] ** 2 for i in range(feature)]))  
+ sum([max(0, 1 - y_train[i] * f(x_train[i, :], w, b)) for i in range(x_train_len)])/x_train_len  
gradient_aver =  
w[i]+a*v[i]+(-y_train[k])*(f(x_train[k, :], w, b)*y_train[k]<1)  
)*x_train[k, i]
```

10.实验结果和曲线图: (各种梯度下降方式分别填写此项)

超参数选择:

逻辑回归和随机梯度下降:

NAG:

lr=0.3

a=0.7

Adadelata:

p=0.7

RMSProp:

lr=0.1

p=0.5

Adam:

p1=0.9

p2=0.99

lr = 0.0056

线性分类和随机梯度下降:

NAG:

lr=0.3

a=0.7

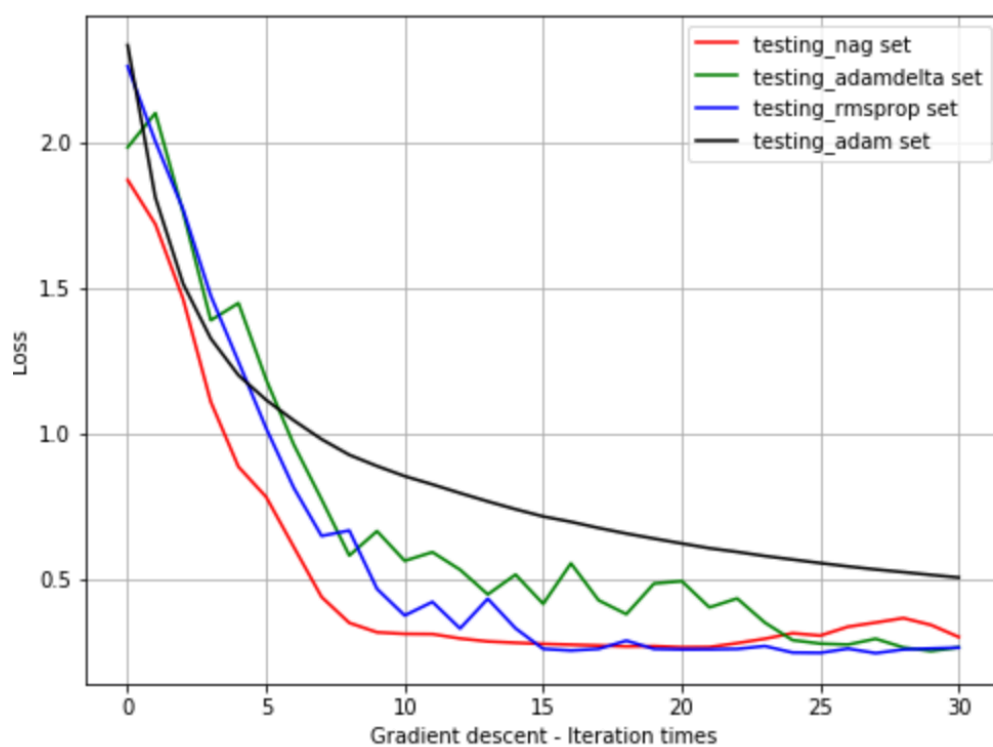
Adadelata:

lr=0.1
p=0.7
RMSProp:
lr=0.1
p=0.5
Adam:
p1=0.9
p2=0.99
lr = 0.0056

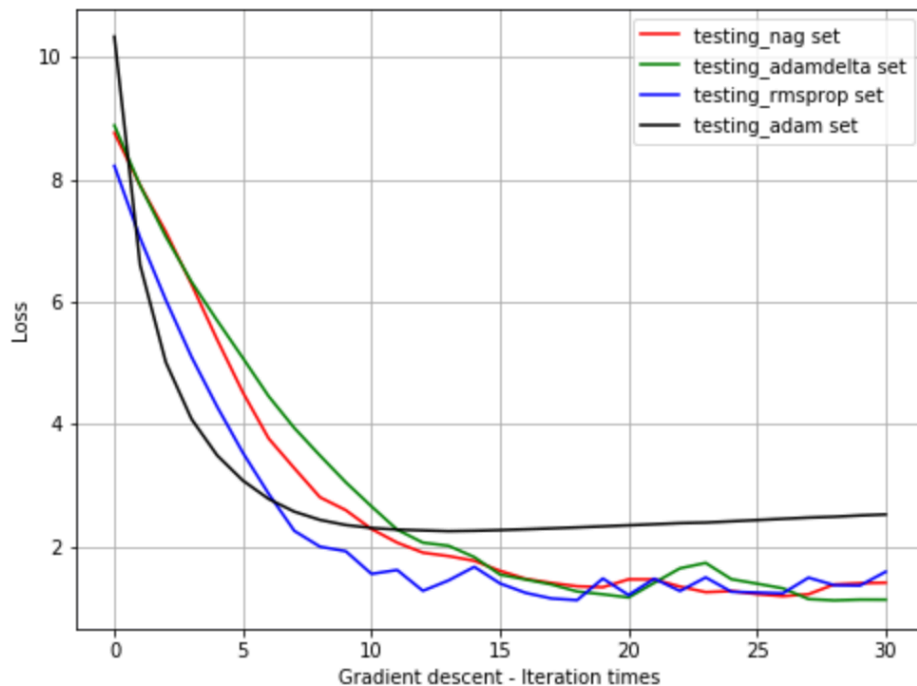
预测结果（最佳结果）：

loss 曲线图：

逻辑回归和随机梯度下降：



线性分类和随机梯度下降：



(adam 错误: 猜想: 可能由于 learning rate 小导致, 因为看到途中 adam 收敛速度比其他的快是符合常理的但是并不能达到一个最终的最佳结果)

11.实验结果分析:

可以看出, 在数据集属性非常多的时候采用随机梯度的方法是可以大幅度提升训练速度的, 同时, 使用不同的优化算法也可以加快优化速度。

其中:

对于普通的随即梯度下降算法,发现 loss function 会以不同的强度波动, 有助于我们发现更优的局部最小值。且容易收敛到局部最优。

对于 Adadelta 算法, 它不是累计之前所有的平方梯度, 而是将积累之前梯度的窗口先知道某个固定大小的 p 。在训练初期和中期加速效果不错, 后期收敛但在最小值附近抖动。

对于 adam 算法, 能计算每个参数的自适应学习率, 并且可以看出其 loss 变化比较平稳, 也就是其参数变化比较平稳。对内存需求较少, 为不同的参数计算不同的自适应学习率, 是目前比较优秀的算法。

12.对比逻辑回归和线性分类的异同点:

比较两者的 $f(x)$ 我们可以发现, 虽然都是分类但是前者的 y 值是在 0 和 1 中进行选择的, 而后者是在 -1 和 1 中进行选择。这就要求我们在进行逻辑回归的时候对数据集进行一定的处理。

比较两者的 loss function, loss function 是两者差距最大的地方但参数更新时的更新参数却是相同的, 也就是说不同的 loss function 对 w 的求导是相同的。

如果将线性分类的 lossfunction 用到逻辑回归中, 将会发现当 $y=1$ 时无论 f

(x) 取什么值, loss function 对 w 的求导都是 0, 导致训练结果不正确。

13.实验总结:

在本次实验中, 通过验证逻辑回归和线性回归(随即梯度方法)了解了两者的区别, 同时对于四种优化方法有了理解。虽然在这次实验中 adam 的方法验证失败, 其效果没有很好的展现出来, 根据论文进行调参之后认为可能是由于学习率的不合适导致, 但在几次调整学习率虽然曲线发生变化但是仍然没有得到最佳结果, 还需要继续调整。(基本效果可以看出, 学习率可能偏大)