**关于SVM梯度下降的一些思考**

以下是我昨天思考、实验和同学讨论得到的结论。

1. 为什么SVM梯度下降算法需要在Loss上增加正则项？
2. 从理论的角度来说，SVM的核心思想是最大化间隔，即：



而的提出显然是基于



此时如果只关注于，即



这里就抛弃了这个条件显然是不太好的。

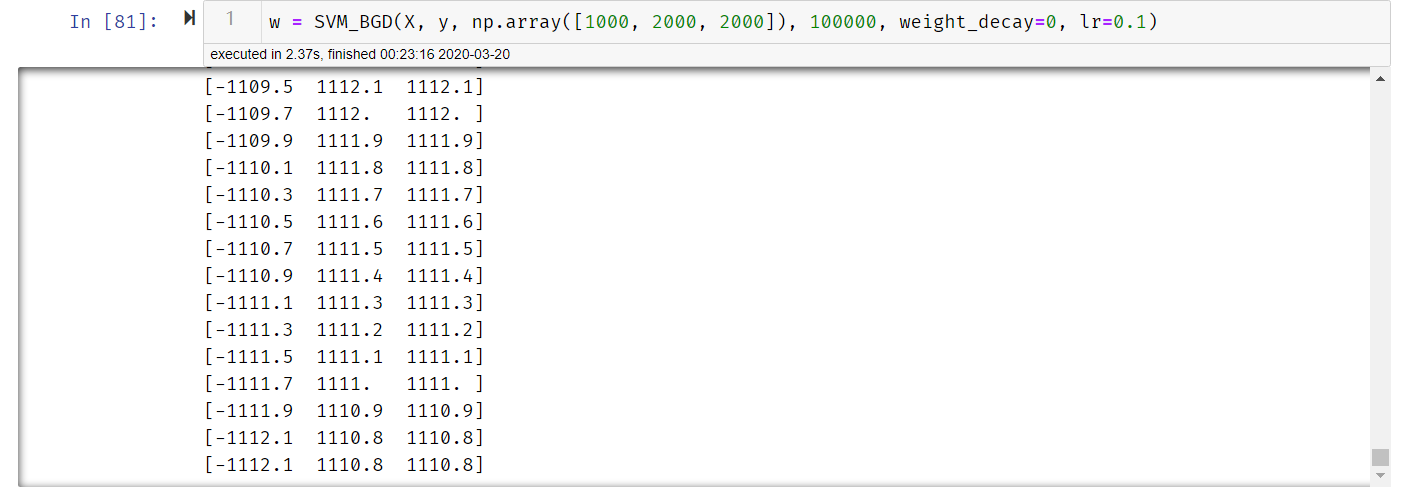
1. 从直观的角度而言，就算满足了为零或最小化但是却不能保证最大化间隔，原因是可以成倍提高和，使得仍然满足为零或最小化，但是真实的间隔距离却是由计算的，由于Hinge Loss的特点（当以后就不再有损失）此时最小的大概是一点几的水平，而却可以无限成倍增大，那么这样的话，这个间隔就趋于零，这样SVM跟感知机没什么区别，只要分对就可以了。而加入正则项，即

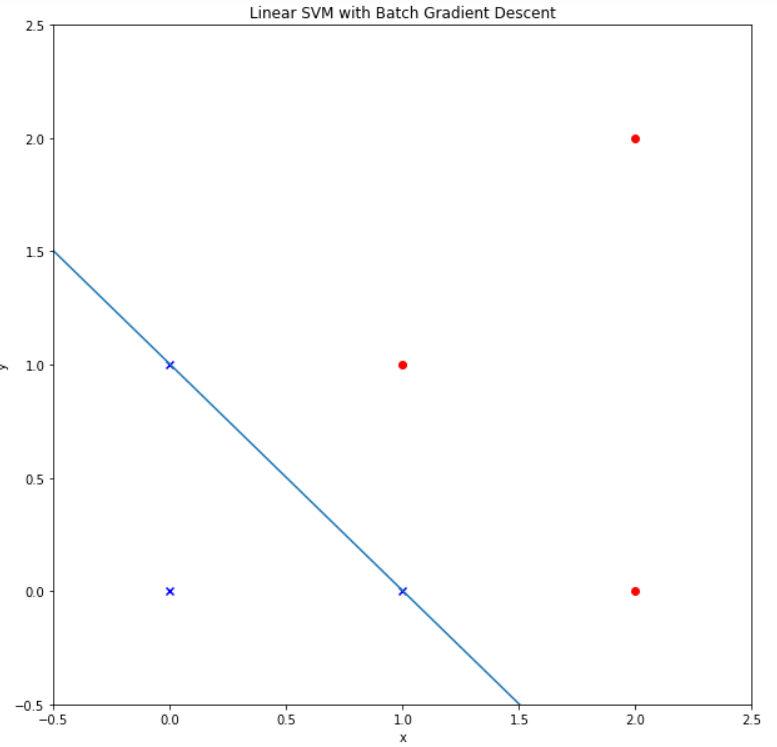


就可以有效抑制的无脑增大，根据，在满足时尽可能减小，这就正对应着最大化间隔。

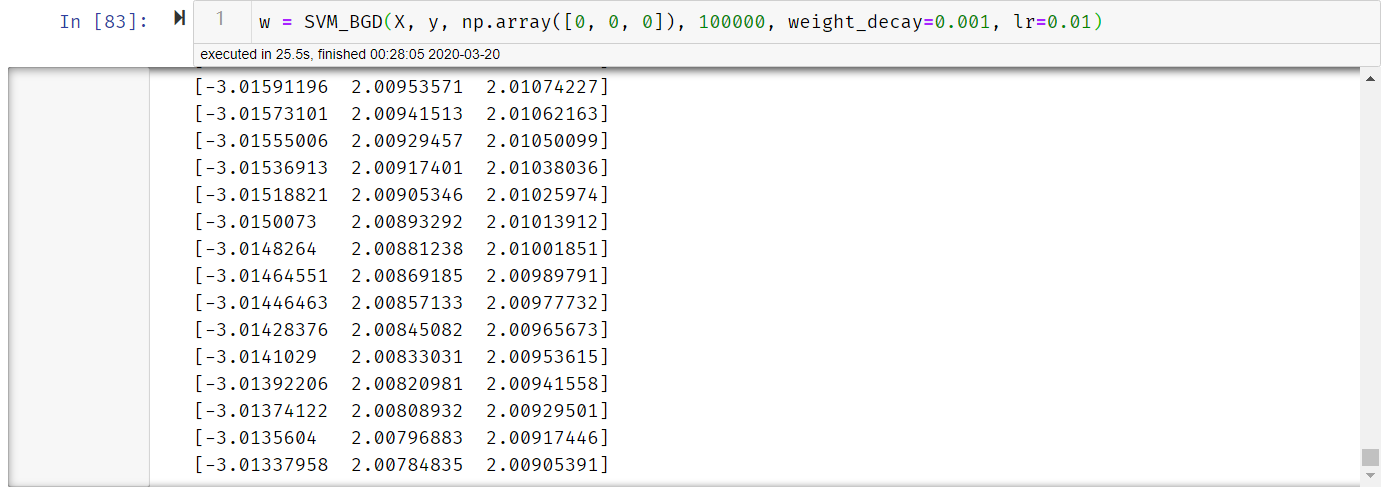
1. 为什么在习题里面不增加正则项却能做到正确答案呢？

个人猜测这是由于初值的选取，因为初值选取在，因此在梯度下降时，梯度的模长开始增加，恰好在就满足了Hinge Loss为零或最小化，迭代停止。假如我将初值选取到，可以看到，结果将收敛到，对应的分类结果如下图所示。这种情况下，损失为0，但是可以直观的看到间隔却极小，效果与感知机无异。

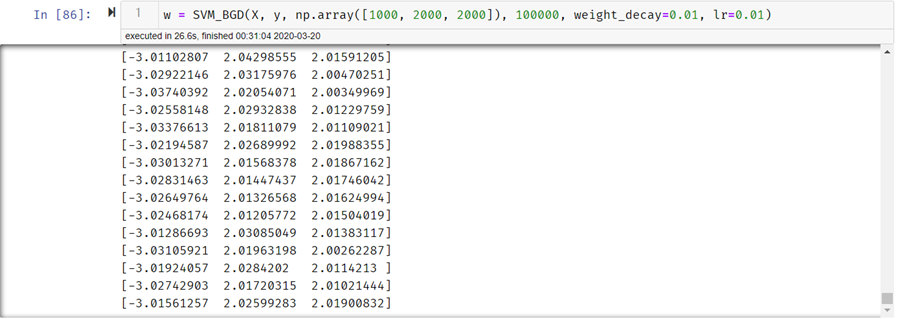




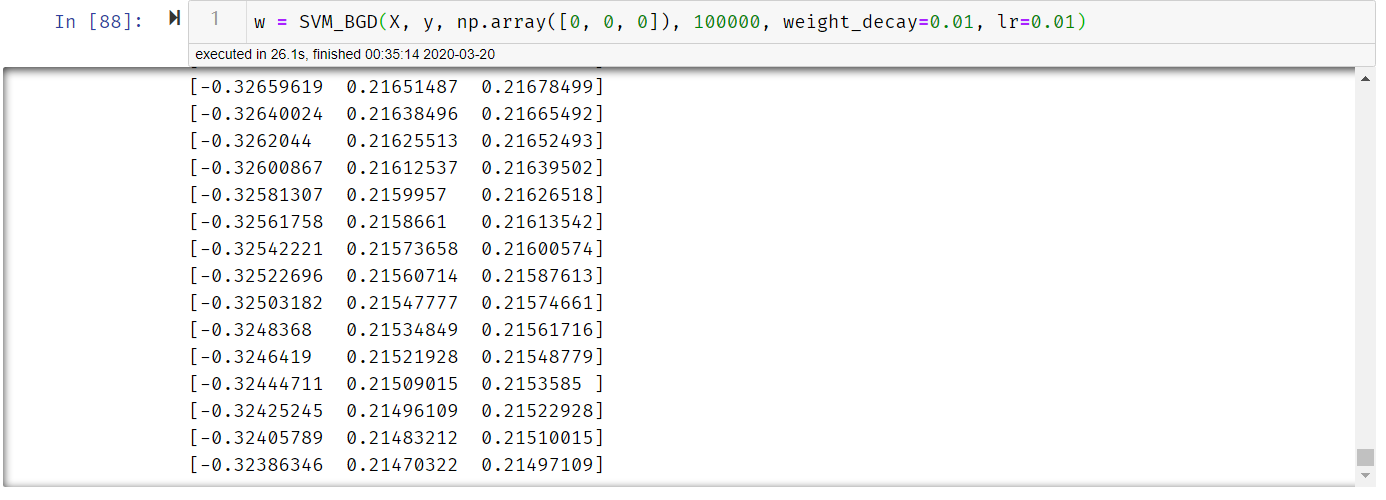
1. 实验探究（其中第三项是初值，即np.array(…)，第四项是迭代次数，weight\_decay是正则项系数）
2. 加入正则项显然是可以收敛到正确答案的

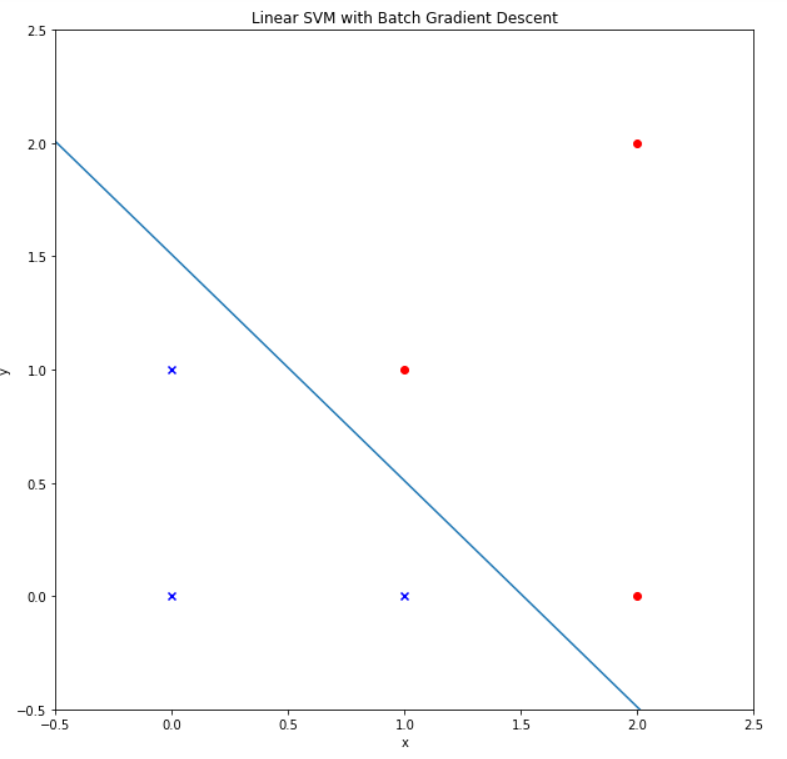


1. 下面更改初值，加入合适的正则项，可以看到收敛到正确答案

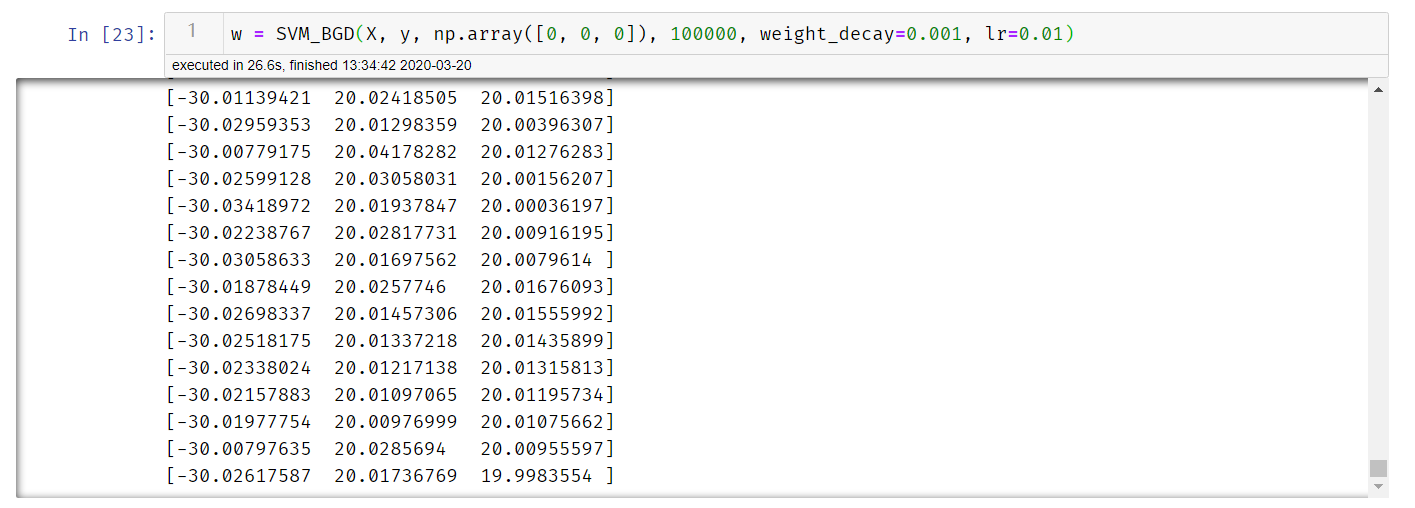


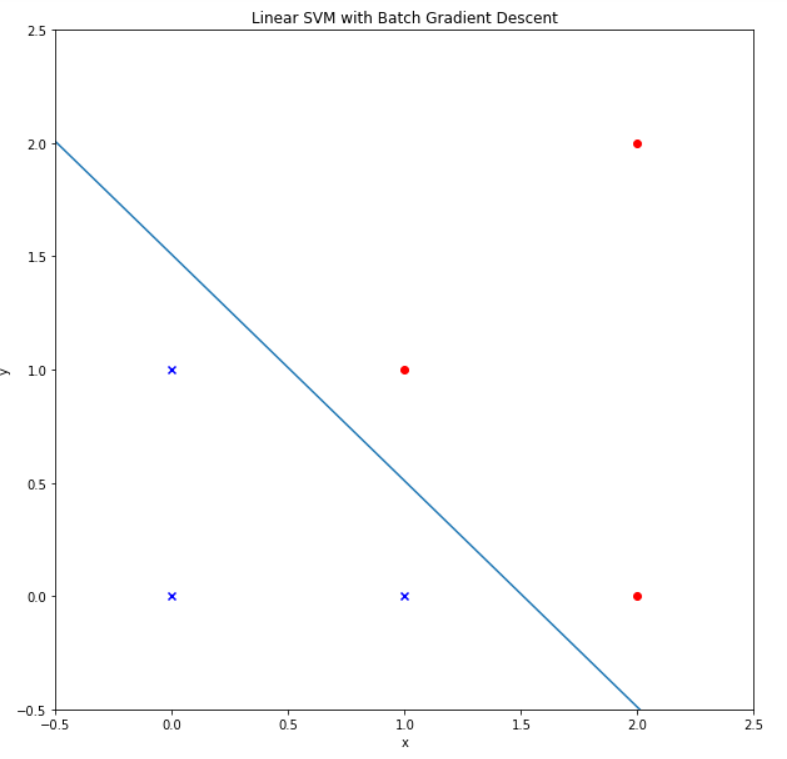
1. 下面探讨Hinge Loss中的1，个人感觉这个1的意义不大，正如课堂上所讲，倍乘并不改变直线，在这里也能体现。我更改Hinge Loss中的阈值为0.1，只要适当调节正则项系数仍然收敛，只不过收敛得到的权重改变了。但是仍然分的不错。



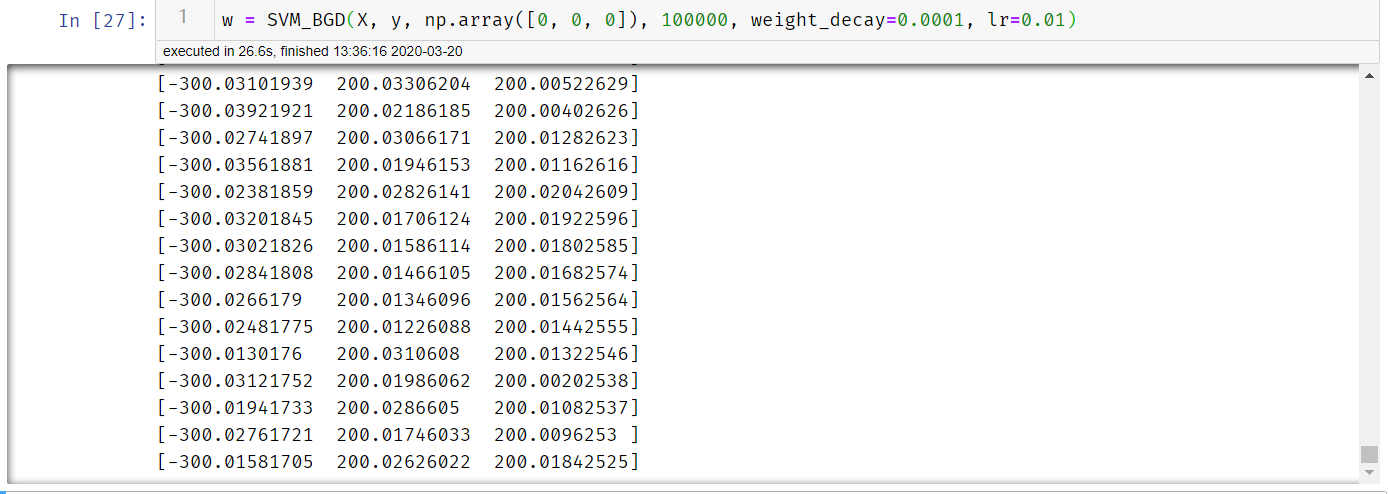


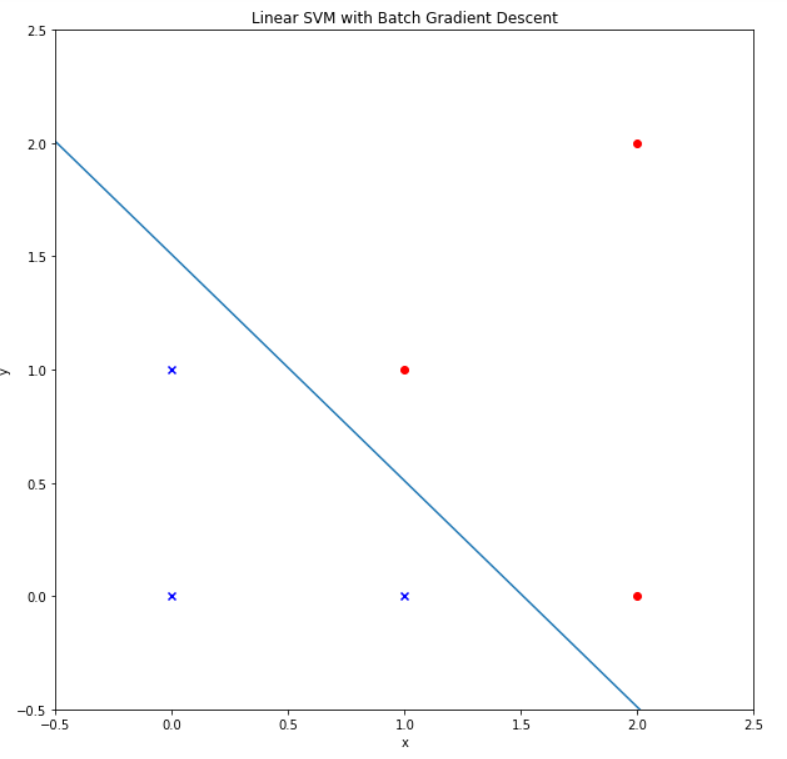
1. 下面我更改Hinge Loss中的阈值为10，只要适当调节正则项系数仍然收敛





1. 下面我更改Hinge Loss中的阈值为100，只要适当调节正则项系数，仍然可以收敛





1. 综上所述
2. SVM梯度下降应该添加正则项，以最大化间隔。
3. 加入正则项后，初值对于最终收敛的结果几乎没有影响，但是如果没加入正则项，则初值对于最终收敛的结果有很大影响，初值选的好，是可以恰好卡到正确答案，但是选的不好，就会收敛到较差的结果。
4. Hinge Loss中的阈值没有多大意义，他是调整最终收敛得到的权重值的范围，例如如果选作1，则最终权值收敛到，而如果选作100，则最终权值收敛到，但是最终的直线是同一条直线。
5. 正则项系数选取也十分重要，如果选取不当也将得不到正确结果。