Web data mining部分：

第二章讨论部分：

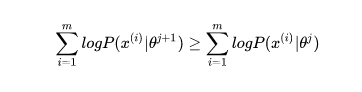
读书报告内容：

1. （必填）自己提出的问题的理解（罗列全部）：
2. 提出的问题1：

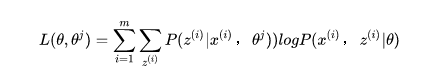
Em算法能够保证必定收敛嘛？

讨论后的理解：

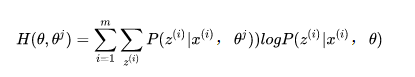
要证明 EM 算法收敛，则我们需要证明我们的对数似然函数的值在迭代的过程中一直在增大。即：

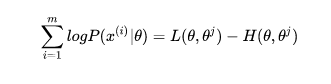


由于：

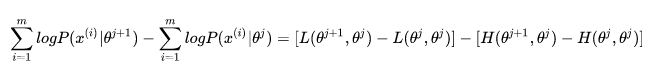


令：



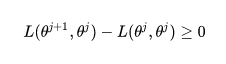
上两式相减得到：

在上式中分别取  为  和 ，并相减得到：

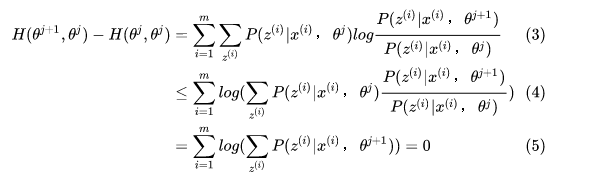


要证明EM算法的收敛性，我们只需要证明上式的右边是非负的即可。

由于使得IMG_265极大，因此有：



而对于第二部分，我们有：



其中第（4）式用到了Jensen不等式，只不过和第二节的使用相反而已，第（5）式用到了概率分布累积为1的性质。

至此，我们得到了：



 ，证明了EM算法的收敛性。

1. 提出的问题2：

自学习算法因为自己训练自己，会不会出现严重的过拟合结果？

讨论后的理解：

这是正常的。因为这是最早的半监督学习算法，没有假设，效果较差。我在讨论中也提出了一些想法尝试更多利用labeled data，如像评估分类器时取验证集一样，多次选取不同部分的labeled data，但都被指出了一些问题，因而还是认为要从利用数据量更多的unlabeled data着手。

1. （必填）别人提出的问题的理解（选择几个问题罗列，并给出理解）：

问题3：对于自学习算法，本身具有一定的分类能力，但是通过这种已经存在的分类能力去增加自己的训练集规模，如果分类错误的话，不就会产生“一错再错”的问题吗？

自己的理解：

这个问题其实和我提的问题属于两个极端。因为如果我们给他一个较高的置信度阈值的时候，可能会出现过拟合的情况，让分类器的结果过分和初始训练集相似，如果设置一个较低的阈值，可能会出现一步错，步步错的情况。就是把本不正确的数据放入了训练集，相当于认为增加了噪声，并且由于算法本身迭代，会导致分类器一错再错。

问题4：监督训练中的svm能够保证最后将两个种类的数据分开，但是SVM使用已经标注的正反例来产生决策边界，可能会产生未标注的点出现在决策边界上的情况，这种情况要如何处理？

自己的理解：

这个问题其实可以将这些在边界上的数据直接舍去，或者随机给他打上label，因为本来这样的数据是极其稀少的，并不会给之后的训练集带来太大影响，并且本身这样的数据就是符合任意一个分类的。

问题6：co-Training算法同时使用两个分类器的意义是什么？

自己的理解：

其实使用两个分类器的意义在于，我们本身已经打上label的data实在是太少了，并没有足够的先验数据，所以要先用训练集把ublabeled data中已经非常确定的data放入训练集，用f2对他训练，再次循环，放入训练集，这样可以增加训练集的数据量，而不影响本来的效果。

问题6：co-Training算法中，假如在某一轮f1f2将分类结果加入L的时候，某一个元素在f1中被标为正标记，又在f2被选中标为负标记，那算法会如何应对这种冲突呢？

自己的理解：

这种冲突实际上本身是不可能存在的，因为要先用训练集把ublabeled data中已经非常确定的data放入训练集，用f2对他训练，再次循环，放入训练集，这样可以增加训练集的数据量，而不影响本来的效果。这样f1训练出来的data会放入训练集，并且在测试集中已经删除了这个data，也即任意一个data至多只会被f1和f2打上一次标签。

1. （必填）读书计划
2. 本周完成的内容章节：

（1）5.1完成并且梳理知识点，寻找问题，自己思考并且于小组会之前完成了自己的思考。

1. 西瓜书295半监督学习中有更详细的讲解配合李航的第九章em算法也有具体介绍。同时参考了知乎上的文章，链接如下：

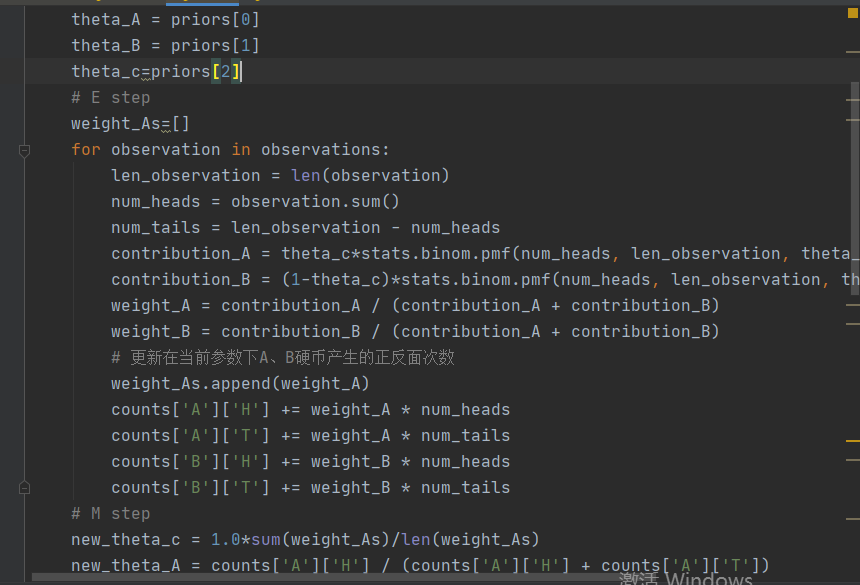
https://zhuanlan.zhihu.com/p/36331115?utm\_source=wechat\_session&utm\_medium=social&utm\_oi=933847744108421120&utm\_content=sec讲解em算法非常的好理解

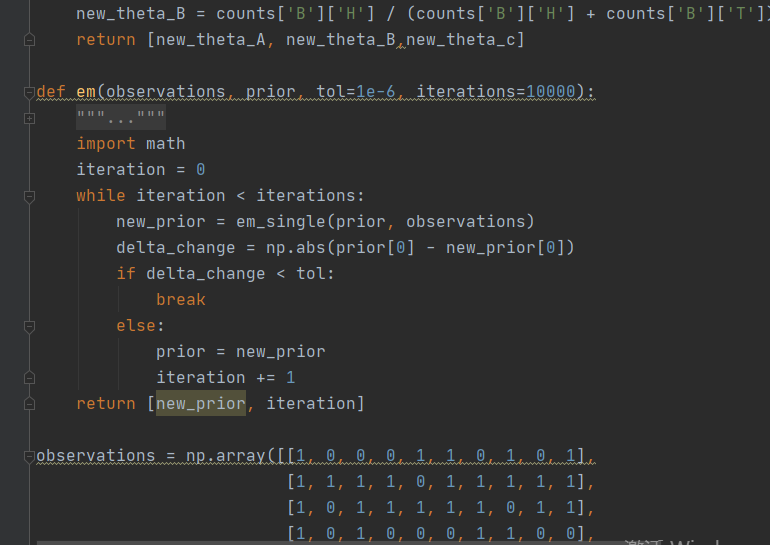
2、下周计划：5.1的阅读.

四、（选做）读书摘要及理解或伪代码的具体实现（读书摘要、伪代码的具体实现代码等可以写到这个部分）

1.代码实现

Em算法的代码实现找到频繁集：





输出结果：

[[0.51392121603987106, 0.79337052912023864, 0.47726196801164544], 42]

1. 读书摘要

EM的意思是“Expectation Maximization”，具体方法为：

举例，男女生的身高都符合正太分布。

先设定男生和女生的身高分布参数(初始值)，然后计算出每个人更可能属于第一个还是第二个正态分布中的（例如，这个人的身高是180，那很明显，他极大可能属于男生），这个是属于Expectation 一步；

我们已经大概地按上面的方法将这 200 个人分为男生和女生两部分，我们就可以根据之前说的极大似然估计分别对男生和女生的身高分布参数进行估计（这不变成了极大似然估计了吗？极大即为Maximization）这步称为 Maximization；

然后，当我们更新这两个分布的时候，每一个学生属于女生还是男生的概率又变了，那么我们就再需要调整E步；

……如此往复，直到参数基本不再发生变化或满足结束条件为止。

上面的学生属于男生还是女生我们称之为隐含参数，女生和男生的身高分布参数称为模型参数。

EM 算法解决这个的思路是使用启发式的迭代方法，既然我们无法直接求出模型分布参数，那么我们可以先猜想隐含参数（EM 算法的 E 步），接着基于观察数据和猜测的隐含参数一起来极大化对数似然，求解我们的模型参数（EM算法的M步)。由于我们之前的隐含参数是猜测的，所以此时得到的模型参数一般还不是我们想要的结果。我们基于当前得到的模型参数，继续猜测隐含参数（EM算法的 E 步），然后继续极大化对数似然，求解我们的模型参数（EM算法的M步)。以此类推，不断的迭代下去，直到模型分布参数基本无变化，算法收敛，找到合适的模型参数。

一个最直观了解 EM 算法思路的是 K-Means 算法。在 K-Means 聚类时，每个聚类簇的质心是隐含数据。我们会假设 K 个初始化质心，即 EM 算法的 E 步；然后计算得到每个样本最近的质心，并把样本聚类到最近的这个质心，即 EM 算法的 M 步。重复这个 E 步和 M 步，直到质心不再变化为止，这样就完成了 K-Means 聚类。