第七周工作报告

丘骏鹏

2013-04-12 Fri

目录

1 本周工作

- 优化原有的算法，做了很多的近似，目前运行时间从 2s 降到了 0.1 到 0.2s。主要的方法是：

  1. 去掉了结束标签，只保留开始标签，这一步速度提高很多，因为目前 LCS 算法的复杂度是 O(mn)，所以如果去掉结束标签理论上复杂度变成$\frac{1}{4}$。
  2. 将字符串做了 hash，避免字符串比较，直接比较 int 数值。这一步速度提升有限。
  3. 做了空间上的一些优化，空间复杂度降到 O(2*m)，这主要是考虑到之后如果载入很多文档，内存占用也会很大，做一些这方面的优化也是有必要的。

  虽然目前运行速度加快了 10 倍左右，但是感觉提高的空间不会太大了（至少从算法上来说）。如果从语言角度考虑的话，可能的方法是使用 C 语言实现一个 LCS 算法，然后用 jni 来调用，可能可以加快一些执行速度（是否有效取决于 jni 调用多花的时间和 C 程序执行加速的时间之间的大小关系）

- 学习了 Actor 模型，基于akka 库实现了一个简单的并行计算的框架。Actor 模型中每个 actor 是一个基本的计算单元，互相之间通过传递异步的消息来进行决策，是比线程、锁这些并行计算的原语更高层次的一个轻量级的抽象并行计算模型。akka 库目前工业上也有较广泛的应用，其中包括 amazon 这些大的公司。我打算采用这个并行框架实现多线程的计算。

2 目前的困难

1. 拿最小的博客集合来做实验的话，共 60000 篇文档。假设目前采用一种简单的聚类算法，只需要每两篇文档计算一次距离即可，时间是$60000^2/2 * 0.1/3600$ 小时，远超可

以承受的范围，所以必须采用并行的计算方法。如果有 20 台计算机可以同时计算，每台上可以开 10 个线程，实际需要的时间为大概 5000 小时。

和郝老师讨论的方案是先将样本数减少，因为这是和样本个数平方成正比的，所以效果会很明显，但是会不会数据量一小就和选题的着重点（大数据量）有些偏差？

也和房磊学长讨论了一下，他说暂时没有想到很好的解决方案，不过可以放到 hadoop 上做。

我想大部分的聚类算法可能都需要至少计算一次两两的距离，如果可以显著减少计算相似度的次数会提高算法的执行速度，不知道老师能不能提供关于这方面的算法的一些参考（或者组里有没有师兄师姐做过类似的工作）？除了减少样本量和计算次数外，只能将计算相似度的方法再进行简化。实际在做了多次简化后，目前这个 LCS 已经算挺粗糙的了，我觉得如果采用 jaccard 或者转化为一个 tag vector 计算欧几里得距离之类的方法就可能太过于简单了（不过时间也可以显著减小）。

2. 聚类的意义。取了些样本做实验，发现实验结果差异较大的是 404 网页和详细页，意义不大。

   关于聚类的意义，我的看法是这样的：大数据的优点是可以让我们更好地发现“共同点”，但是为了计算速度，我们可能采取较粗糙的算法，这样会导致发现出来的“共同点”可能会很稀疏，也就是说模板会很简单。因此如果网页的模板差异不是特别大，差异部分可能在这些稀疏的模板体现不明显，也就很难通过聚类区分开来。实际上因为我们不知道聚出来的类的个数，我们只有通过控制阈值的方法来控制聚类的过程，能分多少类也会取决于我们的阈值是如何取的，如果阈值设置不合适，可能就会有不合理的结果。但这个阈值取多少合适本身也很难决定，没有一个好的方法衡量多大的阈值是好的。

   另外，进行聚类需要耗费的计算资源这么大，聚类结果实际上可能只会有少量甚至只有 1 类，而且还会因为不同的阈值和聚类的方法存在诸多不确定性，因此这样的做法是否真正切合实际的需求？从实用角度考虑，在我们这个系统中这样做是不是有些复杂了？