Computer Architecture

Project 2: Parallel Label Propagation

Project due: 31 May, 23:59pm

1 项目内容描述

本项目内容是实现基于标签传播的半监督学习算法。众所周知，机器学习可以大体分为三大类：监督学习、非监督学习和半监督学习。监督学习可以认为是，我们有非常多的标注（labeled）数据来训练一个模型，期待这个模型能学习到数据的分布，以期对未来没有见到的样本做预测。那这个性能的源头——标注数据，就显得非常感觉。一般情况下，必须有足够的训练数据，以覆盖真正现实数据中的样本分布才可以，这样学习到的模型才有意义。那非监督学习就是没有任何的标注数据，就是平时所说的聚类了，利用他们本身的数据分布，给他们划分类别。而半监督学习，顾名思义就是处于两者之间的，只有少量的标注数据，我们试图从这少量的标注（labeled）数据和大量的非标注（unlabeled）数据中学习到有用的信息。

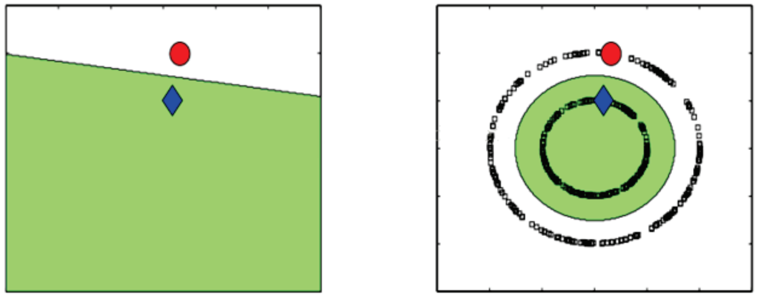
1.1 半监督学习

半监督学习（Semi-supervised learning）发挥作用的场合是：你的数据有一些有 label，一些没有。而且一般是绝大部分都没有，只有少许几个有 label。半监督学习算法会充分的利用 unlabeled 数据来捕捉我们整个数据的潜在分布。它基于三大假设：

1）Smoothness 平滑假设：相似的数据具有相同的 label。
2）Cluster 聚类假设：处于同一个聚类下的数据具有相同 label。
3）Manifold 流形假设：处于同一流形结构下的数据具有相同 label。

例如下图，只有两个 labeled 数据，如果直接用他们来训练一个分类器，例如 LR 或者 SVM，那么学出来的分类面就是左图那样的。如果现实中，这个数据是右图那边分布的话，显而易见，左图训练的这个分类器非常的不可信。因为 labeled 训练数据太少，都没办法覆盖我们未来可能遇到的情况。但是，如果右图那样，把大量的 unlabeled 数据（黑色的）都考虑进来，有个全局观念，那么，半监督算法会把大圈的数据都归类为红色类别，把内圈的数据都归类为蓝色类别，发现原来数据是两个圆形的流形。在实践中，labeled 数据是昂贵，很难获得的，但 unlabeled 数据则非常容易获取。因此，如果能充分利用大量的 unlabeled 数据来辅助提升我们的模型学习，就有非常大的价值。

Figure 1: Unlabeled Data and Prior Beliefs



半监督学习的算法有很多，我们需要实现的是最简单的标签传播算法（label propagation）。

2 标签传播算法

标签传播算法（label propagation）的核心思想非常简单：相似的数据应该具有相同的 label。LP 算法包括两大步骤：1）构造相似矩阵；2）传播标签。

LP 算法基于 Graph，因此需要为所有的数据构建一个图，图的节点就是一个数据点，包含 labeled 和 unlabeled 的数据。节点 i 和节点 j 的边表示他们的相似度。图的构建方法有很多，这里我们假设这个图是全连接的，节点 i 和节点 j 的边权重为

$$w_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\alpha^2}\right)$$

这里 α 是超参数。knn 图是更加简单的方法，也就是只保留每个节点的 k 近邻权重，其他的为 0，也就是不存在边，因此是稀疏的相似矩阵。

标签传播算法通过节点之间的边传播 label。边的权重越大，表示两个节点越相似，那么 label 越容易传播过去。我们定义一个 NxN 的概率转移矩阵 P：

$$P(i \to j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{ik}}$$

P_{ij} 表示从节点 i 转移到节点 j 的概率。假设有 C 个类和 L 个 labeled 样本，我们定义一个 LxC 的 label 矩阵 YL，第 i 行表示第 i 个样本的标签指示向量，即如果第 i 个样本的类别是 j，那么该行的第 j 个元素为 1，其他为 0。同样，我们也给 U 个 unlabeled 样本一个 UxC 的 label 矩阵 YU。把他们合并，我们得到一个 NxC 的 soft label 矩阵 F=[YL;YU]。soft label 的意思是，我们保留样本 i 属于每个类别的概率，而不是互斥性的，这个样本以概率 1 只属于一个类。当然了，最后确定这个样本 i 的类别的时候，是取 max 也就是概率最大的那个类作为它的类别的。F 里面的 YU 的初值值可以用]0-1]的随机数设置。

得到标签传播算法如下：

1）执行传播：F=PF；

2）重置 F 中 labeled 样本的标签：FL=YL；

3）重复步骤 1）和 2）直到 F 收敛。

步骤 1）就是将矩阵 P 和矩阵 F 相乘，这一步，每个节点都将自己的 label 以 P 确定的概率传播给其他节点。如果两个节点越相似（在欧式空间中距离越近），那么对方的 label 就越容易被自己的 label 赋予，就是更容易拉帮结派。步骤 2）非常关键，因为 labeled 数据的 label 是事先确定的，它不能改变，所以每次传播完，它都得回归它本来的 label。随着 labeled 数据不断的将自己的 label 传播出去，最后的类边界会穿越高密度区域，而停留在低密度的间隔中。相当于每个不同类别的 labeled 样本划分了势力范围。

进一步地，我们将矩阵 P 做以下划分：

$$P = \begin{bmatrix} P_{LL} & P_{LU} \\ P_{UL} & P_{UU} \end{bmatrix}$$

我们可以得到

$$f_U \leftarrow P_{UU} f_U + P_{UL} Y_L$$

一般地，迭代上面这个步骤直到收敛就行了：

$$f_U = \lim_{n\to\infty} (P_{UU})^n f_U^0 + \left(\sum_{i=1}^{n} (P_{UU})^{(i-1)} \right) P_{UL} Y_L$$

可以看到 FU 不但取决于 labeled 数据的标签及其转移概率，还取决了 unlabeled 数据的当前 label 和转移概率。因此 LP 算法能额外运用 unlabeled 数据的分布特点。

这个算法的收敛性也非常容易证明是可以收敛到一个凸解：

$$f_U = (I - P_{UU})^{-1} P_U$$

所以也可以直接这样求解，以获得最终的 YU。但是在实际的应用过程中，由于矩阵求逆需要 $O(n^3)$的复杂度，所以如果 unlabeled 数据非常多，那么 I – PUU 矩阵的求逆将会非常耗时，因此这时候一般选择迭代算法来实现。

3 评价机制

3.1 评分

我们采用 http://cs.joensuu.fi/sipu/datasets/ 的 Shape sets 数据进行验证分类验证，包括 http://cs.joensuu.fi/sipu/datasets/Aggregation.txt 等 8 个数据。你可以设置不同个数的初始 labelled 数据，并将剩下的数据作为测试数据。请在你的实验报告里面说明详细的实现情况以及取得的结果。

3.2 并行算法的加速比

这部分的评价标准是利用并行算法优化后获得的加速比，加速比定义为 $R = T0 / TP$，这里 TP 为并行算法使用的时间，T0 为未优化的算法使用时间。此部分评分占总分之 70%。

4 代码算法

本次 Project 允许使用任语言进行编写。

5 于指定时间前提交至 yyliang@must.edu.mo。