MapReduce 开题报告

周华平，蒋雅楠，高翼枭

2018 年 2 月 27 日

小组信息

学号	姓名	邮箱
MG1733098	周华平	zhp@smail.nju.edu.cn
MF1733026	蒋雅楠	mf1733026@smail.nju.edu.cn
DZ1733004	高翼枭	dz1733004@smail.nju.edu.cn

课题分工

本课题的选题和讨论由三人共同完成；项目的代码部分将包含多个 MapReduce Job，预计在定义好输入输出格式并完成 MapReduce Job 的初步划分后分工完成；最终的设计报告将根据完成课题时的具体分工情况具体安排。

研究题目

我们研究的题目为“基于近邻成分分析的距离度量学习算法的并行化”。该题目来源于实现高级机器学习作业的过程中遇到的实际问题：在样本数据量较大或者维度较高的情况下，由于计算和存储开销较大，在单机上无法执行全梯度下降，因此该算法需要使用随机梯度下降代替常规的梯度下降来完成计算，这可能会导致算法无法正确收敛。

因此，在本课题中我们计划使用 MapReduce 对 NCA 算法进行并行化，使得该算法能够完整地运行在在较高维度和较大规模的数据集上，并且能够缩短训练时间。此外，我们还将用 MapReduce 实现 KNN 最近邻分类算法，来对距离度量学习的结果进行验证。

研究问题背景

在机器学习领域中，如何选择合适的距离度量准则一直都是一个重要而困难的问题。因为度量函数的选择非常依赖于学习任务本身，并且度量函数的好坏会直接影响到学习算法的性能。为了解决这一问题，我们可以尝试通过学习得到合适的度量函数。距离度量学习 (Distance Metric Learning, DML) 的目标是学习得到合适的度量函数，使得在该度量下更容易找出样本之间潜在的联系，进而提高那些基于相似度的学习器的性能。

在本课题中, 我们采用近邻成分分析 (Neighbourhood Component Analusis, NCA) 来实现距离度量学习,

度量函数学习目标

根据马氏距离的定义

$$dist^2_{mah}(x,y) = (x-y)^\top Q(x-y) = (Ax-Ay)^\top (Ax-Ay)$$

其中 Q 称为 "度量矩阵", 而 DML 则是对 Q 的学习。为了保持距离非负且对称, Q 必须是 (半) 正定对称矩阵, 即必有正交基 A 使得 Q 能写为 $Q = AA^\top$。

为了提高近邻分类器的性能, 我们将 Q 直接嵌入到近邻分类器的评价指标中去, 通过优化该性能目标相应地求得 Q。在本实验中我们采用近邻成分分析进行学习。

近邻分类器在进行判别时通常使用多数投票法, 领域中的每个样本投 1 票, 领域外的样本投 0 票。NCA 将其替换为概率投票法, 对于任意样本 x_j, 它对 x_i 分类结果影响的概率为

$$p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k\neq i}\exp(-\|Ax_i - Ax_k\|^2)}, \qquad p_{ii} = 0$$

若以留一法正确率的最大化为目标, 则可计算 x_i 的留一法正确率, 即它被自身之外的所有样本正确分类的概率为

$$p_i = \sum_{j\in C_i} p_{ij}$$

其中 $C_i = \{j|c_i = c_j\}$, 即与 x_i 属于相同类别的样本的下标集合。于是, 整个样本集上被正确分类的点的个数的期望为

$$f(A) = \sum_i \sum_{j\in C_i} p_{ij} = \sum_i p_i$$

NCA 的优化目标是使得 $f(A)$ 最大化, 即

$$\max_A \sum_i \sum_{j\in C_i} \frac{\exp(\|Ax_i - Ax_j\|^2)}{\sum_{k\neq i}\exp(\|Ax_i - Ax_k\|^2)}$$

优化算法

梯度下降法可以用来求解目标函数: 通过求 f 对 A 的偏导, 可以得到梯度公式 (令 $x_{ij} = x_i - x_j$)

$$\frac{\partial f}{\partial A} = -2A\sum_i \sum_{j\in C_i} p_{ij}(x_{ij}x_{ij}^\top - \sum_k p_{ik}x_{ik}x_{ik}^\top)$$

根据该公式, 使用梯度下降法即可求解 NCA 的目标函数。得到最大化近邻分类器留一法正确率的距离度量矩阵 Q。

主要技术难点和拟解决的问题

在高级机器学习课程中, 我们使用了 Python 实现了该算法。在该实现中, 为了尽可能利用 numpy 高效的矩阵操作, 我们需要将中间结果的计算尽可能转化为矩阵的运算, 从而

提高并行度。然而当样本的维度较高或者样本数据量较大时，无论是存储中间结果矩阵还是计算梯度的开销都会大到单机无法承受的程度。

因此，在实际实验中我们会用随机梯度下降来代替全梯度下降，即以计算梯度中的某些项来代替梯度全体，以此降低计算和存储开销。然而随机梯度下降并不是保证迭代将沿着目标函数的最快下降方向前进，甚至不保证其沿着下降方向前进。这有可能导致算法收敛过慢、无法正确地收敛、以及在接近最优解附近时精度较差等问题。

在本课题中我们计划使用 MapReduce 来完成 NCA 算法中梯度下降的并行化，使其在能够处理维度较高、数据量较大的样本训练的同时缩短训练时间。

该算法需要使用多个 MapReduce 作业，通过迭代的方式完成计算。对 $\frac{\partial f}{\partial A}$ 的计算分为了几个阶段，我们需要将其中的计算抽象为若干个 MapReduce 过程，合理地设计每个阶段的输入和输出，并将不同阶段组织成具有依赖关系的任务链；在中间结果的计算中还需要涉及多数据源的连接。简单来说，我们首先需要利用 MapReduce 并行地计算 $\exp(\|Ax_i - Ax_j\|^2)$；接着将该结果作为输入，并行地计算 p_{ij} 和 p_i；利用上述中间结果，我们可以通过 DataJoin 来计算 $p_{ij}x_{ij}x_{ij}^\top$；最后计算出当前的 $\frac{\partial f}{\partial A}$ 并更新矩阵 A，完成一次迭代。

基本解决方法和设计思路

NCA 模型训练

对于 NCA，我们拟采用组合式 MapReduce 计算作业来实现。由于梯度下降法需要用迭代方法求得逼近结果，因此在 NCA 的主控程序中，需要使用一个循环来控制 MapReduce 作业的执行，直到第 n 次迭代后结果与第 $n-1$ 次的结果小于某个指定的阈值时结束，或者通过经验值可确定在运行一定的次数后能得到接近的最终结果，也可以控制循环固定的次数。

对于梯度下降中需要求解的变量，我们采用顺序组合式 MapReduce 作业来依次计算：首先我们可以通过 DataJoin 对集合 X 做笛卡尔积，进而可以计算出 x_{ij}；在此基础上以 x_{ij} 作为输入，我们可以进一步计算出中间结果 $\exp(\|Ax_i - Ax_j\|^2)$，其中矩阵 A 作为 Distributed Cache 在 Mapper 的 setup() 阶段读入；基于以上结果，我们可以通过 MapReduce 对所有 i 计算出 $\sum_{k\neq i}\exp(\|Ax_i - Ax_k\|^2)$，通过 DataJoin 连接上述两个中间结果，我们可以计算出 p_{ij}。

p_i 的计算需要考虑 x_j 与 x_i 的 label 是否相同。我们可以在上述的每个中间结果后加上 x_i 与 x_j 的 label，并通过 Mapper 过滤掉与 $x_i$label 不同的元素，在 Reducer 端只需做简单的求和即可。

基于以上数据，我们可以对梯度 $\frac{\partial f}{\partial A}$ 进行求解。我们首先使用 DataJoin 计算出中间结果 $p_{ij}x_{ij}x_{ij}^\top$，然后使用不同的 Mapper 过滤元素，分别计算 $\sum_k p_{ik}x_{ik}x_{ik}^\top$ 以及 $\sum_{j\in C_i} p_{ij}x_{ij}x_{ij}^\top$，需要注意的是由于这两者的计算没有依赖关系，所以可以通过配置 Job 使它们并行执行。在计算出以上两个求和的结果后，梯度 $\frac{\partial f}{\partial A}$ 可以通过简单的操作得出，这里不做赘述。在每次迭代的最后，我们利用当前位置的梯度对矩阵 A 进行更新，并启动下一次迭代。

KNN 最近邻分类算法

为了验证 NCA 模型训练的结果，我们将使用 KNN 来进行分类预测。关于 KNN 最近邻分类算法，在书上已经有了比较详细的介绍，并且也给出了 MapReduce 的实现方法。我们对其所做的改动是使用 NCA 训练出的模型作为其距离度量。

参考文献

[1] Sam Roweis, Geoffrey Hinton, and Ruslan Salakhutdinov. Neighbourhood component analysis. *Adv. Neural Inf. Process. Syst.(NIPS)*, 17:513–520, 2004.

[2] 周志华. 机器学习. Qing hua da xue chu ban she, 2016.

[3] 黄宜华 and 苗凯翔. 深入理解大数据: 大数据处理与编程实践, 2014.