实验 1. 度量学习实验报告

MG1733098，周华平，zhp@smail.nju.edu.cn

2017 年 12 月 30 日

综述

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) 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。

在本实验中, 我们采用近邻成分分析 (Neighbourhood Component Analusis) 来实现距离度量学习，并使用 Python 实现算法并测试了其性能。

任务 1

度量函数学习目标

根据马氏距离的定义

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

其中 Q 称为“度量矩阵”，而度量学习则是对 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

在本实验中我们采用梯度下降法来求解目标函数。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。

由于 Python 的循环操作执行效率不高，因此算法优化的关键在于将尽量多的运算转化为矩阵操作，通过 numpy 提供的函数完成计算。由于在此之前没有用过 numpy，因此在编写代码前我参考了 github 上的开源实现 (RolT/NCA-python)，从中我进一步了解了 numpy 中比较重要的一种机制: broadcasting。虽然该机制在 numpy 的 tutorial 中也有介绍，但是在实际项目中的代码才使我意识到这种机制对于将运算向量化的重要性。因此在我自己实现 NCA 的过程中也大量使用了 broadcasting 机制。代码编写部分由我自己独立完成，因此与该开源项目的实现存在较大的不同，仅参考了其利用 broadcasting 的思想。

任务 2

与任务 1 中的数据集相比，任务 2 的数据集的维度和样本数都有明显增长，这导致内存无法容纳全梯度下降产生的中间结果，训练时间所需也明显上升。因此在任务 2 中我采用了随机抽样的批量梯度下降法，在每次迭代开始时都随机选取一定数量的样本进行训练并更新 A，这样能够使训练开销显著下降，同时取得还行的训练效果。

在任务 2 中，我将 A 初始化为单位矩阵，梯度下降的学习率设置为 0.005，迭代次数设置为 750，每次迭代随机选取的训练样本数为 500。最终结果如下

baseline+knn(k=1)	0.166880 ± 0.012082
myMetric+knn(k=1)	0.103632 ± 0.009503
baseline+knn(k=3)	0.206282 ± 0.013218
myMetric+knn(k=3)	0.124957 ± 0.010555
baseline+knn(k=5)	0.223248 ± 0.013466
myMetric+knn(k=5)	0.138932 ± 0.009505

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.