实验 1. 度量学习实验报告

MF1733034，李青坪，lqp19940918@163.com

2017 年 11 月 6 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。本实验的目的是掌握距离度量学习的基本思路方法并应用到真实场景中去。

任务 1

度量函数学习目标

根据马氏距离：

$$dist^2_{mah}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||^2_{\mathbf{M}}$$

其中的 $\mathbf{M}$ 称为 “度量矩阵”，而度量学习就是对 $\mathbf{M}$ 进行学习。为了保持距离非负且对称，$\mathbf{M}$ 必须是 (半) 正定对称矩阵。通过输入训练样本，使矩阵 $\mathbf{M}$ 能更准确地度量测试样本与其他样本之间的距离。本任务中，进行预测的时候便可以通过马氏距离计算样本之间的距离，使用近邻分类器对样本进行分类，矩阵 $\mathbf{M}$ 训练地越好，近邻分类器对样本标签的预测就越好。

优化算法

本次任务采用近邻成分分析 [1] (Neighbourhood Component Analysis，简称 NCA) 算法优化度量矩阵 $\mathbf{M}$。如果我们使用矩阵 $\mathbf{A}$ 作为一个变换矩阵，我们可以有效地学习度量矩阵 $\mathbf{M} = \mathbf{A}^T\mathbf{A}$, 使得

$$d(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathbf{M} (\mathbf{x} - \mathbf{y}) = (\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y})^T (\mathbf{A}\mathbf{x} - \mathbf{A}\mathbf{y})$$

近邻分类器在进行判别时通常使用多数投票法，邻域中的每个样本投 1 票，邻域外的样本投 0 票。不妨将其替换为概率投票法。对任意的样本 $\mathbf{x}_j$ ，它对 $\mathbf{x}_i$ 分类结果影响的概

率为

$$p_{ij} = \frac{exp(-||\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_j||^2)}{\sum_{k \neq c} exp(-||\mathbf{A}\mathbf{x}_i - \mathbf{A}\mathbf{x}_k||^2)} \quad , \quad p_{ii} = 0 \tag{1}$$

当 i=j 时，p_{ij} 最大。显然，$\mathbf{x}_j$ 对 $\mathbf{x}_i$ 的影响随着它们之间的距离的增大而减小，若以留一法 (LOO) 正确率的最大化为目标，则可计算 $\mathbf{x}_i$ 的留一法正确率，即它被除自身之外的所有样本正确分类的概率为

$$p_i = \sum_{j \in C_i} p_{ij} \tag{2}$$

C_i 表示与 $\mathbf{x}_i$ 属于相同类别的样本的下标集合。整个样本被正确分类的概率如下

$$f(\mathbf{A}) = \sum_i \sum_{i \in C_i} = \sum_i p_i \tag{3}$$

我们希望 f 尽可能的大。对 f 求偏导即产生了用于学习的梯度规则

$$\frac{\partial f}{\partial \mathbf{A}} = 2\mathbf{A} \sum_i \left(p_i \sum_k p_{ik} \mathbf{x}_{ik} \mathbf{x}_{ik}^T - \sum_{j \in C_i} p_{ij} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \right) \tag{4}$$

求得 f 的偏导之后，计算

$$\mathbf{A} = \mathbf{A} + \eta \frac{\partial f}{\partial \mathbf{A}} \tag{5}$$

其中，η 表示学习速率，根据经验设定值。在训练样本时，每次迭代都更新 $\mathbf{A}$, 使 $\mathbf{A}$ 收敛，计算 $\mathbf{M} = \mathbf{A}^T\mathbf{A}$，最后将 $\mathbf{M}$ 带入马氏距离求解的公式，计算样本之间的距离，使用 k 近邻分类器预测样本的标签。求 $\frac{\partial f}{\partial \mathbf{A}}$ 的时候，使用随机梯度下降算法，每次迭代，用样本中的一个例子来近似所有的样本 [2]。

实验结果

本次实验在 randnca.py 文件中实现了名为 randnca 的类，通过在 myDML.py 文件中引入并实例化该类，调用对象的 train 方法即可对训练样本进行训练。测试 η 设置为 0.03, 迭代 50000 次之后的错误率为：

$$k = 1\text{时} \quad test\_error = 0.000000$$
$$k = 3\text{时} \quad test\_error = 0.015000$$
$$k = 5\text{时} \quad test\_error = 0.020000$$

实验证明，在训练超过一定的次数后，通过训练得到的度量矩阵计算马氏距离来预测样本的标签的错误率比通过欧式距离来预测样本的标签要低。

任务 2

实验结果

通过调用任务 1 中实现的度量函数，对 30 个样本集进行训练与测试，每个样本集有 1820 个训练样本和 780 个测试样本，汇报 test error 的均值和标准差。η 设置为 0.03, 迭代

5000 次后的结果如下：

$$
\begin{aligned}
baseline + knn(k=1): &\quad 0.166880 \pm 0.012082 \\
myMetric + knn(k=1): &\quad 0.132906 \pm 0.014434 \\
baseline + knn(k=3): &\quad 0.206282 \pm 0.013218 \\
myMetric + knn(k=3): &\quad 0.160940 \pm 0.018085 \\
baseline + knn(k=5): &\quad 0.223248 \pm 0.013466 \\
myMetric + knn(k=5): &\quad 0.175897 \pm 0.017932
\end{aligned}
$$

η 设置为 0.02, 迭代 10000 次后的结果如下：

$$
\begin{aligned}
baseline + knn(k=1): &\quad 0.166880 \pm 0.012082 \\
myMetric + knn(k=1): &\quad 0.120513 \pm 0.013816 \\
baseline + knn(k=3): &\quad 0.206282 \pm 0.013218 \\
myMetric + knn(k=3): &\quad 0.145513 \pm 0.016387 \\
baseline + knn(k=5): &\quad 0.223248 \pm 0.013466 \\
myMetric + knn(k=5): &\quad 0.160085 \pm 0.015909
\end{aligned}
$$

η 设置为 0.01, 迭代 10000 次后的结果如下：

$$
\begin{aligned}
baseline + knn(k=1): &\quad 0.166880 \pm 0.012082 \\
myMetric + knn(k=1): &\quad 0.102094 \pm 0.011555 \\
baseline + knn(k=3): &\quad 0.206282 \pm 0.013218 \\
myMetric + knn(k=3): &\quad 0.125598 \pm 0.012827 \\
baseline + knn(k=5): &\quad 0.223248 \pm 0.013466 \\
myMetric + knn(k=5): &\quad 0.142949 \pm 0.011782
\end{aligned}
$$

η 设置为 0.006, 迭代 10000 次后的结果如下：

$$
\begin{aligned}
baseline + knn(k=1): &\quad 0.166880 \pm 0.012082 \\
myMetric + knn(k=1): &\quad 0.097607 \pm 0.009112 \\
baseline + knn(k=3): &\quad 0.206282 \pm 0.013218 \\
myMetric + knn(k=3): &\quad 0.122094 \pm 0.011422 \\
baseline + knn(k=5): &\quad 0.223248 \pm 0.013466 \\
myMetric + knn(k=5): &\quad 0.138291 \pm 0.010730
\end{aligned}
$$

实验结果表明，在经过一定的训练过后，使用 NCA 度量学习方法产生的距离度量方法比使用欧式距离作为 k 近邻分类器的距离度量方法预测样本标签的效果要好。

参考文献

[1] S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood component analysis. 2004.

[2] 周志华. *机器学习：= Machine learning.* 清华大学出版社, 2016.