

基于流形学习的多示例回归算法

詹德川 周志华

(南京大学软件新技术国家重点实验室 南京 210093)

摘 要 多示例学习是一种新型机器学习框架, 以往的研究主要集中在多示例分类上, 最近多示例回归受到了国际机器学习界的关注. 流形学习旨在获得非线性分布数据的内在结构, 可以用于非线性降维. 文中基于流形学习技术, 提出了用于解决多示例回归问题的 ManiMIL 算法. 该算法首先对训练包中的示例降维, 利用降维结果出现坍塌的特性对多示例包进行预测. 实验表明, ManiMIL 算法比现有的多示例算法例如 Citation- k NN 等有更好的性能.

关键词 机器学习; 多示例学习; 多示例回归; 流形学习

中图法分类号 TP18

A Manifold Learning-Based Multi-Instance Regression Algorithm

ZHAN De-Chuan ZHOU Zhi-Hua

(National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

Abstract Multi-instance learning is regarded as a new learning framework. Previous researches mainly focus on multi-instance classification. Recently, multi-instance regression attracts the attention of the machine learning community. Manifold learning attempts to obtain the intrinsic structure of non-linearly distributed data, which can be used in non-linear dimensionality reduction (NLDR). In this paper, a manifold learning-based multi-instance regression algorithm, ManiMIL, is proposed. ManiMIL performs NLDR on the instances in training bags, selects the most diverse dimension that NLDR brings and builds a classifier only on this dimension and then makes the prediction. Experimental results show that the performance of ManiMIL outperforms that of existing multi-instance algorithms such as Citation- k NN.

Keywords machine learning; multi-instance learning; multi-instance regression; manifold learning

1 引 言

1997 年, Dietterich 等人^[1]通过在药物活性预测(drug activity prediction)问题方面的研究工作, 提出了多示例学习(multi-instance learning)的概念. 在多示例学习中, 每个训练包(bag)包含多个示例, 示例没有概念标记, 但包有概念标记. 若包中至

少有一个示例是正例, 则该包被标记为正(positive); 若包中所有示例都是反例, 则该包被标记为反(negative). 学习系统通过对多个包所组成的训练集进行学习, 以尽可能正确地预测训练集之外的包的概念标记. 由于多示例学习具有广阔的应用前景和独特的性质且属于以往机器学习研究的一个盲区, 因此在国际机器学习界引起了极大的重视, 被认为是与监督学习、非监督学习和强化学习并列的第

四种示例学习框架^[2].

在多示例学习研究的早期,工作主要集中在概念标记为离散值的多示例分类上. 近些年, Ray 和 Page^[3]以及 Amar^[4]等人对类别标记为连续值的多示例回归进行了研究,并指出在很多实际问题中,多示例回归技术可能比多示例分类技术更为有用,例如在药物活性预测问题中,连续值输出可以表征分子绑定的强弱,这对药物分子的设计更有作用. 为了更准确地对新包进行预测,本文提出了一种基于流形学习(manifold-learning)的多示例回归算法 ManiMIL (Manifold-based Multi-Instance Learning). ManiMIL 使用流形学习算法 LLE^[5]对训练包中的示例进行非线性降维,然后在特征空间中利用降维算法在某些维度上产生的数据重叠现象(坍塌现象)对新包进行预测. 实验表明该算法可以取得比现有的多示例回归算法例如 Citation- k NN^[6]、BP-MIP 算法^[7]等更强的泛化能力.

本文首先简单介绍多示例学习和流形学习,然后提出 ManiMIL 算法,并给出实验结果和分析,最后对本文工作进行总结.

2 研究背景

2.1 多示例学习

20 世纪 90 年代中期, Dietterich 等人^[1]对药物活性预测问题进行了研究. 其目的是让学习系统通过对已知适于或不适于制药的分子进行分析,以尽可能正确地预测某种新的分子是否适合制造这种药物. 该问题的困难之处在于,每一个分子都有很多种可能的低能形状,只要该分子的某一种低能形状与期望的绑定区域(binding site)紧密耦合,该分子就适于制药. 生物化学家目前只知道哪些分子适于制药,并不知道具体的哪一种形状起到了决定性作用. 为了解决上述问题, Dietterich 等人将每一个分子作为一个包,而将分子的每一种低能形状作为包中的一个示例,由此提出了多示例学习. 在此基础上,他们将分子的低能形状通过属性-值对的形式表示出来,提出了 3 种轴平行矩形(Axis-Parallel Rectangles, APR)学习算法^[1]. 这些算法都是通过对属性值进行合取,在属性空间中寻找合适的轴平行矩形. Dietterich 等人发现, iterated-discrim APR 算法在药物活性预测问题上取得了最好的效果,而直接将 C4.5 决策树、BP 神经网络等常用的监督学习算法用于解决多示例学习问题效果很不理想. 由此可见,

如果不考虑多示例学习本身的特点,将难以很好地完成此类学习任务.

1998 年 Maron 和 Lozano-Pérez^[8]提出了多样性密度(diverse density)方法. 对于属性空间中的某一点,如果该点附近出现的正包数越多,而反包示例出现得越远,则该点的多样性密度越大. 他们使用梯度法来寻找多样性密度的最大点. 除了多样性密度方法以外,研究者们还提出了很多实用的多示例学习算法. 2000 年, Wang 和 Zucker^[6]通过结合惰性学习(lazy learning)和 Hausdorff 距离,成功地对 k -近邻(k -nearest neighbor)方法进行了扩展,提出了 Bayesian- k NN 和 Citation- k NN 两种方法用以处理多示例学习问题. 同年, Ruffo^[9]给出了 C4.5 决策树的多示例版本 Relic, 将其成功地应用于数据挖掘领域. 2001 年, Chevalyere 和 Zucker^[10]对决策树算法 ID3 以及规则学习算法 RIPPER 进行了扩展,得到了多示例决策树算法 ID3-MI 以及多示例规则学习算法 RIPPER-MI. Zhang 和 Goldman^[11]将 EM(Expectation Maximization)方法和多样性密度方法相结合,提出了 EM-DD 算法. 2002 年, Zhou 和 Zhang^[7]将神经网络引入了多示例分类问题,在包的层次上定义了全局误差函数,提出了一种多示例神经网络分类算法 BP-MIP, 此后,他们又通过引入属性选择技术,对 BP-MIP 进行了改进^[12]. 2003 年, Zhou 和 Zhang^[13]将集成学习技术引入多示例学习,获得了比单一多示例学习算法更强的泛化能力. 2003 年, Zhang 和 Zhou^[14]对 BP-MIP 进行扩展,提出了基于神经网络的多示例回归算法 BP-MIR. 目前,多示例学习已经在股票选择^[8]、图像分析^[8]、自然场景分类^[15]、Web 挖掘^[16]等领域得到了成功的应用.

2.2 流形学习

机器学习算法应用于实际问题时经常遇到高维数据,形成所谓的维度灾难^[17]. 通常需要对数据进行降维处理,相对于样本数据所处的空间样本空间,降维以后的样本所处的空间称为特征空间. 对数据进行降维就是建立一个从高维样本空间向低维特征空间的映射. 目前成熟的降维方法有主成分分析(PCA)、独立成分分析(ICA)及多维缩放(MDS)等. 主成分分析寻求数据在伸展的最大的几个方向上的投影,忽略那些有可能是由噪音引起的数据波动. 独立成分分析和主成分分析相似,它寻求的是数据相互独立的几个方向上的投影. 而使用多维缩放进行的降维映射可以用来保持数据两两之间的距离. 这

些线性降维方法便于实现,易于理解,并且对于测试样本可以轻易地对其进行降维处理,投影至特征空间,所以得到了广泛的应用,但是因为它们属于线性降维方法,存在固有的缺陷——大部分真实世界的数据是非线性的。

2000年, Tenenbaum 等人^[18]提出流形学习算法 Isomap, 于是机器学习研究者开始更加关注非线性数据内在本质的分布. 对于样本空间中的任意两点, 它们之间的距离用它们的测地线距离度量, 而该距离度量的近似值可以使用最短路径算法通过局部的邻域中欧氏距离来重构获得. Isomap 利用了这一信息, 它首先找出每个样本的邻域, 保留邻域内样本之间的距离, 邻域外样本到该样本不连通, 然后使用 Dijkstra 算法重构任意两个样本之间的距离矩阵, 最后对样本间的两两距离作为标准 MDS 的输入, 从而得到数据的低维映射。

同年, Roweis 和 Lawrence^[5]提出了 LLE. LLE 算法和 Isomap 算法都试图寻求数据本质的分布, 并将数据投影至一个低维空间, 同时保持数据之间局部的相似度. 对于 LLE 算法, 其思想在于: 对于一个分布于流形表面的数据集, 在样本空间与特征空间局部邻域间的点的关系应该不变. 即在样本空间中每个示例可以用它的近邻线性表示, 在低维空间中保持每个邻域中的权值不变, 重构原数据点, 使重构误差最小. 具体来说是: 假设样本 x_i 是 N 维实值向量 ($i=1, 2, \dots, m$), x_j ($j=j_1, j_2, \dots, j_k$) 是 x_i 的 k 个近邻, LLE 首先将 x_i 的邻域信息, 通过最小化下式

$$\epsilon(\mathbf{W}) = \sum_i |x_i - \sum_j W_{ij} x_j|^2 \quad (1)$$

保留在权值矩阵 \mathbf{W} 中, 然后使用已经获得的权值矩阵 \mathbf{W} , 通过最小化整体重构误差

$$\Phi(\mathbf{W}) = \sum_i |y_i - \sum_j W_{ij} y_j|^2 \quad (2)$$

来获取 x_i 对应的投影 y_i ($i=1, 2, \dots, m$).

虽然流形学习算法 Isomap 和 LLE 可以在训练示例上建立从高维样本空间向低维特征空间的非线性映射模型, 但是由于这些投影涉及对距离矩阵或者权值矩阵的特征分解, 并且不能像 PCA 那样生成变换矩阵, 因而对于测试示例, 它们不能被投影至特征空间中, 所以流形学习算法缺乏对测试示例的处理能力. Vlachos 等人^[19]于 2002 年提出使用径向基网络 (RBF) 学习流形学习算法 Isomap 建立的映射模型, 然后将测试示例提交给训练好的 RBF, 其输出即为特征空间中的坐标, 这一过程利用了 RBF 的

学习能力, 间接地从 Isomap 建立的模型中获得了可以处理测试示例的映射函数. 然而该方法过于依赖 RBF, 并且加上了一个输入为样本空间坐标、输出为特征空间坐标的学习过程, 使得整个学习过程过于繁杂, 效率低下. 为了能够将测试包中的示例有效地投影至特征空间, 本文使用了一种基于 LLE 思想的映射方法 LLEP (LLE based Projection), 能够很好地解决流形学习中测试样本的映射问题。

另外, 由于流形学习算法关注于局部空间中数据的相似度, Isomap 和 LLE 的第一步都是寻找样本之间的近邻. 为简便起见, 如果样本 x_1 是 x_2 的近邻并且 x_2 是 x_3 的近邻, 就定义样本 x_1 和 x_3 是连通的. 然而当数据分布呈集簇状时, 就会出现样本 x_i 和样本 x_j 不连通的现象, 此时所有和 x_i 相连通的样本形成一个闭包, 闭包中的所有样本都不与 x_j 相连通. 此时 Isomap 算法就无法将样本映射至一个连贯的坐标系中, Geng 等人^[20]于 2005 年对 Isomap 进行改进提出了 S-Isomap, 他们将类别信息引入距离度量, 用来保证不同类别之间的样本可以互相连通, 并将该方法用于分类, 获得良好的泛化能力。

当出现不连通现象时, 和 Isomap 相比, LLE 依然可以将样本进行映射, 但是可能在特征空间的某些维度上出现所谓的坍塌 (collapse) 现象. 所谓坍塌即不同的样本, 由于相似度很高, 在降维处理后的特征空间中, 前几个维度上的坐标相同或者极为近似, 就如同投影至同一个点上, 当然流形学习算法 LLE 也会出现这种现象, 在下文中不加说明时是将样本投影至发生坍塌的前几个维度上. 大部分情况下, 坍塌应该避免, 例如在进行数据可视化时, 人们期望能够在一张图上显示各种数据, 以便从整体上获取对这些数据的认识, 这时数据聚在一点没有区分度或者不连通是不行的, 所以要加以避免. 坍塌现象和邻域选取有关, 一般来说, 由于同类示例间的相似度较大, 当同类示例数目远大于近邻选取参数时, 选出的近邻可能全是同一类的示例, 这样就无法和其它类相连, 这时对于 LLE 就可能会出现坍塌. 一个极端的例子如: 两类数据是由 3 维空间的两个高斯分布形成, 其中第一类的高斯分布的中心是 $\mathbf{0}(0, 0, 0)$, 另一个的中心是 $\mathbf{5}(5, 5, 5)$, 方差都为 1. 那么 LLE 将会把两类的样本分别投影至 3 维空间的两条不相交的直线上 ($x=0, y=-1.5$) 和 ($z=0, y=0$), 这种情况显然发生了坍塌. 当然对于真实数据不会有这种极端的现象, 但是仍然可能发生在某些维度上, 例如实验部分中提到的多示例数据很容易发生这种

现象. 正是由于多示例学习的特殊性, 坍塌现象会经常发生, 而且实验表明 LLE 用于多示例学习时和目标概念相关的信息往往富集在发生坍塌的某些维度中. 所以本文提出的 ManiMIL 算法就要利用这种性质, 以便在多示例数据上获得更好的学习效果.

3 ManiMIL 算法

一些多示例学习算法暗示不同正例之间应该有着较高的相似程度, 而反例和正例之间的相似程度小于正例之间的相似程度. 例如在 Citation- k NN 中, 包和包之间的距离使用最小 Hausdorff 距离定义, 如果测试正包中存在的正例到反包中样例的距离反而小于到其它正包中的正例的距离, 则会得到该正包是反包这样错误的预测. 同时, 由于正包和反包中都包含反例, 所以一般反例样本数目较多, 当正例数目也大于近邻选取参数时(通常情况下这是满足的, 除非正例数目特别少, 而这时一般的多示例学习算法也难以进行有效的学习), 如第 2 节所述, 这就可能使得 LLE 出现坍塌现象. 在出现了坍塌现象的特征空间内, 相似的示例投影后的坐标的前几维会聚成一点. 不妨设 $\mathbf{z}_i (i = i_{11}, i_{12}, i_{13}, \dots, i_{ab})$ 为训练包中的所有正例, 其中 i_{ab} 表示第 a 个正包中的第 b 个正例, $\mathbf{z}_j (j = j_{11}, j_{12}, \dots, j_{ef})$, j_{ef} 表示第 e 个包(正包或者反包)中的第 f 个反例. 由于出现了坍塌现象, 加之 \mathbf{z}_i 自身的相似性, 正例可能被投影至特征空间的相互靠近的几点 $\mathbf{z}'_i (i = 1, 2, \dots, c, c \ll m \times n)$, 反例也同样地被投影至特征空间相互靠近的几个点 \mathbf{z}'_j 上, 并且和 \mathbf{z}'_i 相距较大. 值得注意的是, 虽然特征空间的前几个维度会发生坍塌, 但是并不是在所有的发生坍塌的维度上正例都是可以聚集的. 在某些维度上, 有一些示例虽然发生了坍塌, 但是仍然很难区分正例和反例, 所以选取一个合理的维度也是很重要的, 而在本文的实验部分也将揭示选取合理维度的重要性.

如果能够将测试包中的示例一一映射至特征空间, 并且只要存在一个示例在特征空间的坐标等于或贴近某个 \mathbf{z}'_i , 并且远离所有的 \mathbf{z}'_j , 就说明该示例极有可能为正例, 而该包也极有可能为正包, 否则为反包. 为了能将测试示例映射至特征空间, 本文使用了 LLEP. LLE 算法本质上是将在样本空间中样本的邻域信息表示成权值矩阵并在特征空间中保留下来, LLEP 算法使用了与此相似的思想来处理未知样本的映射. 在 LLEP 中, 对于测试样本 \mathbf{x} , 首先在训练

集上找到它的 k 个近邻, 不妨设为 $\mathbf{x}_j (j = j_1, j_2, \dots, j_k)$, 通过最小化下式来计算权值矩阵 \mathbf{W} , 保留邻域信息:

$$\epsilon'(\mathbf{W}) = \left| \mathbf{x} - \sum_j \mathbf{W}_{ij} \mathbf{x}_j \right|^2 \quad (3)$$

由于 \mathbf{x}_j 对应的 $\mathbf{y}_j (j = j_1, j_2, \dots, j_k)$ 是训练示例在特征空间的坐标, 可以由 LLE 算法获得, 所以可以使用线性插值的方法来获取测试样本 \mathbf{x} 的投影坐标 \mathbf{y} , 即

$$\mathbf{y} = \sum_j \mathbf{W}_{ij} \mathbf{y}_j \quad (4)$$

这样做可以将测试样本映射至特征空间, 却未必能够保证训练示例和测试示例整体重构误差 $\Phi(\mathbf{W})$ 最小化, 但是如同大多数学习算法在测试时都不改变训练生成的模型一样, 它不会像重新计算 LLE 那样改变已经生成的模型, 这体现了对训练示例的置信高, 并且也更加节省时间. 但是这样做至少在保持现有模型的前提下可以使测试示例不增加重构误差. 所以, LLEP 算法寻求的是在不改变 LLE 建立好的模型的基础上, 使用“保持数据之间局部的相似性”这一思想对测试样本的最优映射.

使用 LLEP 就可以处理测试数据, 进而使用 ManiMIL 算法对测试包进行处理. 首先, 如前所述应该找出包中样例之间坍塌得最为合理的一个维度 h . 实际上这个过程可以看作一个简化的特征选择, 因为 LLE 算法使得几乎所有和分类相关的信息都集中在发生坍塌的某一个维度上, 所以只要找出特征空间中的这个最具有区分性的一个维度/特征就可以了, 这样不但能够获取好的性能还可以节省计算开销. 然后再在该维度上训练一个线性分类器, 因为只选取一维, 相当于映射至一维直线上, 而训练一个线性分类器就是寻找一个阈值; 又因为正例相互靠近的投影点 \mathbf{z}'_i 和反例的投影点相距较远, 所以也应该存在这样的阈值: 若包中存在大于或者小于该值的样例则判为正包, 否则为反包. 对于投影后的测试样本, 使用该阈值进行分类, 最终可以对包的属性进行快速判别. 其中, 关于维度 h 的选取, 因为坍塌通常发生在 LLE 降维后的前几个维度上, 可以使用交叉验证的方法从前 l 个维度中寻找, 目标是在维度 h 上交叉验证对包的精度预测值最高. 而寻找阈值的目的在于: 对于测试样例在特征空间中的坐标 \mathbf{z}'_i 大于或小于某个阈值就判正. ManiMIL 算法描述如下. 其中第 2~5 步是 LLE 算法, 第 6~9 步是 LLEP 的步骤.

ManiMIL(trainbags, testbags, label, k , l , t)

输入:

trainbags: 训练包

testbags: 测试包

label: 训练包的标记

k : LLE 和 LLEP 用来寻找邻域的参数

l : 从 LLE/LLEP 中返回前 l 个维度坐标

t : 使用 t -fold CV 寻找最合适的维度

步骤:

1. 将训练包中的示例组成示例集合 trainins, 测试包中的示例组成集合 testins
 2. For trainins 中的每个示例 x_i
 3. 找到它的 k 个最近邻 $x_j (j=1, 2, \dots, k)$
 4. End For
 5. 最小化式(1), 获取权值矩阵 W , 再最小化式(2), 得到 trainins 中的示例投影 y_i
($i=1, 2, \dots, m$, m 是 trainins 中示例的个数, y_i 的维数为 l).
 6. For testins 中的每个示例 x_c
 7. 找到它在 trainins 中的 k 个近邻 $x_j (j=1, 2, \dots, k)$
 8. End For
 9. 通过最小化式(3), 获取 W , 再使用式(4), 得到 testins 的示例投影 y_c
($c=1, 2, \dots, n$, n 是 testins 中示例的个数, y_c 的维数为 l).
 10. 将 trainins 分成 t 份, 1 份作为验证, $t-1$ 份用作训练, 对每个维度上的投影使用 t -fold CV 进行精度的评估, 进而在 LLE/LLEP 返回的 l 维中寻求最合适的维度 h
 11. 将 y_i 和 y_c 投影至维度 h 上得 y_{ih} 和 y_{ch}
 12. 在 y_{ih} 训练一个线性分类器, 其阈值为 th
 13. For $c=1, 2, \dots, n$
 14. If $y_{ch} > th$ (or $y_{ch} < th$)
 15. 包含示例 x_c 的测试包为正包;
 16. End If
 17. End For
- 输出:
- 测试包的标记

4 实验结果

2001 年, Amar 等人^[4] 根据计算分子之间的绑定耦合度的经验公式, 设计出了一些具有物理含义的基准数据集(人造分子). 数据集的命名规则如下: LJ- $r.f.s$, 其中 r 表示相关的属性数目, f 表示示例的属性个数, s 表示属性相关系数的设置, 如果 $s=1$, 则表示存在 0 和 1 两种相关系数. 使用绑定耦合度的经验公式, 可以计算出目标分子的活性, 是一个 0~1 之间的比值. Amar 等人指定, 对于活性大于 0.5 的分子是有效的. 这样, 在 LJ 数据集中, 每个包

的活性由该包中活性最大的分子所决定, 如果包中存在活性大于 0.5 的分子, 则该包为正包, 否则为反包.

本文使用 ManiMIL 算法在 LJ-80.166.1, LJ-80.166.1-s, LJ-150.283.2, LJ-150.283.4, LJ-150.283.10, LJ-150.283.15, LJ-40.283.4, LJ-80.283.4 这 8 个数据集上进行了实验(为了描述方便, 下文数据集分别编号为 1~8). 实验均采用了 10 次 10 倍交叉验证的方法对学习器精度进行估计, 具体说就是将包按类别平均分成 10 个子集, 取其中 1 个子集用于测试, 9 个子集用于训练, 10 个子集轮换作为测试集, 这称为一次 10 倍交叉验证. 这一过程重复执行 10 次, 每次对 10 个子集进行随机划分, 最终的平均结果即为 10 次 10 倍交叉验证的结果. 实验参数设置如下: $l=5$, k 从 5 取至 10, h 也通过交叉验证来确定. 实验中将 ManiMIL 和 Citation- k NN 进行了比较, 其中 Citation- k NN 中的参数设置和 Amar 等人^[4] 文中的设置相同 $ref=3$, $cite=5$, 因为此时 Citation- k NN 在 LJ 数据集上的表现最为优秀. 另外, 2003 年 Zhang 和 Zhou^[16] 曾设计了用于多示例回归的神经网络算法 BP-MIR, 并在数据集 1 和 2 上进行了实验, 对这两个结果本文也进行了比较. 表 1 中是实验结果.

从表 1 可以看出, 在 8 个数据集上, ManiMIL 算法的最优值均优于 Citation- k NN 算法, 尤其在 2, 3, 4, 5, 6 几个数据集上, 无论 k 取任何值, ManiMIL 的算法性能明显优于 Citation- k NN. 在 7, 8 数据集上, k 取 5~9 时 ManiMIL 表现并非最佳, 但是和 Citation- k NN 仍然高度可比. 只有在数据集 1 上, 并且 k 取 7~10 时, ManiMIL 算法性能逊于 Citation- k NN. 在 1, 2 两个数据集上, ManiMIL 的结果大都优于 BP-MIR, 只有在数据集 1 上, $k=10$ 时, ManiMIL 略差于 BP-MIR. 而从计算复杂度上考虑, ManiMIL 测试时因为也需要进行近邻计算, 和 k NN 的耗时相近, 但是 Citation- k NN 涉及 $cite$ 和 ref 两个方面, 其中获取 $cite$ 的耗时较多, 所以 ManiMIL 在时间代价上优于 Citation- k NN.

表 1 ManiMIL 和 Citation- k NN 的算法性能比较

数据集	ManiMIL						Citation- k NN	BP-MIR
	$k=5$	$k=6$	$k=7$	$k=8$	$k=9$	$k=10$		
1	0.921±0.003	0.916±0.011	0.884±0.005	0.857±0.004	0.831±0.011	0.806±0.012	0.915±0.011	0.815±N/A
2	1.000±0.000	1.000±0.000	0.999±0.000	0.999±0.000	0.999±0.000	0.999±0.003	0.992±0.009	0.815±N/A
3	0.878±0.022	0.874±0.001	0.880±0.004	0.900±0.000	0.916±0.000	0.924±0.005	0.644±0.014	N/A
4	0.872±0.016	0.870±0.000	0.882±0.001	0.898±0.000	0.912±0.000	0.925±0.002	0.646±0.011	N/A
5	0.878±0.007	0.873±0.001	0.888±0.001	0.898±0.000	0.918±0.001	0.925±0.001	0.649±0.008	N/A
6	0.876±0.014	0.876±0.001	0.886±0.000	0.903±0.000	0.910±0.000	0.923±0.001	0.647±0.010	N/A
7	0.909±0.000	0.910±0.000	0.911±0.000	0.917±0.000	0.914±0.001	0.922±0.001	0.921±0.003	N/A
8	0.975±0.003	0.974±0.001	0.980±0.001	0.984±0.000	0.989±0.001	0.993±0.004	0.991±0.015	N/A

本文认为,ManiMIL 算法之所以表现良好,是因为在 LLE 的作用下,数据发生了合理的坍塌.例如,图 1(a)中是从数据集 4 中取样出的已经按标记排序好的 54 个包,对于 LJ 数据集其标记有着明确的物理含义——分子活性,用作训练(交叉验证中的一次),共有 714 个示例,横坐标就是这些示例索引,纵坐标表示的是与这些样例所属的包相对应的标记(分子活性),因为示例的活性本身未知,所以用包的活性代替.例如前 4 个示例来自于第 1 个包,它是一个标记活性为 1.00 的包,所以前四个示例都标记为 1.00;如图 1(a)中竖直的实线所示,可以看出前 225 个样例因为包含它们的包的标记大于 0.5,所以它们来自于正包,其它的来自于反包.图 1(b)的纵坐标是将这些训练示例进行 LLE 投影后第二维的坐标值,横坐标也是相应的示例索引.明显这些坐标发生了坍塌,很多示例投影至 0 和 1 之间,这些示例有些是来自正包,有些来自于反包;正包中另有许多的样例被投影至约 -0.9 和 -2.1 处,如图中右边框上的黑色点所示,其示例的索引均位于竖直实线左方,也就是说它们来自于前 225 个示例;反包中的示例的投影大都大于 0,如图 1(b)右边框上的加重部分

标出的区域,但是从第 226 个到第 380 个样本,和正例比较相似,正如图中纵坐标较大的右边框上黑点和加重的部分有相交部分所示,这是因为包含它们的包的活性比较接近前 225 个样本所属的包.图 1(b)表示在数据集 4 上第 2 维坐标是比较合理的,此时如果学习到的阈值为 -1.5 ,就可以快速准确地对包进行标记(含有小于 -1.5 的示例的包是正包,如图中虚线所示),而相对于 Citation- k NN 等其他多示例算法对阈值的学习是相对简单的.

但是并非所有发生坍塌的维度都是合理的,如图 2(a)所示,在第 3 维上数据虽然已经坍塌,但是第 3 维正例所坍塌到的几个坐标和某些反例映射后的坐标相近,图中正例所映射得到的右边框上黑点被淹没在加重标出的反例映射得到的坐标中,所以其效果不如第 2 维坐标效果好.图 2(b)表示的是降维后的第 4 维坐标,可以看出并没有发生坍塌现象.图 2 说明使用交叉验证对合理维度的选取是必要的.

图 1(b)还说明了一个问题,因为 LLE 将许多样本投影至约 0.5 处,其中样本可能来自正包,也可能来自反包,这说明 Citation- k NN 所依赖的最小 Hausdorff 距离可能引起误导,因为正包中的反例

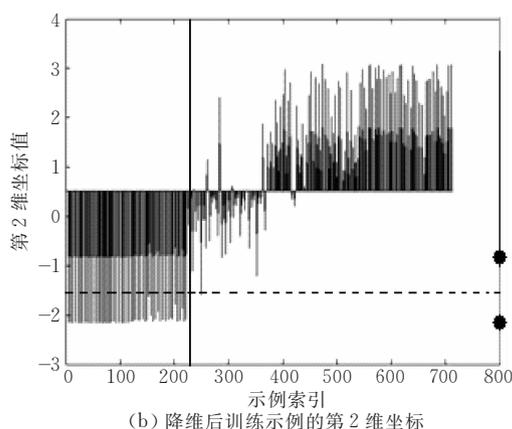
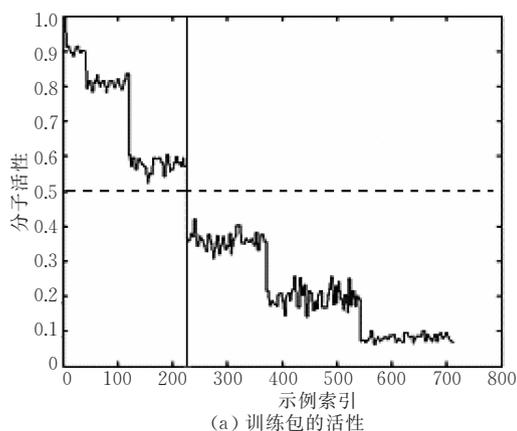


图 1 LLE 的坍塌现象(数据集 4,第 2 维)

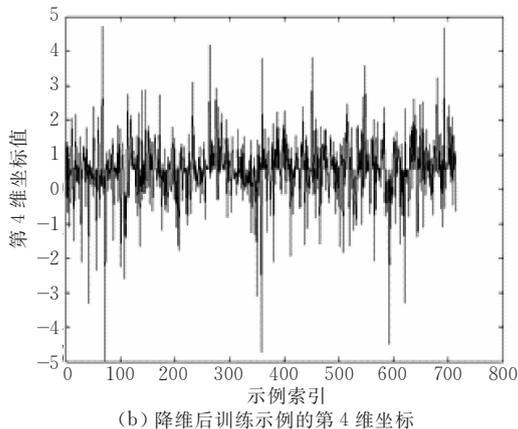
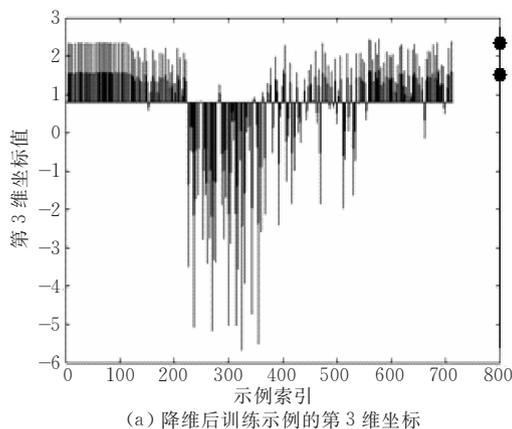


图 2 LLE 的坍塌现象(数据集 4,第 3 和 4 维)

和反包中的示例也可能有较高的相似度,从而很有可能将测试的正包误作为反包的近邻,导致错误的预测;所以在 LJ-150.283 系列数据集上,Citation- k NN 表现欠佳。

5 结束语

本文利用了流形学习算法 LLE 作用于多示例数据上容易在某些维度产生坍塌的现象,提出了一种基于流形学习的多示例学习算法 ManiMIL. ManiMIL 算法首先使用 LLE 将训练包中的示例投影至特征空间,然后使用 LLEP 将测试包中的示例投影至相同的空间;在完成投影后,ManiMIL 利用 LLE 的坍塌现象来进行预测. 在 LJ 系列数据集上的实验结果表明,ManiMIL 算法拥有良好的性能;并且由于 LLE 富集了目标信息使得只需要使用交叉验证选择一个维度,就能够获得很好的泛化性能,所以 ManiMIL 算法的时间开销也是远远小于 Citation- k NN.

值得注意的是,在数据存在较多的噪音和无关属性时,正例之间的相似度可能会有较大差异,而在正包数量较少时,正例数量有可能不足,这样,LLE 可能难以发生 ManiMIL 所需的坍塌. 为了解决这些问题,不仅需要多示例学习问题进行更深入的研究,还需要设计出更强有力的流形学习算法.

另外,本文工作一方面显示出 LLE 坍塌这一通常被认为不好的现象在多示例学习上反倒可能起到积极的作用,另一方面也可以看作一种特殊的特征选择思路. 由于在本文实验中 ManiMIL 算法只需要使用一个具有区分性的维度就可以很好地完成多示例回归任务,所以 ManiMIL 并没有在 LLE 特征空间再进行特征选择. 是否选取多个维度的结果会更好也是一个值得研究的问题.

参 考 文 献

- 1 Dietterich T. G., Lathrop R. H., Lozano-Pérez T.. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 1997, 89(1/2): 31~71
- 2 Maron O.. Learning from ambiguity [Ph. D. dissertation]. Department of Electrical Engineering and Computer Science, MIT, 1998
- 3 Ray S., Page D.. Multiple instance regression. In: Brodley C. E., Danyluk A. P. eds. *Proceedings of the 18th International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, 425~432
- 4 Amar R. A., Dooly D. R., Goldman S. A., Zhang Q.. Multi-

- 5 ple-Instance learning of real-valued data. In: Brodley C. E., Danyluk A. P. eds. *Proceedings of the 18th International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 2001, 3~10
- 6 Roweis S. T., Lawrence K. S.. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 2000, 290(5500): 2323~2326
- 7 Wang J., Zucker J.-D.. Solving the multiple-instance problem: A lazy learning approach. In: Langley P. ed. *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, MK, 1998, 341~349
- 8 Zhou Z.-H., Zhang M.-L.. Neural networks for multi-instance learning. AI Lab, Computer Science & Technology Department, Nanjing University, Nanjing, China; Technical Report, 2002
- 9 Maron O., Lozano-Pérez T.. A framework for multiple-instance learning. In: Jordan M. I., Kearns M. J., Solla S. A. eds. *Advances in Neural Information Processing Systems 10*. Cambridge: MIT Press, 1998, 570~576
- 10 Ruffo G.. Learning single and multiple instance decision trees for computer security applications [Ph. D. dissertation]. Torino; Department of Computer Science, University of Turin, 2000
- 11 Chevalere Y., Zucker J.-D.. Solving multiple-instance and multiple-part learning problems with decision trees and decision rules. Application to the mutagenesis problem. In: Stroulia E., Matwin S. eds. *Proceedings of the 14th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*. Berlin: Springer-Verlag, 2001, 204~214
- 12 Zhang Q., Goldman S. A.. EM-DD: An improved multiple-instance learning technique. In: Dietterich T. G., Becker S., Ghahramani Z. eds. *Advances in Neural Information Processing System 14*. Cambridge: MIT Press, 2002, 1073~1080
- 13 Zhang M.-L., Zhou Z.-H.. Improve multi-instance neural networks through feature selection. *Neural Processing Letters*, 2004, 19(1): 1~10
- 14 Zhou Z.-H., Zhang M.-L.. Ensembles of Multi-Instance Learners. In: Lavrac N., Gamberger D., Todorovski L., Blockeel L. eds. *Proceedings of the 14th European Conference on Machine Learning*. Berlin: Springer-Verlag, 2003, 492~502
- 15 Zhang M.-L., Zhou Z. H.. A multi-instance regression algorithm based on neural network. *Journal of Software*, 2003, 14(7): 1238~1241(in Chinese)
(张敏灵,周志华. 基于神经网络的多示例回归算法. 软件学报, 2003, 14(7): 1238~1241)
- 16 Maron O., Ratan A. L.. Multiple-Instance learning for natural scene classification. In: Koller D., Fratkin R. eds. *Proceedings of the 15th International Conference on Machine Learning*. San Francisco: MK, 1998, 341~349
- 17 Zhou Z.-H., Jiang K., Li M.. Multi-instance learning based web mining. *Applied Intelligence*, 2005, 22(2): 135~147
- 18 Duda R. O., Hart P. E., Stork D. G.. *Pattern Classification*. New York: John Wiley & Sons, 2001

- 18 Tenenbaum J. B. , de Silva V. , Langford J. C. . A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000, 90(5500): 2319~2323
- 19 Vlachos M. , Domeniconi C. , Gunopulos D. , Kollios G. , Koudas N. . Non-linear dimensionality reduction techniques for classification and visualization. In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton. NY: ACM Press, 2002, 645~651
- 20 Geng X. , Zhan D. -C. , Zhou Z. -H. . Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, 2005, 35(6): 1098~1107



ZHAN De-Chuan, born in 1982, M.S. candidate. His research interests include manifold learning and machine learning.

ZHOU Zhi-Hua, born in 1973, Ph.D., professor, Ph.D. supervisor. His main research interests include machine learning, data mining, pattern recognition, information retrieval, neural computing and evolutionary computing.

Background

At present, roughly speaking, there are three frameworks for learning from examples. That is, supervised learning, unsupervised learning and reinforcement learning. Multi-instance learning is regarded as a new learning framework. Previous researches mainly focus on multi-instance classification. Recently, multi-instance regression attracts

the attention of the machine learning community. Manifold learning attempts to obtain the intrinsic structure of non-linearly distributed data, which can be used in non-linear dimensionality reduction(NLDR). In this paper, a manifold learning-based multi-instance regression algorithm, ManiMIL, is proposed.