

基于多源的跨领域数据分类快速新算法

顾鑫^{1,2} 王士同¹ 许敏^{1,3}

摘要 研究跨领域学习与分类是为了将对多源域的有监督学习结果有效地迁移至目标域,实现对目标域的无标记分类. 当前的跨领域学习一般侧重于对单一源域到目标域的学习,且样本规模普遍较小,此类方法领域自适应性较差,面对大样本数据更显得无能为力,从而直接影响跨域学习的分类精度与效率. 为了尽可能多地利用相关领域的有用数据,本文提出了一种多源跨领域分类算法 (Multiple sources cross-domain classification, MSCC), 该算法依据被众多实验证明有效的“罗杰斯特回归模型”与“一致性方法”构建多个源域分类器并综合指导目标域的数据分类. 为了充分高效利用大样本的源域数据,满足大样本的快速运算,在 MSCC 的基础上,本文结合最新的 CDdual (Dual coordinate descent method) 算法,提出了算法 MSCC 的快速算法 MSCC-CDdual, 并进行了相关的理论分析. 人工数据集、文本数据集与图像数据集的实验运行结果表明,该算法对于大样本数据集有着较高的分类精度、快速的运行速度和较高的领域自适应性. 本文的主要贡献体现在三个方面: 1) 针对多源跨领域分类提出了一种新的“一致性方法”,该方法有利于将 MSCC 算法发展为 MSCC-CDdual 快速算法; 2) 提出了 MSCC-CDdual 快速算法,该算法既适用于样本较少的数据集又适用于大样本数据集; 3) MSCC-CDdual 算法在高维数据集上相比其他算法展现了其独特的优势.

关键词 跨领域, 多源, 罗杰斯特回归, 后验概率, 分类

引用格式 顾鑫, 王士同, 许敏. 基于多源的跨领域数据分类快速新算法. 自动化学报, 2014, 40(3): 531–547

DOI 10.3724/SP.J.1004.2014.00531

A New Cross-multidomain Classification Algorithm and Its Fast Version for Large Datasets

GU Xin^{1,2} WANG Shi-Tong¹ XU Min^{1,3}

Abstract Cross-domain learning and classification involved in this paper attempts to effectively transfer the classification results obtained from supervised multisource domains to an unsupervised target domain. Generally speaking, although current cross-domain learning methods have obtained great successes for cross-single-domain learning problems, they will encounter overwhelming troubles in the sense of classification accuracy and running speed when carrying out them on large cross-multisource datasets. In this paper, based on the logistic regression model and the proposed consensus measure, a multi-source cross-domain classification (MSCC) algorithm is proposed to realize effective cross-domain classification for the target domain. In order to enable the MSCC to work well for large datasets, based on the algorithm CDdual (Dual coordinate descent method) as the recent advance about large-scale logistic regression, an MSCC's fast version MSCC-CDdual for large datasets is derived and theoretically analysed. The experimental results on artificial data, text data and image data indicate that the proposed algorithm MSCC-CDdual has a fast speed, high classification accuracy and good domain adaption for large cross-multisource datasets. The contributions of the work here contain three aspects: 1) A novel consensus measure is proposed, which is suitable for boosting multi-classifiers and convenient for us to develop MSCC's fast version for large datasets; 2) The proposed algorithm MSCC-CDdual is demonstrated to be suitable for cross-multisource learning for both small and large datasets; 3) MSCC-CDdual exhibits its additional advantage, i.e., the applicability for high dimensional datasets from another “large” perspective.

Key words Cross-domain, multi-source, logistic regression, posterior probability, classification

Citation Gu Xin, Wang Shi-Tong, Xu Min. A new cross-multidomain classification algorithm and its fast version for large datasets. *Acta Automatica Sinica*, 2014, 40(3): 531–547

收稿日期 2012-06-25 录用日期 2013-02-04
Manuscript received June 25, 2012; accepted February 4, 2013
国家自然科学基金 (60903100, 60975027) 资助
Supported by National Natural Science Foundation of China (60903100, 60975027)
本文责任编辑 张学工
Recommended by Associate Editor ZHANG Xue-Gong
1. 江南大学数字媒体学院 无锡 214122 2. 江苏北方湖光光电有限公司 无锡 214035 3. 无锡职业技术学院 无锡 214000
1. School of Digital Media, Jiangnan University, Wuxi 214122

近年来,随着计算机以及网络技术的飞速发展,在社会生活的各个领域出现了大量的数据,如何在这些数据中提取有用的信息,几乎成为所有领域的共同需求.而获得有标记的数据是费时费力的,所以如何利用大量存在的未标记数据成为备受关注的问

2. Jiangsu North Huguang Opto-Electronics Co.Ltd., Wuxi 214035 3. Wuxi Institute of Technology, Wuxi 214000

题. 传统的知识学习一般假设训练集和测试集满足相同的数据分布. 但实际情况下数据并不能总是严格地服从这个假设, 往往测试集是一些经过快速演变的无标签数据, 其数据特性或语义与训练集相似但数据分布不同, 这种情况在现实世界里经常发生, 如互联网中的网页分类, 当考虑怎样将大学网站的主页与该校的其他网页分开时, 如果每次都靠人工收集网页数据并进行标签标注, 其付出的人工成本较高. 这时可以将已含有标签属性的其他大学的网页数据作为训练集, 但要注意的是每所大学的主页风格不尽相同, 如关于阅读部分有的习惯用语为“Required Reading List”, 有的为“Textbooks”, 还有的为“Reference”, 其主页数据满足不同的数据分布. 将已获得标签属性的多个大学网页数据作为多个源域数据, 这样就可以通过对多源域的学习完成对不含标签属性的目标域大学的网页分类, 换句话说通过对多源域的学习能够获取主页信息的一些共性特征并将其迁移应用在目标域主页分类. 这就是一种真实的多源跨领域学习应用, 当只是考虑将一所大学的含标签属性的数据做为源域, 那就是传统的跨领域学习^[1-3] 或者称为单源域数据到目标域的迁移学习^[4-6]. 而本文研究方向为多源的跨领域学习, 其源域与目标域数据分布不同但近似, 源域与目标域的分布差异可参考文献 [7-8] 进行量化.

一种似乎可行的方法是人为地将多个源域合并为一个源域后, 用现有的单源跨领域算法进行学习, 但是我们必须指出此方法将丢失不同源域数据分布间的差异性这一重要信息. 所以有必要提出一种新的针对多源的跨领域学习算法. 在文献 [9] 中, Zhuang 等提出了一种基于罗杰斯特回归函数的一致性模型来解决多源跨领域学习. 他们从理论和实验上证明了通过最大化源域分类器的一致性能提高目标域的分类精度. 这给我们带来了启示, 本文应用该模型并在此基础上提出了新的一致性方法. 在文献 [10] 中 Bollegala 等也提出了一种基于情感分类的多源跨领域学习方法, 其实验结果也证明了多源算法的有效性.

现实世界中源域数据规模往往较大, 当我们开发多源跨领域算法时, 更加关注如何将其进一步发展为针对大规模数据集的快速算法. 这也是本文的主要关注点, 而上文所提的其他算法却很少关注. 罗杰斯特回归函数 (Logistic regression, LR)^[11] 被广泛应用于数据分类、语义识别 (Natural language processing, NLP)^[12] 等领域, 被认为是一种行之有效的方法, 所以本文将其应用在一致性函数模型上并提出了 MSCC (Multi-source cross-domain classification) 算法. 为了满足大规模数据集的快速运算引入 CDdual (Dual coordinate descent method)

理论^[13] 生成快速算法 MSCC-CDdual, CDdual 算法是最近出现的一种针对大样本罗杰斯特回归函数的快速学习方法. 本文中的人工数据、文本数据、图像数据的实验结果验证了无论是大样本数据集还是样本较少的数据集 MSCC-CDdual 都拥有高识别率和快速性. 需要指出的是本文从两个角度展示了 MSCC-CDdual 的快速性, 第一种是大样本量数据, 第二种是高维数据. 本文在以下三个方面做出了贡献:

1) 提出了一种新的一致性函数模型, 该模型方便我们将 MSCC 算法演化为面向大样本的快速算法. 所提出的算法模型可以减少单个训练域分类器的错分率并提高综合分类精度.

2) 提出了面向大样本的快速算法 MSCC-CDdual, 该算法被证明既适用于大样本数据集又适用于样本较少的数据集.

3) MSCC-CDdual 从高维数据角度展示了其运行速度优势.

本文组织如下, 第 1 节首先介绍“罗杰斯特回归函数”和“一致性模型”概念, 然后提出了 MSCC 算法和其求解步骤; 第 2 节提出了基于 CDdual 快速算法 MSCC-CDdual, 并分析了其对大样本数据的快速性; 第 3 节对文中所提算法进行了相关实验和分析; 第 4 节总结全文.

1 MSCC 算法

在本节中首先介绍了一些预备概念其中包括“罗杰斯特回归函数”、“一致性方法”, 然后介绍了一种多源域的跨领域算法公式, 并阐述了其中的一致性模型. 最后, 解释了如何将现有的罗杰斯特回归模型与多源跨领域算法模型相结合, 并在此基础提出了一种改进的跨领域算法 MSCC.

1.1 数据准备和罗杰斯特回归函数

D_s^1, \dots, D_s^m 表示的是 m 组带有标签属性的源域数据集 (m 大于 1), 每一组具体表示为 $D_s^l = \{(\mathbf{x}_i^l, y_i^l)\}_{i=1}^{n^l}$, 其中, \mathbf{x}_i^l 为第 l 组源域数据, $y_i^l \in \{-1, +1\}$ 为第 l 组源域标签数据, n^l 为第 l 组源域数据的规模. D_t 为不含标签属性的目标域数据集, 其具体表示为 $D_t = \{(\mathbf{x}_i)\}_{i=1}^n$, 其中, \mathbf{x}_i 为目标域数据, 其数据规模为 n . 我们设定所有源域与目标域均来自数据集 \mathbf{R}^κ , 其中 κ 为所有样本点的维数.

罗杰斯特回归函数是一种对概率函数 $P(Y|X)$ 进行预测的方法, 其中, Y 为离散值, X 可以是离散量或连续量. 在这里通过 LR 函数为每一个源域构造一个分类器. 罗杰斯特回归函数设定了一种基于参数的概率分布模型 $P(Y|X)$, 通过对训练数据的学习获取该参数值. 对于二分类问题 $Y = \{+1, -1\}$,

对应基于参数的罗杰斯特回归模型表示如下:

$$P(y = \pm 1 | \mathbf{x}; \mathbf{w}) = \frac{1}{1 + \exp(-y\mathbf{w}^T \mathbf{x})} \quad (1)$$

其中, \mathbf{w} 为 LR 模型的权重参数, 其值可通过最大后验概率 (Maximum a posteriori, MAP) 理论^[14] 应用拉普拉斯先验概率^[9] 求得 (通过式 (2) 求得). 如果给定数据集 $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, 可以通过式 (2) 的最大化求得 \mathbf{w} , 观察式 (2) 发现其为一相对于 \mathbf{w} 的凹函数^[9] 可采用非线性优化方法求其最优解.

$$C \sum_{i=1}^N \log \frac{1}{1 + \exp(-y_i \mathbf{w}^T \mathbf{x}_i)} - \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (2)$$

C 是预设的正则化参数. 当求得 \mathbf{w} 后代入式 (1) 可以用来计算样本点属于正负类的概率.

1.2 一致性方法和一致性模型

在跨领域学习中一般源域 $D_s^1, D_s^2, \dots, D_s^m$ 的数据分布虽然不同但彼此相关, 同时目标域与源域的数据分布也不相同. 我们的目的是通过对带有标签的多源域数据的学习, 更好地指导不含标签属性的目标域分类.

算法中假定针对 m 个源域都设计了 m 个分类器, 即为 h^1, \dots, h^m . 在理想情况下源域与目标域满足相同的数据分布, 此时每一个分类器都能够高精度地预测目标域数据 D_t . 但现实情况是 D_s 本身内部之间分布存在差异, D_s 与 D_t 分布也存在差异. 这些 m 组分类器都存在一定的偏向性, 在不同程度上不能准确预测或分类 D_t . 这就要求在做目标域数据预测时, 要进一步最大化各个分类器的一致性, 即各个源域分类器对目标域数据的预测结果尽量相似. 在文献 [9] 中, Zhuang 等将这种一致性融入在一种标准模型下, 其模型算法表示如下:

$$\max \sum_{l=1}^m P(h^l | D_s^l) + \theta \cdot \text{Consensus}(h^1, h^2, \dots, h^m | D_t) \quad (3)$$

其中, $P(h^l | D_s^l)$ 为 h^l 相对于 D_s^l 的后验概率, $\text{Consensus}(h^1, h^2, \dots, h^m | D_t)$ 为针对各源域分类器 h^1, h^2, \dots, h^m 预测目标域的一致性方法, 其中 D_t 为不含标签属性的目标与数据, θ 为一权重调节参数. 参考文献 [9], 此时式 (3) 中的第一部分等价

于式 (4):

$$\begin{aligned} \max \sum_{l=1}^m P(h^l | D_s^l) = \\ \max \sum_{l=1}^m \left(C \sum_{i=1}^{n^l} \log \frac{1}{1 + \exp(-y_i^l \mathbf{w}^{lT} \mathbf{x}_i^l)} - \frac{1}{2} \mathbf{w}^{lT} \mathbf{w}^l \right) \end{aligned} \quad (4)$$

现在考虑式 (3) 中的第二部分, 在文献 [9] 中 Zhuang 等采用了基于香农熵的一致性方法 $C_e(h^1, h^2, \dots, h^m | D_t)$. 在这里通过观察给出了另一种一致性方法, 我们认为一种好的一致性方法应该使 $P(h^l | D_t)$ 和 $P(h^{l'} | D_t)$ 之间的差值平方项尽可能小. 在这里提出一种新的一致性方法如下:

$$\begin{aligned} \text{Consensus}(h^1, h^2, \dots, h^m | D_t) = \\ \min \sum_{t=1}^n \sum_{x_t \in D_t} \sum_{l, l'=1}^m (g(\mathbf{w}^l) - g(\mathbf{w}^{l'}))^2 \end{aligned} \quad (5)$$

其中, $g(\mathbf{w}^l) = P(h^l | \mathbf{x}_t) = P(y_t = \pm 1 | \mathbf{x}_t; \mathbf{w}^l) = 1 / (1 + \exp(-y_t \mathbf{w}^{lT} \mathbf{x}_t))$.

通过使不同源域分类器 ($h^l, h^{l'}$) 对应函数 ($g(\mathbf{w}^l), g(\mathbf{w}^{l'})$) 的差值平方项最小化, 可以使现有各源域模型能分享某种共性特征, 而这些特征往往也是目标域的特征.

将式 (5) 按泰勒公式在 $\mathbf{w}^{l'}$ 展开:

$$\begin{aligned} [g(\mathbf{w}^l) - g(\mathbf{w}^{l'})]^2 = [g'(\mathbf{w}^{l'}) (\mathbf{w}^l - \mathbf{w}^{l'})]^2 \leq \\ C_0 (\mathbf{w}^l - \mathbf{w}^{l'})^T \mathbf{x}_t \mathbf{x}_t^T (\mathbf{w}^l - \mathbf{w}^{l'}) \end{aligned} \quad (6)$$

具体推导见附录. 将式 (5) 一致性函数重新定义如下:

$$\begin{aligned} \text{Consensus}(h^1, \dots, h^m | D_t) = \\ \min \sum_{t=1}^n \sum_{x_t \in D_t} \sum_{l, l'=1}^m C_0 (\mathbf{w}^l - \mathbf{w}^{l'})^T \mathbf{x}_t \mathbf{x}_t^T (\mathbf{w}^l - \mathbf{w}^{l'}) \end{aligned} \quad (7)$$

设 $\mathbf{H} = \sum_{t=1}^n \mathbf{x}_t \mathbf{x}_t^T$ 式 (7) 变形为

$$\begin{aligned} \text{Consensus}(h^1, h^2, \dots, h^m | D_t) = \\ \min C_0 \sum_{l, l'=1}^m (\mathbf{w}^l - \mathbf{w}^{l'})^T \mathbf{H} (\mathbf{w}^l - \mathbf{w}^{l'}) \end{aligned} \quad (8)$$

式 (8) 为新提出的一致性方法, 将式 (4) 和 (8) 代入式 (3), 并将 θ 和 C_0 合并为 λ , 我们提出了新的基于

一致性方法的函数模型式 (9), 具体表示如下:

$$\begin{aligned} & \max f(\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^m) = \\ & \max \sum_{l=1}^m C \left(\sum_{i=1}^{n^l} \log \frac{1}{1 + \exp(-y_i \mathbf{w}^l \mathbf{x}_i)} - \frac{1}{2} \mathbf{w}^{lT} \mathbf{w}^l \right) - \\ & \lambda \sum_{l, l'=1}^m (\mathbf{w}^l - \mathbf{w}^{l'})^T \mathbf{H} (\mathbf{w}^l - \mathbf{w}^{l'}) = \\ & \max \sum_{l=1}^m C \left(- \sum_{i=1}^{n^l} \log(1 + \exp(-y_i \mathbf{w}^l \mathbf{x}_i)) - \right. \\ & \left. \frac{1}{2} \mathbf{w}^{lT} \mathbf{w}^l \right) - \lambda \sum_{l, l'=1}^m (\mathbf{w}^l - \mathbf{w}^{l'})^T \mathbf{H} (\mathbf{w}^l - \mathbf{w}^{l'}) \quad (9) \end{aligned}$$

只要求出 $\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^m$, 就能获得 m 组分类器 h^1, h^2, \dots, h^m , 利用式 (10) 就可以对目标域数据 \mathbf{x}_t 进行分类:

$$\bar{P}_t = \frac{\sum_{l=1}^m P(h^l | \mathbf{x}_t)}{m} \quad (10)$$

定理 1. 式 (9) 中的一致性函数模型可以提高各个分类器相对于目标域的分类精度.

证明. 仔细观察文献 [9] 中的定理 1 和定理 2, 这两个定理告诉我们最大化任意两个分类器的一致性, 可以提高各自分类器对目标域的识别精度. 需要指出的是这些定理的证明过程是独立的, 并不依赖于某一具体的一致性方法, 而本文的一致性方法是文献 [9] 中引申出的方法. 因此可以证明定理 1 的有效性.

本文以式 (8) 做为一致性方法, 其理论依据在于我们不仅通过参数 \mathbf{w} 考虑了不同源域分类器之间的差异性, 同时也通过 \mathbf{H} 中的目标域数据考虑了不同源域分类器与目标域分类器的差异性. 随后的实验结果证明了其有效性. 除了上述优点外, 式 (8) 还为 MSCC 向其快速版本 MSCC-CDdual 演化提供了必备条件, 由式 (8) 生成的新的一致性方法函数模型式 (9), 可以通过本文 2.2 节中的方法, 将其推导为与 CDdual^[13] 快速算法目标式 (12) 相似的对偶表达式 (26), 从而形成快速算法 MSCC-CDdual. MSCC-CDdual 算法正是一种基于 dual^[13] 梯度下降法的面向大数据集罗杰斯特回归模型的快速算法. 而文献 [9] 中基于香农熵的一致性方法, 其对应一致性函数模型 (见文献 [9] 中的式 (21)) 则无法优化为 CDdual^[13] 算法的目标式 (12), 继而不能采用 dual 的方法进行多源大样本快速运算, 文献

[9] 中基于香农熵的一致性算法 CCR (Centralized consensus regularization) 采用共轭梯度法优化求解, 在运算过程中涉及矩阵乘法其时间复杂度大于 $O(n^2 \times \kappa)$ (其中 n 为样本规模, κ 为样本维数), 而 MSCC-CDdual 算法的时间复杂度与样本规模呈线性关系 (具体分析见 2.4 节). MSCC-CDdual 算法在大样本运算速度方面要优于基于香农熵的一致性算法 CCR.

1.3 MSCC 细节描述

通过观察可以发现式 (9) 的前 1 项因与文献 [13] 中的基本模型相同, 故是凹函数, 第 2 项也是凹函数, 故式 (9) 是凹函数. 这里采用文献 [9] 中预先给定初始值的非线性最优化的方法进行求解. 本文采用共轭梯度法^[15] 作为 MSCC 的最优化算法. 在求解过程中将当前 \mathbf{w}^k 设为变量, 而其余 $\mathbf{w}^{k'}$ 设定为一常数然后逐一求出. 算法 1 为 MSCC 算法求解式 (9) 的细节过程描述.

算法 1. MSCC 求解过程

输入. 带有标签属性的源域数据集 $D_s^1, D_s^2, \dots, D_s^m$, 不含标签属性的目标域数据集 D_t , 误差阈值 $\varepsilon > 0$, 最大迭代次数 max. 输出. m 组分类器 $\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^m$.

步骤 1. 初始化 \mathbf{w}_0^l ($l = 1, 2, \dots, m$), $k = 0$;

步骤 2. For $l = 1, 2, \dots, m$ 设置搜索方向;

步骤 2.1. $d_0^l = \nabla f(\mathbf{w}_0^l)$

$d_{k+1}^l = \nabla f(\mathbf{w}_k^l) + \beta_k^l d_k^l$ ($\nabla f(\mathbf{w}_k^l)$ 为梯度值)

$\beta_k^l = \frac{-\nabla f(\mathbf{w}_k^l) \mathbf{A}_k^l d_k^l}{d_k^{lT} \mathbf{A}_k^l d_k^l}$

其中

$\nabla f(\mathbf{w}_k^l) = \frac{\partial f}{\partial \mathbf{w}_k^l} =$

$C \left(\sum_{i=1}^{n^l} (1 + \exp(-y_i^l \mathbf{w}_k^l \mathbf{x}_i^l)) y_i^l \mathbf{x}_i^l - \mathbf{w}_k^l \right) -$

$\lambda \mathbf{H} (\mathbf{w}_k^l - \mathbf{w}_0)$

\mathbf{w}_0 为一给定常数, $\mathbf{A}_k^l \in \mathbf{R}^{\kappa}$ 是针对 $f(\mathbf{w}_k^l)$ 的 $\kappa \times \kappa$ Hessian 矩阵, 其中 κ 是源域数据集的维数;

步骤 2.2. 判断是否为最优解, If $\left\| \sum_{l=1}^m \nabla f(\mathbf{w}_k^l) \right\| < \varepsilon$, 中断迭代并跳至步骤 5 (ε 为误差阈值);

步骤 3. For $l = 1, 2, \dots, m$ 求出搜索步长 λ_k^l :

$\lambda_k^l = \frac{-\nabla f(\mathbf{w}_k^l) d_k^l}{d_k^{lT} \mathbf{A}_k^l d_k^l}$

求出下次迭代点: $\mathbf{w}_{k+1}^l = \mathbf{w}_k^l + \lambda_k^l d_k^l$;

步骤 4. $k = k + 1$, If $k < MAX$, 返回至步骤 2;

步骤 5. 返回优化值 $(\mathbf{w}_k^1, \dots, \mathbf{w}_k^m)$.

这里需要指出的是 MSCC 是一种基础算法, 后续将进一步将其发展为针对大样本的 MSCC-CDdual 快速算法. 实验结果表明针对样本较少的数据集 MSCC 和 MSCC-CDdual 都有较好的收敛速度和分类精度, 因此, 我们建议 MSCC-CDdual 既可应用在样本较少的数据集上, 又可应用在大样本数据集上, 而 MSCC 可应用在样本较少的数据集上.

2 MSCC-CDdual 大样本快速算法

在本文的实验中, 当源域样本总量超过 4000 时, MSCC 算法就已无效, 其主要原因是需要重复计算 Hessian 矩阵 A_k^l , 而大样本数据的 A_k^l 的计算相当耗时、耗空间. 需要指出的是多源跨领域学习往往源域样本量较大, 例如随着英特网技术的发展, 我们能从互联网或者是维基网获得大量不同用户的信息, 而这些信息有许多潜在价值, 此时我们将其作为大样本源域信息进行训练, 就可以将结果用来识别大量未含标签属性的目标域数据. 在这一节中我们将 MSCC 与面向大样本罗杰斯特回归模型的 CDdual 相结合提出了大样本快速算法 MSCC-CDdual.

2.1 CDdual 理论

坐标下降法是一种解决无约束优化问题的技术, 但是该算法在解决大规模 LR 对偶问题中探索较少, 文献 [13] 中 Yu 等指出坐标下降法不仅可以解决 LR 原问题^[16] 而且可以用来求解 LR 的对偶问题, 算法名称命名为“对偶坐标下降法”简称 CDdual. 该算法已经被实验证明相比其他罗杰斯特回归模型算法有着较大的速度优势. 因此选择其做为 MSCC 算法的升级方案.

先回顾 LR 原问题的对偶形式:

$$\begin{aligned} \min_{\alpha} D^{\text{LR}}(\alpha) &= \frac{1}{2} \alpha^T Q \alpha + \sum_{i=1}^N \alpha_i \log \alpha_i + \\ &\quad (C - \alpha_i) \log(C - \alpha_i) \\ \text{s.t. } &0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \end{aligned} \quad (11)$$

其中, Q 的分量 $Q_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$.

CDdual 算法是一个逐步迭代的过程, 在每一次迭代中优化 α 的一个分量 α_i , 也就是说每次迭代只需要解一个单变量的子优化问题, 如果在子优化问题的解决过程中, 使用高效的方法会使得整个运算的收敛速度加快, 从而适合大样本数据的运算, 为此 CDdual 采用了一种改进了的牛顿迭代优化算法, 使用坐标下降法解决该对偶问题, 其过程与求解 LR 原优化问题类似. 整个算法的外迭代过程由 α^0 开始, 得到一个序列 $\{\alpha^k\}_{k=0}^{\infty}$, α^{k+1} 是对上一步的 α^k 的迭代更新. α^k 由各个子优化分量组成 $\alpha^k = [\alpha_1^k, \alpha_2^k, \dots, \alpha_N^k]$, CDdual 算法通过改进的牛顿迭代法解决单变量 α_i 的子优化问题如下:

$$\begin{aligned} \min_z g(z) &= (c_1 + z) \log(c_2 + z) + \\ &\quad (c_2 - z) \log(c_2 - z) + \frac{a}{2} z^2 + bz \\ \text{s.t. } &-c_1 \leq z \leq c_2 \end{aligned} \quad (12)$$

其中, $c_1 = \alpha_i, c_2 = C - \alpha_i, a = Q_{ii}, b = y_i \mathbf{w}^T \mathbf{x}_i$

在子优化过程中为了加速牛顿迭代法的方向选取又将式 (12) 进一步优化为如下两个对偶问题:

$$\begin{aligned} \min_z g_1(z_1) &= (z_1) \log(z_1) + (s - z_1) \\ &\quad \log(s - z_1) + \frac{a}{2} (z_1 - c_1)^2 + b_1 (z_1 - c_1) \\ \text{s.t. } &0 \leq z_1 \leq s, \quad b_1 = b \end{aligned} \quad (13)$$

$$\begin{aligned} \min_z g_2(z_2) &= (z_2) \log(z_2) + (s - z_2) \\ &\quad \log(s - z_2) + \frac{a}{2} (z_2 - c_2)^2 + b_2 (z_2 - c_2) \\ \text{s.t. } &0 \leq z_2 \leq s, \quad b_2 = -b \end{aligned} \quad (14)$$

其中, $s = c_1 + c_2$.

依照上述公式 CDdual 的求解过程如算法 2 和算法 3 中所示. 更多具体细节可参考文献 [13], 该文献显示 CDdual 的运行速度明显大于其他罗杰斯特回归模型算法是一种适用于大样本的快速算法.

算法 2. CDdual 求解过程

CDdual 求解 LR 对偶问题

步骤 1 (初始化). 输入数据集 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, 初始化阈值 $\varepsilon_1, \varepsilon_2$.

For $i = 1, 2, \dots, N$,

$\alpha_i = \min(\varepsilon_1 C, \varepsilon_2), \quad \alpha_i' = C - \alpha_i, \quad Q^{ii} = \mathbf{x}_i^T \mathbf{x}_i, \quad \mathbf{w} = \sum_i \alpha_i \mathbf{x}_i y_i;$

步骤 2. While (当 α 非最优解), 外部迭代;

步骤 2.1. For $i = 1, 2, \dots, N$,

令式 (12) 中的 $c_1 = \alpha_i, c_2 = \alpha_i', a = Q_{ii}, b = y_i \mathbf{w}^T \mathbf{x}_i;$

步骤 2.2. 应用改进的牛顿迭代优化法求解式 (12) 的最优解 z 并返回值 z_1, z_2 (见表 3), 在表 3 中 $s \equiv c_1 + c_2 = C;$

步骤 2.3. $\mathbf{w} = \mathbf{w} + (z_1 - \alpha_i) y_i \mathbf{x}_i;$

步骤 2.4. $\alpha_i = z_1, \alpha_i' = z_2.$

算法 3. 改进牛顿迭代法解子优化问题 (式 (13), (14))

步骤 1 (初始化). $s = c_1 + c_2, z_t^0 \in (0, s), z_m = \frac{c_2 - c_1}{2};$

步骤 2. $t = \begin{cases} 1, & z_m \geq \frac{-b}{a} \\ 2, & z_m < \frac{-b}{a} \end{cases};$

步骤 3. For $k = 0, 1, 2, \dots;$

步骤 3.1. If $\nabla g_t'(z_t^k) = 0$ break;

步骤 3.2. $d = \frac{-\nabla g_t(z_t^k)}{\nabla^2 g_t(z_t^k)};$

步骤 3.3. $z_t^{k+1} = \begin{cases} \xi z_t^k, & \text{若 } z_t^k + d \leq 0, \xi \in (0, 1); \\ z_t^k + d, & \text{否则} \end{cases}$

For end

步骤 4. $z_2^k = s - z_1^k, \quad \text{若 } t = 1;$
 $z_1^k = s - z_2^k, \quad \text{若 } t = 2;$

步骤 5. Output $(z_1^k, z_2^k).$

2.2 MSCC-CDdual 模型的建立

在本节中我们将 MSCC 算法与 CDdual 算法相结合, 形成面向大样本多域的快速算法 MSCC-CDdual. 而结合两种算法的关键是能否将式 (9)

转化为类似 CDdual 对偶表达式 (11), 并以此求解 $\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^m$. 随后将给出解答.

当采用 CDdual 求解第 l 组训练域的最优解 \mathbf{w}^l 时, 其他无关的源域参数 \mathbf{w}^k ($k \neq l$) 可以被设定为常数 \mathbf{w}_0 . 这样当我们求解 \mathbf{w}^l 时, 式 (9) 可被简化如下优化表达式:

$$\min L = \frac{1}{2} \|\mathbf{w}^l\|^2 + c \sum_{i=1}^{n^l} g(\xi_i^l) + \lambda(\mathbf{w}^l - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w}^l - \mathbf{w}_0)$$

其中

$$g(\xi_i^l) = \log \frac{1}{1 + e^{\xi_i^l}}, \quad \xi_i^l = -y_i^l \mathbf{w}^{lT} \mathbf{x}_i^l \quad (15)$$

对比式 (9), $c = -C$, 然后通过拉格朗日法将式 (15) 进一步优化为如下表达式:

$$\begin{aligned} \min L = & \frac{1}{2} \|\mathbf{w}^l\|^2 + c \sum_{i=1}^{n^l} g(\xi_i^l) + \\ & \lambda(\mathbf{w}^l - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w}^l - \mathbf{w}_0) + \\ & \sum_{i=1}^{n^l} \alpha_i^l (-\xi_i^l - y_i^l \mathbf{w}^{lT} \mathbf{x}_i^l) \end{aligned} \quad (16)$$

其中 α_i^l 为拉格朗日系数 ($i = 1, 2, \dots, n^l$). 参考文献 [16] 的优化方法, 对 L 中的各个参数分别求导结果如下:

$$\frac{\partial L}{\partial \mathbf{w}^l} = \mathbf{w}^l - \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^l + \lambda \mathbf{H}(\mathbf{w}^l - \mathbf{w}_0)$$

设 $\frac{\partial L}{\partial \mathbf{w}^l} = 0$, 得:

$$\mathbf{w}^l = (\mathbf{I} + \lambda \mathbf{H})^{-1} \left(\lambda \mathbf{w}_0 + \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^l \right) \quad (17)$$

其中, \mathbf{I} 为单位矩阵, 设 $\frac{\partial L}{\partial \xi_i^l} = 0$, 得到下式:

$$\begin{aligned} \frac{\partial L}{\partial \xi_i^l} = & c \frac{e^{\xi_i^l}}{1 + e^{\xi_i^l}} - \alpha_i^l = c g'(\xi_i^l) - \alpha_i^l = 0 \\ \text{i.e. } g'(\xi_i^l) = & \frac{\alpha_i^l}{c}, \quad \xi_i^l = g'^{-1} \left(\frac{\alpha_i^l}{c} \right) \end{aligned} \quad (18)$$

请注意当 $\xi_i^l \rightarrow \infty$ 时, $\frac{e^{\xi_i^l}}{1 + e^{\xi_i^l}} \rightarrow 1$ 所以 $0 \leq \alpha_i^l \leq c$. 设 $\delta = \alpha_i^l / c$ 构造函数 $G(\delta) = \delta \xi_i^l - g(\xi_i^l)$, 同时可知:

$$G'(\delta) = \delta \frac{d\xi_i^l}{d\delta} + \xi_i^l - g'(\xi_i^l) \frac{d\xi_i^l}{d\delta} = \xi_i^l = g'^{-1}(\delta) \quad (19)$$

因为 $g(\xi_i^l) = \log \frac{1}{1 + e^{\xi_i^l}}$, 所以 $g'(\xi_i^l) = \frac{e^{\xi_i^l}}{1 + e^{\xi_i^l}}$, 同时得到下式:

$$\begin{aligned} G(\delta) = & \delta \log \delta + (1 - \delta) \log(1 - \delta) \\ \text{i.e. } G\left(\frac{\alpha_i^l}{c}\right) = & \frac{\alpha_i^l}{c} \log \frac{\alpha_i^l}{c} + \left(1 - \frac{\alpha_i^l}{c}\right) \log \left(1 - \frac{\alpha_i^l}{c}\right) \end{aligned} \quad (20)$$

因此, 式 (16) 中的 $c \sum_{i=1}^{n^l} g(\xi_i^l)$ 可推导为

$$\begin{aligned} c \sum_{i=1}^{n^l} g(\xi_i^l) = & c \sum_{i=1}^{n^l} G\left(\frac{\alpha_i^l}{c}\right) = \\ & \sum_{i=1}^{n^l} [\alpha_i^l \log \alpha_i^l + (c - \alpha_i^l) \log(c - \alpha_i^l)] \end{aligned} \quad (21)$$

现在我们考虑式 (15) 中的其余部分 $\frac{1}{2} \|\mathbf{w}^l\|^2 + \lambda(\mathbf{w}^l - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w}^l - \mathbf{w}_0)$.

将式 (17) 代入可得:

$$\begin{aligned} \frac{1}{2} \|\mathbf{w}^l\|^2 = & \frac{1}{2} \left(\lambda \mathbf{w}_0^T + \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^{lT} \right) \mathbf{C} \times \\ & \left(\lambda \mathbf{w}_0 + \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^l \right) \end{aligned} \quad (22)$$

其中, $\mathbf{C} = [(\mathbf{I} + \lambda \mathbf{H})^{-1}]^2$.

$$\begin{aligned} \lambda(\mathbf{w}^l - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w}^l - \mathbf{w}_0) = \\ \lambda(\mathbf{w}^{lT} \mathbf{H} \mathbf{w}^l - 2\mathbf{w}^{lT} \mathbf{H} \mathbf{w}_0) \end{aligned} \quad (23)$$

因为式 (15) 是关于 L 的极小值最优化问题, 而 $\mathbf{w}_0^T \mathbf{H} \mathbf{w}_0$ 是常数项不影响求解过程, 所以可以在式 (23) 中舍去. 将式 (17) 代入式 (23) 结果如下:

$$\begin{aligned} \mathbf{w}^{lT} \mathbf{H} \mathbf{w}^l = & \left(\lambda \mathbf{w}_0^T + \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^{lT} \right) \mathbf{D} \times \\ & \left(\lambda \mathbf{w}_0 + \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^l \right) \\ \mathbf{D} = & (\mathbf{I} + \lambda \mathbf{H})^{-1} \mathbf{H} (\mathbf{I} + \lambda \mathbf{H})^{-1} \\ \mathbf{w}^{lT} \mathbf{H} \mathbf{w}_0 = & \left(\lambda \mathbf{w}_0^T + \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^{lT} \right) (\mathbf{I} + \lambda \mathbf{H})^{-1} \mathbf{H} \mathbf{w}_0 \end{aligned}$$

这里需要指出的是 $\mathbf{H}, \mathbf{C}, \mathbf{D}$ 为 $\kappa \times \kappa$ 矩阵, 可以依据目标域数据预先求出. 则进一步简化为

$$\frac{1}{2} \|\mathbf{w}^l\|^2 + \lambda(\mathbf{w}^l - \mathbf{w}_0)^T \mathbf{H}(\mathbf{w}^l - \mathbf{w}_0) = \frac{1}{2} \boldsymbol{\alpha}^{lT} \mathbf{Q}^l \boldsymbol{\alpha}^l + \sum_{i=1}^{n^l} E_i^l \quad (24)$$

其中, $Q_{ij} = y_i^l y_j^l \mathbf{x}_i^{lT} (\frac{\mathbf{C}}{2} + \lambda \mathbf{D}) \mathbf{x}_j^l$, $E_i^l = \alpha_i^l y_i^l \mathbf{x}_i^{lT} [\mathbf{C}/2 + \lambda \mathbf{D} - (\mathbf{I} + \lambda \mathbf{H})^{-1} \mathbf{H}] \mathbf{w}_0$.

将式 (21) 与式 (24) 结合, 可以获得式 (15) 的对偶表达式用来求解 \mathbf{w}^l , 具体如下:

$$\begin{aligned} \min \quad & \frac{1}{2} \boldsymbol{\alpha}^{lT} \mathbf{Q}^l \boldsymbol{\alpha}^l + \sum_{i=1}^{n^l} \alpha_i^l E_i^l + \\ & \sum_{i=1}^{n^l} [\alpha_i^l \log \alpha_i^l + (c - \alpha_i^l) \log (c - \alpha_i^l)] \\ \text{s.t.} \quad & 0 \leq \alpha_i^l \leq c, \quad i = 1, \dots, n^l \end{aligned} \quad (25)$$

通过比较式 (12) 和式 (25), 得出结论我们能可以通过 CDdual 算法求解式 (25) 的最优化问题, 从而可以求出 \mathbf{w}^l . 我们只需针对式 (25) 重新定义 CDdual 的子问题求解函数:

$$\begin{aligned} \min_z \quad & g(z) = (c_1 + z) \log(c_2 + z) + (c_2 - z) \times \\ & \log(c_2 - z) + \frac{a}{2} z^2 + bz \\ \text{s.t.} \quad & -c_1 \leq z \leq c_2, \quad c_1 = \alpha_i^l, c_2 = c - \alpha_i^l, \\ & a = Q_{ii}^l, b = (\mathbf{Q}^l \boldsymbol{\alpha}^l)_i + \alpha_i^l E_i^l \end{aligned} \quad (26)$$

一旦求出 α_i^l ($i = 1, 2, \dots, n^l$) 就能够根据式 (17) 很容易求出 \mathbf{w} , 同时根据式 (10) 可以对任一不含标签属性的目标域数据进行分类.

2.3 MSCC-CDdual 细节描述

通过上面的方法本文将 MSCC 与 CDdual 算法结合起来形成大样本快速算法 MSCC-CDdual. 在此总结 MSCC-CDdual 算法并给出具体求解过程见算法 4.

算法 4. MSCC-CDdual 求解过程

输入. 带有标签的源域数据集 $D_s^1, D_s^2, \dots, D_s^m$, 不带标签的测试目标域数据集 D_t

输出. m 组分类器 $\mathbf{w}^1, \mathbf{w}^2, \dots, \mathbf{w}^m$.

步骤 1 (初始化). 设定阈值 $\varepsilon_1, \varepsilon_2$, $k = 1$

$\alpha_i^{l(k)} = \min(\varepsilon_1 c, \varepsilon_2)$, $\alpha_i^{l'(k)} = c - \alpha_i^{l(k)}$,

for $i = 1, 2, \dots, n^l, l = 1, 2, \dots, m$,

$Q_{ii}^l = y_i^l y_i^l \mathbf{x}_i^{lT} (\frac{\mathbf{C}}{2} + \lambda \mathbf{D}) \mathbf{x}_i^l$

$\mathbf{w}^l = (\mathbf{I} + \lambda \mathbf{H})^{-1} (\lambda \mathbf{w}_0 + \sum_{i=1}^{n^l} \alpha_i^l y_i^l \mathbf{x}_i^l)$;

步骤 2. For $l = 1, 2, \dots, m$

While $\boldsymbol{\alpha}^l(k)$ is not optimal

For $i = 1, 2, \dots, n^l$;

步骤 2.1. 令 $c_1 = \alpha_i^{l(k)}, c_2 = c - \alpha_i^{l(k)}, a = Q_{ii}^l, b = (\mathbf{Q}^l \boldsymbol{\alpha}^l(k))_i + \alpha_i^{l(k)} E_i^l$;

步骤 2.2. 应用改进的牛顿迭代优化法求解式 (26), 具体采用算法 3 求得 z_1 和 z_2 ;

步骤 2.3. $\mathbf{w}^l = \mathbf{w}^l + (z_1 - \alpha_i^{l(k)}) y_i^l \mathbf{x}_i^l$;

步骤 2.4. $\alpha_i^{l(k)} = z_1 \quad \alpha_i^{l'(k)} = z_2$.

For end

$k = k + 1$

While end

$l = l + 1$

For end

2.4 MSCC-CDdual 算法复杂度分析

现有的大部分半监督学习算法时间复杂度一般为 $O(N^2)$ (N 训练集数据规模), 它们并不适用于大规模数据集的快速求解. MSCC-CDdual 算法由两层迭代组成, 算法最后收敛到局部最优解, 其外部迭代为算法 4 中的 While 循环, 内部迭代为算法 4 中 While 内部的 For 循环. 其中外部迭代计算量较小, 而主要计算量都在内部迭代中. 在每次内部迭代中只需要解决单变量的子优化问题, 在解决子优化问题时使用牛顿搜索法, 在内部迭代求解 \mathbf{w}^l 的过程中一共有三个步骤: 1) “子函数构造” 通过计算 $c_1 = \alpha_i^{l(k)}, c_2 = c - \alpha_i^{l(k)}, a = Q_{ii}^l, b = (\mathbf{Q}^l \boldsymbol{\alpha}^l(k))_i + \alpha_i^{l(k)} E_i^l$ 满足式 (13) 和 (14) 的最优化求解, 在这一部分由于 Q_{ii}^l 可以事先存储, 故计算量体现在计算 b 的复杂度. 在这里 E_i^l 中的 $\mathbf{C}/2 + \lambda \mathbf{D} - (\mathbf{I} + \lambda \mathbf{H})^{-1} \mathbf{H}$ 可以预先计算并存储, 这样可使 b 的时间复杂度由 $O(n\kappa)$ 缩减为 $O(\kappa)$ 其中 κ 为每一个样本向量的维数. 2) 算法 3 中“牛顿方向的求解” 依据文献 [13] 中的分析其时间复杂度为 $O(1)$. 3) 算法 3 中下一迭代位置的选取与计算 $z_t^{k+1} = \begin{cases} \xi z_t^k, & \text{若 } z_t^k + d \leq 0 \\ z_t^k + d, & \text{否则} \end{cases}$, 这里改进

了牛顿迭代算法去除了线性搜索迭代位置的昂贵时间开销, 对照文献 [13] 中的分析其时间复杂度为 $O(1)$. 如果设定外部迭代 While 的平均迭代次数为 M , 则 MSCC-CDdual 算法的时间复杂度为 $O(\sum_{l=1}^m (M \times n^l \times (\kappa + \#Newton \text{ steps} \times O(2)))) = O((\sum_{l=1}^m n^l) \times M \times (\kappa + \#Newton \text{ steps} \times O(2)))$

其中, $\#Newton \text{ steps}$ 为牛顿迭代法的平均迭代次数与文献 [13] 中结果相似, 该值在本文实验中的大多数情况下为 1. 观察后发现 MSCC-CDdual 其时间复杂度和训练域的样本规模之和呈线性关系, 这

说明该算法适合于大样本数据集的运算. 需要指出的是 MSCC-CDdual 还有另外一个优点, 其时间复杂度与源域或目标域数据集的维数 κ 呈线性关系, 适用于高维数据集运算. 我们将在第 3 节中对其进行分析.

从本质上讲 MSCC-CDdual 是一种半监督学习算法. 为了展示其相比现有半监督算法的优势我们将它们进行了对比. 为了公平我们采用只含有一个源域的 MSCC-CDdual 算法进行比较, 在这种情况下其时间复杂度为 $O(n \times M \times (\kappa + \# \text{Newton steps} \times O(2)))$, n 为带有标签的单个源域数据集规模. MSCC-CDdual 算法的时间复杂度是远远小于现有的 S^3VM (Semi-supervised support vector machine) 算法 S^3VM^{light} ^[17]、CCCP (Concave convex procedure)^[18] 的时间复杂度 $O(n_{sv}^3 + (n+u)^2)$, n_{sv} 为支撑向量, n 为带有标签的样本规模, u 为不带有标签的样本规模. 另外 3 种半监督学习算法 TSVM (Transductive support vector machines)^[19]、SGT (Spectral graph transducer)^[20]、CoCC (Coclustering based classification)^[5]. TSVM 和 SGT 的时间复杂度为 $O((n+u)^3)$, 当 n 增大时, 其时间复杂度明显高于 MSCC-CDdual. CoCC 的时间复杂度为 $O(n \times T \times (\kappa + 2))$, 其中最大迭代次数通常情况下为 10. 实验显示 MSCC-CDdual 的分类精度要高于 CoCC 的分类精度.

接着我们与一些现有的梯度下降算法进行比较. ∇S^3VM ^[21] 和 cS^3VM ^[22] 的时间复杂度为 $O((n+u)^3)$, TRON (Trust region Newton method)^[23] 的时间复杂度为 $O(M \times (n\kappa + \# \text{Conjugate gradient steps}) \times n\kappa)$, LBFGS (Limited memory quasi Newton implementation)^[24] 的时间复杂度为 $O(M \times (n+m)\kappa)m$ 通常取值 5. 对比发现 MSCC-CDdual 运行速度明显优于上述算法, 适合大样本数据和高维数据运算. 我们将在下节的实验中进行验证和分析.

3 实验结果及其分析

在本节中我们通过人工数据集和真实数据集验证算法的有效性、快速性. 所有实验都为二分类测试, 当然该算法也可以进一步扩展延伸至数据多分类本文在此不作说明.

3.1 实验说明与准备

本文针对所提算法安排了三组实验, 并与现有的一些算法进行了对比分析. 考虑到 MSCC-CDdual 从本质上讲是一种面向大样本的快速跨领域半监督算法, 在对比算法中包括了 2 种跨领域算法 (CCR^[9] 和 CoCC^[5])、两种半监督算法

(TSVM^[19] 和 SGT^[20])、两种针对大规模罗杰斯特回归函数模型的算法 (TRON^[23] 和 LBFGS^[24]).

第一组实验为人工数据, 主要验证算法的分类精度和对大样本数据的快速收敛性; 第二组实验为文本类型的真实数据集, 侧重于算法的参数选择和大样本验证, 并与 TSVM、SGT、CoCC、TRON 和 LBFGS 算法进行了对比分析. 第三组为多源域的图像数据从图像处理和高维运算的角度进行算法验证. 实验环境: Intel Core 2 GHz CPU, 1 GB RAM Windows XP, Matlab 7.5.0.

由于一般的标准数据集都不是为跨领域算法而设定的, 因此需要对数据集做些处理. 所有实验采用三组源域数据、一组目标域数据的模式, 所有数据组均取自人工数据集或真实数据集, 且每组都含有正负两种类别. 在数据准备上分为代表正类的 A 与代表负类的 B 两大父类别, 每一父类下又划分为属性近似但分布不同的 4 小子类分别为 $A_1, \dots, A_4, B_1, \dots, B_4$. 将 A_1, \dots, A_4 与 B_1, \dots, B_4 两两随机混合组成数据源 $D = A_i \cup B_j$ (每种组合都必须不同). 在数据源 D 中抽取 4 组作为实验数据, 其中三组作为源域数据 $D_s = D_s^1, \dots, D_s^3$ (大域), 另一组数据隐去标签属性作为目标域测试数据 D_t . 可知对于 $A_1, \dots, A_4, B_1, \dots, B_4$ 一共可以产生 $96 (4 \times P_4^4)$ 种实验组合方式.

为了对比算法在训练过程中的收敛速度, 在算法求解过程中将权重参数 \mathbf{w} 的“相对差异值” RDV (Relative difference of value) 作为衡量算法收敛速度的一重要指标, 该值表示如下:

$$\text{RDV} = \frac{P^{\text{LR}}(\mathbf{w}^*) - P^{\text{LR}}(\mathbf{w})}{P^{\text{LR}}(\mathbf{w}^*)}$$

其中, \mathbf{w}^* 为最优解, \mathbf{w} 为求最优解过程中的当前值. $P^{\text{LR}}(\mathbf{w})$ 的定义参考式 (2).

分类精度是实验结果的一个重要指标其结果按百分制定义, 具体表示为

$$\frac{\text{正确分类样本}}{\text{总样本数}} \times 100\%$$

为了使实验结果尽量公正, 在一组实验中选定最优参数并 10 次随机运行后, 用均值统计分类精度, 训练时间 Training (s). 需要指出的是在算法比较中 TRON 和 LBFGS 是两种针对大样本罗杰斯特回归模型的梯度下降算法, 其本身并不是大样本跨领域分类算法, 我们安排这两种算法与 MSCC-CDdual 进行对比, 主要目的是比较算法的收敛速度. 另外, TSVM、SGT、CoCC 这些算法是单源域跨领域算法, 此类算法做比较时, 需将本文实验中的三组源域数据 D_s^1, \dots, D_s^3 合并为一组训练集数据. 通过比较

可以发现 MSCC-CDdual 的收敛速度和分类精度都高于对应算法。

所有 MSCC 实验中的正则化参数 C 设定为 1, 最大迭代次数 \max 设定为 200, 误差阈值 ε 设定为 0.05, 所有 MSCC-CDdual 实验中的参数 c 设定为 -1 , $\varepsilon_1, \varepsilon_2$ 分别设定为 0.02, 0.06.

3.2 人工生成数据测试

人工数据采用三维双螺旋线数据, 这是一个二分类问题, 是人们普遍认可的检验学习算法的有效方法. 该问题要求将 x, y, z 坐标平面上 2 条不同的螺旋线上的点正确地分开. 螺旋线方程为

$$\begin{cases} x = (k\theta + \alpha) \cos(\theta) \\ y = (k\theta + \alpha) \sin(\theta) \\ z = \theta \end{cases} \quad (27)$$

其中, k 和 α 是常量, 分别代表速度和原始距离; θ 是以弧度为单位的相角. 取不同的 k 和 α , 可得到不同的螺旋线. 这里一共取 4 组双螺旋线, 通过 α^+ , α^- 的不同取值来区分正负类. 具体参数设定见表 1.

其具体图示如图 1 所示. 取图 1(b)~(d) 组做为源域数据 D_s^1, \dots, D_s^3 , 图 1(a) 做为目标域数据 D_t , 在 θ 为 $(0 \sim 10\pi)$ 的范围内选取样本点. 为了测试 MSCC 和 MSCC-CDdual 的运算速度 (训练

时间), 我们随机选取满足图 1(a) 分布的 2000 个点作为目标域数据, 并在每一个源域中随机选取 1000 到 150000 不等的数据进行实验. 在实验过程中 λ 设定值为 5 (参数设定见下文). 表 2 显示了不同数据规模下 CCR, MSCC 和 MSCC-CDdual 的平均训练时间和平均分类精度对比. 从表 2 中可以发现当源域数据样本大于 10000 时, MSCC 和 CCR 算法已无法运行 (见 “-”, Matlab 内存溢出错误), 而 MSCC-CDdual 算法可运行在表 2 任何数据规模下. 在样本容量较小时, MSCC-CDdual 运行速度较 CCR 和 MSCC 稍慢, 但随着样本容量的增大, 当样本量超过 4000 时, 其快速的优势就能体现且越加明显. 同时 MSCC-CDdual 与 CCR 和 MSCC 在样本较少的数据集下分类精度近似 (本文中其他实验结果类似后不做重复表述), 即 MSCC-CDdual 并没有因为优化过程而

表 1 双螺旋参数设定
Table 1 Four groups of 2-spirals

组别	k	α^+	α^-	θ
1	4	400	10	$(0, 10\pi)$
2	4	300	30	$(0, 10\pi)$
3	4	200	50	$(0, 10\pi)$
4	4	100	70	$(0, 10\pi)$

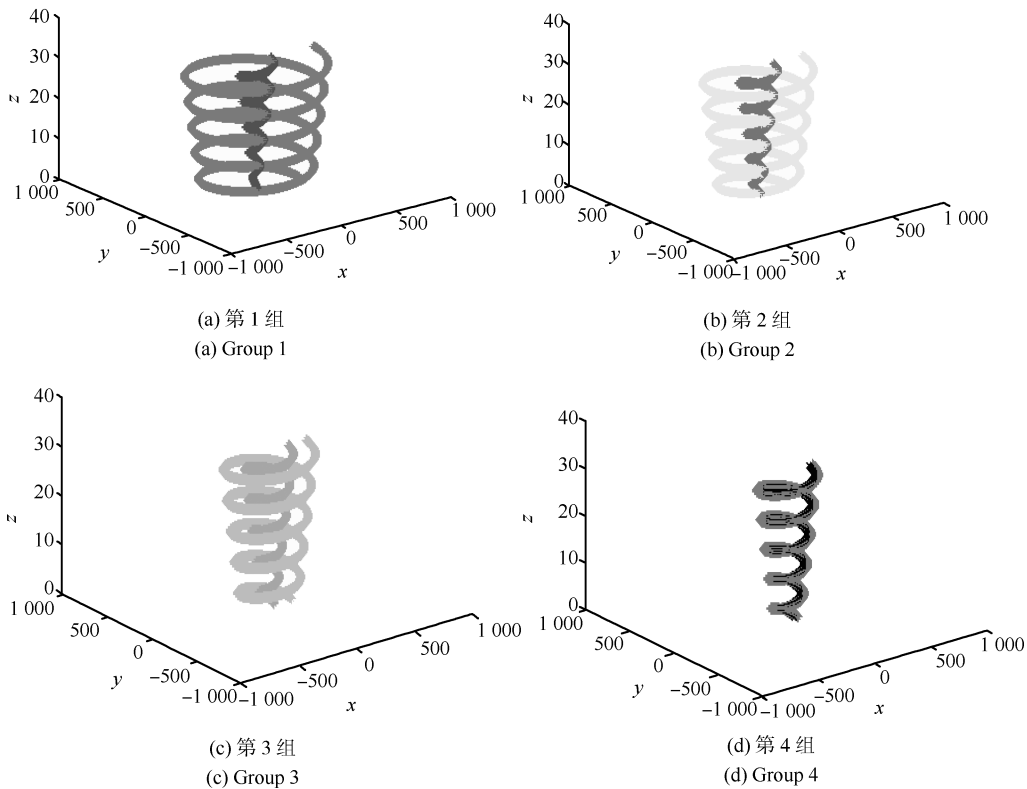


图 1 三维双螺旋样图
Fig.1 Four groups of 2-spirals

表 2 大样本运算时间对比
Table 2 Training time of both algorithms

数据	CCR	MSCC	MSCC-CDdual	MSCC	CCR	MSCC-CDdual
设定	训练时间 (s)	训练时间 (s)	训练时间 (s)	分类精度 (%)	分类精度 (%)	分类精度 (%)
1 000	0.73	0.60	0.79	92.23	91.09	90.64
2 000	0.86	0.81	0.84	93.62	92.31	91.18
3 000	1.07	0.89	1.02	93.39	92.47	91.21
4 000	1.93	1.09	1.25	93.43	92.51	92.15
5 000	3.56	2.12	1.33	93.55	92.64	93.06
6 000	6.94	5.89	1.45	93.97	92.73	93.11
8 000	12.98	10.49	1.67	94.71	93.27	94.26
10 000	—	—	1.72	—	—	94.79
40 000	—	—	5.90	—	—	95.21
80 000	—	—	11.41	—	—	95.28
100 000	—	—	13.55	—	—	96.62
120 000	—	—	22.82	—	—	96.82
150 000	—	—	31.96	—	—	97.73

损失太多精度. 这表明了 MSCC-CDdual 算法既适用于样本较少的数据又适用于大样本数据, 而图 2 则显示 MSCC-CDdual 时间复杂度和样本规模呈近似线性关系.

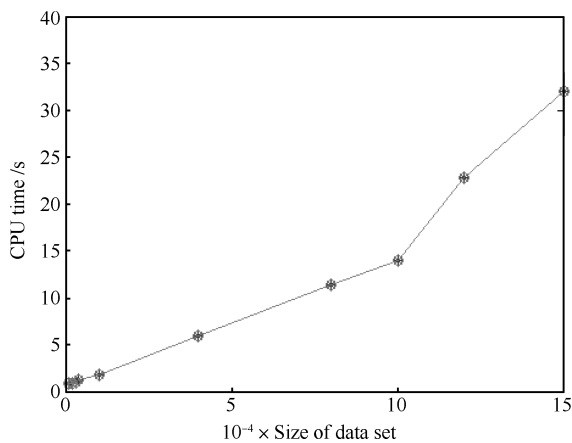


图 2 样本规模与运算时间

Fig. 2 Training time of MSCC-CDdual vs. data size

为了评估 MSCC-CDdual 算法在大样本下的收敛速度和分类精度, 我们将其与另外两种针对大样本罗杰斯特回归模型的 TRON 和 LBFSG 算法进行了对比实验. 图 3 显示了不同算法下的训练时间与 RDV 对比, 图 4 显示了不同算法下的训练时间与分类精度对比. 图 3 和图 4 中的实验每一组源域样本规模均为 15 万, λ 设定值为 5. 从图 3 和

图 4 中可以明显发现同一时刻 MSCC-CDdual 相比 TRON 和 LBFSG 有更小的 RDV 和更高的分类精度, 其收敛速度也更快.

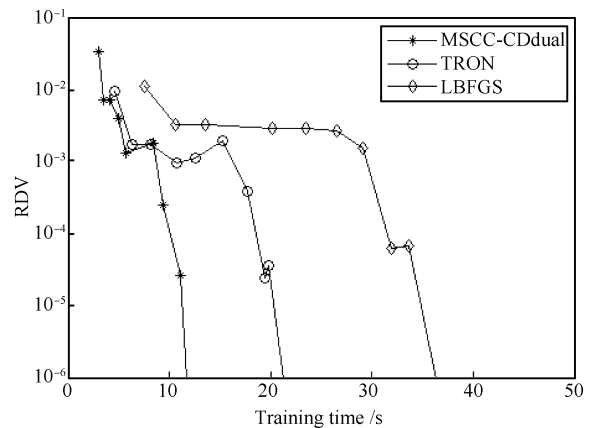


图 3 不同算法下 RDV 与运算时间

Fig. 3 RDV vs. training time

在 MSCC 和 MSCC-CDdual 算法中, 参数 λ 反映了一致性权重对分类精度有着很大的影响. 表 3 列出了 MSCC-CDdual 当源域样本规模为 10 万时, λ 不同取值与平均分类精度的对照关系. MSCC 算法的参数对比, 其结果与 MSCC-CDdual 算法类似. 通过观察表 3 可以发现当 λ 取值范围在 0 到 3 时, MSCC-CDdual 各个分类器的一致性在增强, 虽然对各自源域的分类精度有小幅下降, 但各自对目标

域的识别精度和综合精度 (具体定义见式 (9)) 都显著提高, 当 λ 达到 5 时综合精度最高. 当 λ 取值继续增大时 (尤其在 (≥ 10)) MSCC-CDdual 算法出现“过一致性” (Over-consensus), 因过度强调不同分类器的一致性而导致算法各个精度显著下降. 综合来看当 λ 取值范围设定在 3 到 5 时分类精度最佳. 需要说明的是不同数据规模下的 λ 取值规律都近似, 在这不做一一说明.

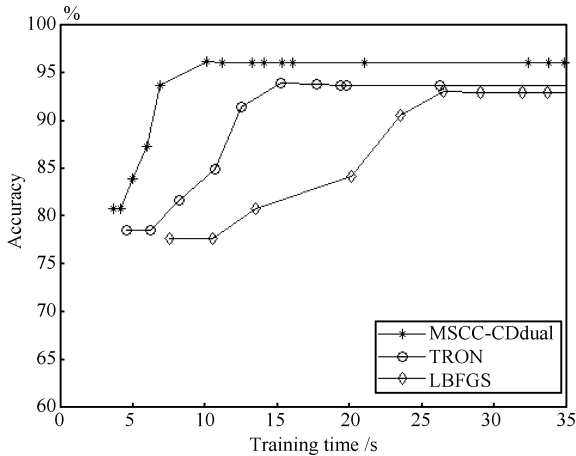


图 4 不同算法下识别精度与运算时间

Fig. 4 Classification accuracy vs. training time

4 文本分类数据实验

这组实验通过与 3 种跨领域半监督算法 TSVM、SGT 和 CoCC 以及两种针对大样本罗杰

斯特回归模型的算法 TRON 和 LBFSG 进行比较, 可以帮助我们观察 MSCC-CDdual 算法在分类精度与收敛速度与其他算法的差别. 所采用的数据为文本数据集 20 Newsgroups¹ 该数据集具有层次结构, 它的语料库中包含 7 个顶级类别, 在这 7 个顶级, 类别下面有 20 个子类别. 我们选择 comp vs. rec, sci vs. rec, talk vs. rec, comp vs. talk 4 种父类别分类, 相对应的具体正负类设置为各父类下的子类, 具体设定见表 4. 源域与目标域生成方式见第 3.1 节.

为了协调各分类器的一致性和自身分类精度, 需要对算法中的参数 λ 进行选择. 这里采用 talk vs. rec 数据组进行测试说明 (其他组情况类似), 具体测试结果如下:

为了使对比实验尽量公正, 我们首先要为 MSCC 和 MSCC-CDdual 寻找参数 λ 的最优值. 这里选择 talk vs. rec 进行实验, 观察表 5 可以发现当 λ 取值 0.1 时综合精度最高, 该值即为最优值. 其他组别实验结果类似在这不做列举.

对照表 4 每一种数据组设定都可构造出 96 种 ($4P_4^4$) 源域数据集和目标域数据集的组合. 我们选择 TSVM、SGT 和 CoCC, 3 种跨领域半监督算法与 MSCC 和 MSCC-CDdual 进行了比较, 图 5 和图 6 为 96 组实验分类精度对比图, 其中每一个源域随机抽取 8 000 非零数据, 目标域随机抽取 1 000 非零数据, 图 5 和图 6 中横坐标为 96 种数据集组合, 纵坐标为数据的识别精度, 其值为目标域数据在标签信息缺失下重复 10 次计算的平均值. 很明显 MSCC

表 3 参数选择与分类结果

Table 3 Classification accuracy of MSCC-CDdual with different λ

λ	h^1 分类精度 (%)		h^2 分类精度 (%)		h^3 分类精度 (%)		综合精度 (%)
	D_s	D_t	D_s	D_t	D_s	D_t	
0.00	83.17	75.51	82.90	73.65	85.49	79.65	76.27
0.20	85.08	95.93	84.04	92.23	86.46	90.43	92.23
0.40	86.84	96.59	85.10	93.09	87.13	91.03	93.15
1.00	89.19	97.71	87.23	94.53	88.62	92.64	94.55
2.00	90.58	98.30	89.40	96.05	90.23	93.82	96.06
3.00	90.79	98.35	90.44	96.41	91.18	94.53	96.43
5.00	90.46	98.20	90.87	96.59	91.78	95.11	96.62
10.00	87.76	96.86	89.31	95.94	91.32	94.67	95.94
15.00	85.50	95.96	87.66	94.90	90.28	93.85	94.90
20.00	83.50	94.80	86.44	93.99	89.46	93.29	93.99
50.00	78.11	91.44	82.70	91.02	86.70	90.73	91.02
120.00	74.58	89.11	79.99	88.97	84.73	88.84	88.97
300.00	72.72	88.11	78.70	88.08	83.52	88.02	88.08
500.00	72.22	87.79	78.13	87.79	83.25	87.72	87.79

¹<http://people.csail.mit.edu/jrennie/20Newsgroups/>

表 4 20 Newsgroups 数据设定
Table 4 Root categories and sub-categories from 20 newsgroups

数据组设定	A_1, \dots, A_4	B_1, \dots, B_4
comp vs. rec	comp.osmswindowmisc, comp.sysibmpchardware comp.sysmachardware, comp.graphicst	rec.motorcycles, rec.sportbaseball rec.autos, rec.sporthockey
sci vs. rec	sci.crypt, sci.electronics sci.med, sci.space	rec.motorcycles, rec.sportbaseball rec.autos, rec.sporthockey
talk vs. rec	talkreligionmisc, talkpoliticsgun talkpoliticsmisc, talkpoliticsmideast	rec.motorcycles, rec.sportbaseball rec.autos, rec.sporthockey
comp vs. talk	comp.osmswindowmisc, comp.sysibmpchardware comp.sysmachardware, comp.graphicst	talkreligionmisc, talkpoliticsguns talkpoliticsmisc, talkpoliticsmideast

表 5 参数选择与分类结果 (talk vs. rec)
Table 5 Classification accuracy of MSCC-CDdual with different λ for talk vs. rec

λ	h^1 分类精度 (%)		h^2 分类精度 (%)		h^3 分类精度 (%)		综合精度 (%)
	D_s	D_t	D_s	D_t	D_s	D_t	
0.00	96.49	78.69	92.58	78.10	92.58	77.62	78.12
0.10	93.51	81.30	81.68	82.33	80.79	80.48	93.83
0.12	93.53	81.39	79.98	80.79	79.38	78.74	92.93
0.15	93.77	81.83	77.84	78.91	75.51	75.78	87.83
0.18	94.22	83.13	75.00	76.67	71.36	71.66	85.92
0.20	94.52	83.78	72.63	74.48	67.54	67.68	84.70
0.30	96.18	88.70	61.95	63.94	55.28	57.03	77.58
0.40	96.90	91.18	50.76	52.98	37.43	39.84	68.40
0.60	33.94	86.72	28.03	31.65	27.90	31.65	51.12
1.00	23.67	63.87	28.03	31.65	27.90	31.65	31.65
2.00	10.34	31.65	28.03	31.65	27.90	31.65	31.65
3.00	10.34	31.65	28.03	31.65	27.90	31.65	31.65
5.00	10.34	31.65	28.03	31.65	27.90	31.65	31.65
10.00	10.34	31.65	28.03	31.65	27.90	31.65	31.65

和 MSCC-CDdual 在比较算法中分类精度最高. 这主要因为 MSCC 和 MSCC-CDdual 着重考虑了源域数据分布的差异性, 而 TSVM、SGT 和 CoCC 却没有考虑到这一点.

为了检验 MSCC-CDdual 算法在大样本文本数据下的收敛速度和精度, 我们选择 comp vs. rec 两大父类进行算法比较, 数据具体描述见表 6, 其中源域数据规模为原始组合大小, 目标域为 1000 非零数据. 图 7 显示了不同算法下 RDV 与训练时间的对比, 图 8 显示了不同算法下分类精度与训练时间的对比. 观察图 7 和图 8 可以发现同一时刻下 MSCC-CDdual 相比 TRON 和 LBFSGS 的 RDV 值更小, 分类精度更高, 收敛速度更快. 这主要是 MSCC-CDdual 算法继承了 CDdual 算法的快速性, 同时考虑了源域分布之间的差异性. 同样的结果也可以

在第 3.4 节的实验中观察到.

4.1 图像识别数据实验

这组实验是为了展示 MSCC-CDdual 算法在较高维多域跨领域分类方面的处理能力, 就像在第 2.4 节中指出的, 这个实验的本质是从较高维大数据量的角度衡量 MSCC-CDdual 的跨领域学习. 因此我们将 MSCC-CDdual 与两种针对大样本罗杰斯特回归模型的梯度下降算法 TRON 和 LBFSGS 进行了对比. 本文采用从 <http://wang.ist.psu.edu/docs/related.shtml> 下载的图像数据库. 该图库内有多个父类别, 选择父类别下近似类别作为子类. 本文选择 Traffic, Flower, Animal, Food 4 大父类作为分类目标, 每个父类下又含 4 个子类. 每个

子类都含有几十幅到上百幅分辨率为 128×96 的图片, 其对应分组情况见表 7. 图 9 显示 Traffic 父类下含有 Ttraffic.plane, Ttraffic.car, Ttraffic.boot, Ttraffic.ballon 4 个子类, Animal 父类下的 Animal.dinosaur, Animal.elephant, Animal.horse, Animal.lion 4 个子类. 然后按第 3.1 节中的方法构成源域数据和目标域数据, 算法任务是在目标域中进行父类级别上的分类.

表 6 comp vs. rec 数据描述
Table 6 Comp vs. rec data description

comp	非零项	rec	非零项
Osmswindows	83 848	Motorcycles	84 174
Sysibmpc	74 842	Sportbaseball	96 486
Sysmac	69 264	Autos	86 354
Graphicst	97 260	Sportshockey	95 386

每个子类都含有几十幅到上百幅图片不等. 将各父类下的子类分别用 A_1, \dots, A_4 和 B_1, \dots, B_4 表示, 然后交叉组合构成源域数据和目标域数据, 其

中源域和目标域的数据实验规模均为表 7 原始组合大小, 目标域算法任务是在目标域中进行父类级别上的分类. 算法处理过程中所有图片分辨率均为 128×96 . 从每幅图片采集其 87 维特征属性值, 其中包含 36 维的颜色直方图属性^[25] 和 51 维的纹理直方图属性^[26]. 其数据样本规模如表 7.

与前面的实验一样, 首先要为 MSCC-CDdual 算法选择合适的参数 λ . 这里选择 Traffic vs. animal 作为参数选择与分类结果测试对象 (其余数据组结果类似), 表 8 中的实验结果显示当 λ 为 0.3 时, 综合分类精度达到最大值, 所以选择该值作为最优值.

表 9 为不同图像分类组合下 MSCC-CDdual, TRON 和 LBGFS 三种算法在平均分类精度和平均训练时间上的对比. 从表 9 中可以很容易发现 MSCC-CDdual 算法相比其他罗杰斯特回归模型算法拥有最短的训练时间和最高的分类识别精度. 显示其具有较高的运行速度. 具体原因与上节实验分析类似.

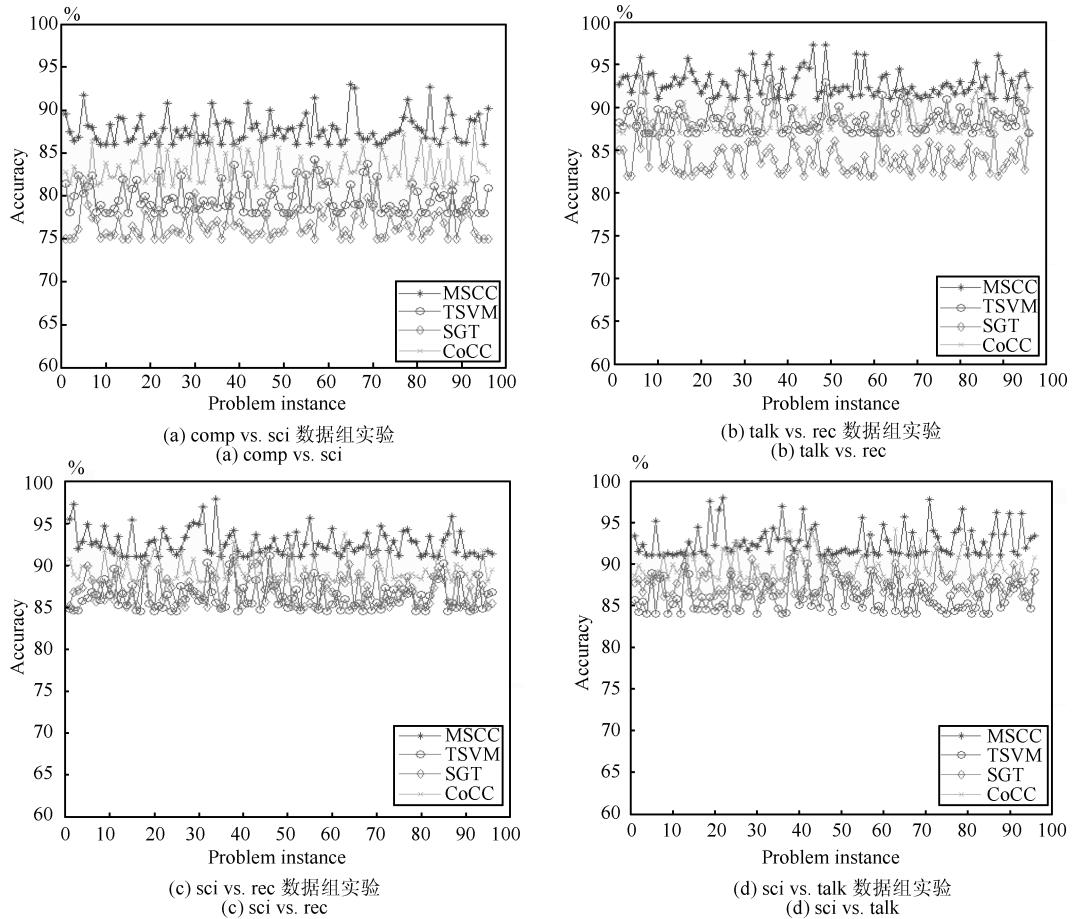


图 5 96 组对比精度图 (MSCC)

Fig. 5 Classification accuracy of the four algorithms with respect to 96 problem instances (MSCC)

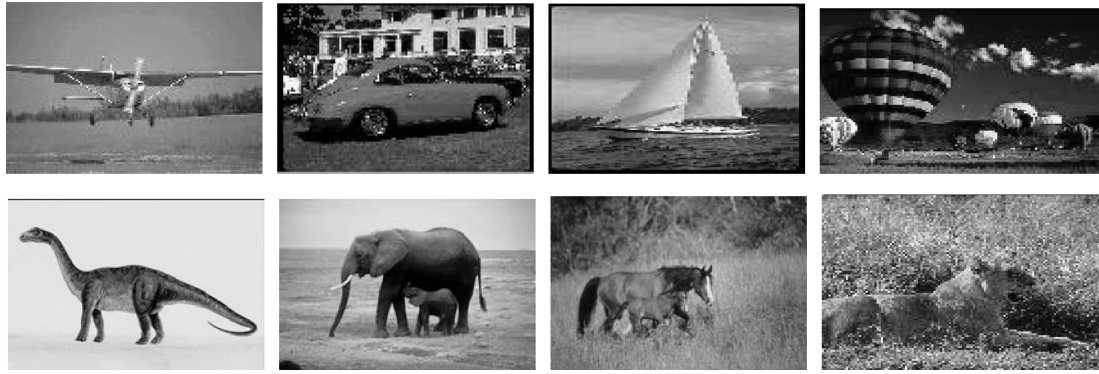


图9 图像数据

Fig.9 The sub-category images about Traffic and Animal

表8 参数选择与分类结果 (Traffic vs. Animal)

Table 8 Classification accuracy of MSCC-CDdual with different λ

λ	h^1 分类精度 (%)	h^2 分类精度 (%)	h^3 分类精度 (%)	综合分类精度 (%)
0	82.80	80.84	79.89	80.94
0.15	94.00	90.00	92.00	92.00
0.3	96.00	90.00	96.00	96.00
1.5	94.00	74.00	88.00	86.00
3.5	92.00	89.00	85.00	83.00
5.5	80.00	72.00	83.00	79.00

表9 三种算法的分类精度及训练时间对比

Table 9 Classification accuracy and training time of three algorithms on the image dataset

数据设定	MSCC-CDdual		LBGFS		TRON	
	分类精度 (%)	训练时间 (s)	分类精度 (%)	训练时间 (s)	分类精度 (%)	训练时间 (s)
Flower vs. Animal	92.33	3.3153	87.47	6.2385	89.66	4.5702
Traffic vs. Animal	96.06	3.4370	91.86	6.3126	93.97	4.3410
Flower vs. Food	86.28	3.5448	82.61	6.5768	84.94	4.7706
Flower vs. Traffic	83.37	3.4102	77.41	6.2210	79.81	4.4766
Average	89.50	3.4262	84.83	6.3372	87.09	4.5396

5 结论

本文研究了针对多源域的跨领域学习, 首先提出了 MSCC 算法, 该算法依据已有的最大一致性函数模型和牛顿梯度下降法完成跨领域学习. 与已有的跨领域算法不同的是 MSCC 面向的是多源域学习而不是单源域, MSCC 通过发掘多源域之间的分布差异来综合促进对目标域的学习. 然后, 依据针对大样本罗杰斯特回归模型的 CDdual 算法开发出了 MSCC 的快速版本 MSCC-CDdual. 通过原理分析以及人工数据集与真实数据集上的实验, 验证了无论是针对样本较少的多源数据集还是大样本多源数据集, MSCC-CDdual 都有较高的识别精度和识别

速度, 在下一步研究当中我们将重点针对高维且样本量较大的数据.

附录

式 (6) 的推导:

设 $\mathbf{w}^l, \mathbf{w}^{l'}$ 为 κ 维向量 $\mathbf{w}^l = (w_1^l, w_2^l, \dots, w_\kappa^l)^T$, $\mathbf{w}^{l'} = (w_1^{l'}, w_2^{l'}, \dots, w_\kappa^{l'})^T$, 将 $g(\mathbf{w}^l)$ 在 $\mathbf{w}^{l'}$ 处泰勒展开:

$$g(\mathbf{w}^l) = g(\mathbf{w}^{l'}) + \frac{\partial g}{\partial w_1^l}(w_1^l - w_1^{l'}) + \frac{\partial g}{\partial w_2^l}(w_2^l - w_2^{l'}) + \dots + \frac{\partial g}{\partial w_\kappa^l}(w_\kappa^l - w_\kappa^{l'}) \quad (28)$$

即 $g(\mathbf{w}^l) - g(\mathbf{w}^{l'}) = (\mathbf{w}^l - \mathbf{w}^{l'})^T \begin{bmatrix} \frac{\partial g}{\partial w_1^l} \\ \frac{\partial g}{\partial w_2^l} \\ \vdots \\ \frac{\partial g}{\partial w_\kappa^l} \end{bmatrix}$. 此时根据式 (5)

中 $g(\mathbf{w}^l)$ 的定义得到下式:

$$\begin{bmatrix} \frac{\partial g}{\partial w_1^l} \\ \frac{\partial g}{\partial w_2^l} \\ \vdots \\ \frac{\partial g}{\partial w_\kappa^l} \end{bmatrix} = \frac{\partial g}{\partial \mathbf{w}^l} = -\frac{\exp(-y_t \mathbf{w}^{lT} \mathbf{x}_t)}{(1 + \exp(-y_t \mathbf{w}^{lT} \mathbf{x}_t))} y_t \mathbf{x}_t^T$$

令 C' 为 $\frac{\exp(-y_t (\mathbf{w}^l)^T \mathbf{x}_t)}{(1 + \exp(-y_t (\mathbf{w}^l)^T \mathbf{x}_t))}$ 的上界, 注意: $y_t \in \{-1, +1\}$ 以及

$$(g(\mathbf{w}^l) - g(\mathbf{w}^{l'}))^2 = (\mathbf{w}^l - \mathbf{w}^{l'})^T \begin{bmatrix} \frac{\partial g}{\partial w_1^l} \\ \frac{\partial g}{\partial w_2^l} \\ \vdots \\ \frac{\partial g}{\partial w_\kappa^l} \end{bmatrix} \times \begin{bmatrix} \frac{\partial g}{\partial w_1^{l'}} & \frac{\partial g}{\partial w_2^{l'}} & \cdots & \frac{\partial g}{\partial w_\kappa^{l'}} \end{bmatrix}^T (\mathbf{w}^l - \mathbf{w}^{l'})$$

故易推得: $(g(\mathbf{w}^l) - g(\mathbf{w}^{l'}))^2 \leq (C')^2 (\mathbf{w}^l - \mathbf{w}^{l'})^T \mathbf{x}_t \mathbf{x}_t^T (\mathbf{w}^l - \mathbf{w}^{l'})$, 即 $(g(\mathbf{w}^l) - g(\mathbf{w}^{l'}))^2 \leq C_0 (\mathbf{w}^l - \mathbf{w}^{l'})^T \mathbf{x}_t \mathbf{x}_t^T (\mathbf{w}^l - \mathbf{w}^{l'})$, 其中 $C_0 = C' \times C'$.

References

- 1 Yang J, Yan R, Hauptmann A G. Cross-domain video concept detection using adaptive SVMs. In: Proceedings of the 15th International Conference on Multimedia. New York, USA: ACM, 2007. 188–197
- 2 Blitzer J, McDonald R, Pereira F. Domain adaptation with structural correspondence learning. In: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2006. 120–128
- 3 Pan S J, Tsang I W H, Kwok J T Y, Yang Q. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2011, **22**(2): 199–210
- 4 Dai W Y, Yang Q, Xue G R, Yu Y. Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning. New York, USA: ACM, 2007. 193–200
- 5 Dai W Y, Xue G R, Yang Q, Yu Y. Co-clustering based classification for out-of-domain documents. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. New York, USA: ACM, 2007. 210–219
- 6 Xing D K, Dai W Y, Xue G R, Yu Y. Bridged refinement for transfer learning. In: Proceedings of the 11th European Conference Practice of Knowledge Discovery in Databases. Berlin: Springer, 2007. 324–335
- 7 Suzuki T, Sugiyama M, Tanaka T. Mutual information approximation via maximum likelihood estimation of density ratio. In: Proceedings of the 2009 IEEE international conference on Symposium on Information Theory. NJ, USA: IEEE, 2009. 463–467
- 8 Suzuki T, Sugiyama M, Sese J, Kanamori T. Approximating mutual information by maximum likelihood density ratio estimation. In: Proceedings of the JMLR: Workshop and Conference Proceedings. NJ, USA: IEEE, 2008. 4: 5–20
- 9 Zhuang F Z, Luo P, Xiong H, Xiong Y H, He Q, Shi Z Z. Cross-domain learning from multiple sources: a consensus regularization perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2010, **22**(12): 1664–1678
- 10 Bollegala D, Weir D, Carroll J. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In: HLT'11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2011. 132–141
- 11 Hosmer D W, Lemeshow S. *Applied Logistic Regression*. Hoboken, NJ: John Wiley & Sons Press, 2001
- 12 Calí D, Condorelli A, Papa S, Rata M, Zagarella L. Improving intelligence through use of natural language processing. A comparison between NLP interfaces and traditional visual GIS interfaces. *Procedia Computer Science*, 2011, **21**(5): 920–925
- 13 Yu H F, Huang F L, Lin C J. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, 2011, **85**(1–2): 41–75
- 14 Gauvain J L, Lee C H. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *IEEE Transactions on Speech and Audio Processing*, 1994, **2**(2): 291–298
- 15 Ruszczyński A. *Nonlinear Optimization*. Princeton, NJ: Princeton University Press, 2006
- 16 Keerthi S S, Duan K B, Shevade S K, Poo A N. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 2005, **61**(1–3): 151–165
- 17 Joachims T. Making large-scale support vector machine learning practical. *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT Press, 1999. 169–184

- 18 Collobert P, Sinz P, Weston P, Bottou L. Large scale transductive SVMs. *The Journal of Machine Learning Research*, 2006, **7**: 1687–1712
- 19 Joachims T. Transductive inference for text classification using support vector machines. In: Proceedings of the 16th International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann, 1999. 200–209
- 20 Joachims T. Transductive learning via spectral graph partitioning. In: Proceedings of the 20th International Conference on Machine Learning. New York, USA: ACM, 2003. 290–297
- 21 Chapelle O, Zien A. Semi-supervised classification by low density separation. In: Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics. San Francisco, CA: Morgan Kaufmann 2005. 57–64
- 22 Chapelle O, Chi M M, Zien A. A continuation method for semi-supervised SVMs. In: Proceedings of the 23rd International Conference on Machine Learning. New York, USA: ACM, 2006. 185–192
- 23 Lin C J, Weng R C, Keerthi S S. Trust region Newton method for large-scale logistic regression. *Journal of Machine Learning Research*, 2008, **9**(4): 627–650
- 24 Deng W B. A limited memory quasi-Newton method for large scale problem. *Numerical Mathematics*, 1996, **5**(1): 71–79
- 25 Zhang Lei. The Research on Human-computer Cooperation in Content-based Image Retrieval [Ph. D. dissertation], Tsinghua University, China, 2001
(张磊. 基于人机交互的内容图像检索研究 [博士论文]. 清华大学, 中国, 2001)

- 26 Shi Z P, Ye F, He Q, Shi Z Z. Symmetrical invariant LBP texture descriptor and application for image retrieval. In: Proceedings of the 2008 Congress on Image and Signal Processing. Sanya, China: IEEE Computer Society, 2008. 825–829



顾鑫 江南大学数字媒体学院博士研究生, 工程师. 主要研究方向为人工智能, 模式识别. 图像处理. 本文通信作者.

E-mail: gurinbest@sina.com

(**GU Xin** Engineer, Ph.D. candidate at the School of Digital Media, Jiangnan University. His research interest covers artificial intelligence, pattern

recognition, and image processing. Corresponding author this paper.)



王士同 教授, 中国计算机学会高级会员. 主要研究方向为人工智能, 模式识别, 数据挖掘, 神经网络, 模糊系统, 医学图像处理和生物信息学.

E-mail: wxwangst@yahoo.com.cn

(**WANG Shi-Tong** Professor, senior member of China Computer Federation. His research interest covers arti-

ficial intelligence, pattern recognition, data mining, neural networks, fuzzy system, medical image processing, and bioinformation.)



许敏 江南大学数字媒体学院博士研究生. 主要研究方向为人工智能与模式识别. E-mail: xum@wxit.edu.cn

(**XU Min** Ph.D. candidate at the School of Digital Media, Jiangnan University. Her research interest covers artificial intelligence and pattern recognition.)