

基于相似度衡量的决策树 自适应迁移

王雪松¹ 潘杰¹ 程玉虎¹ 曹戈¹

摘要 如何解决迁移学习中的负迁移问题并合理把握迁移的时机与方法,是影响迁移学习广泛应用的关键点.针对这个问题,提出一种基于相似度衡量机制的决策树自适应迁移方法(Self-adaptive transfer for decision trees based on a similarity metric, STDT).首先,根据源任务数据集是否允许访问,自适应地采用成分预测概率或路径预测概率对决策树间的相似性进行判定,其亲和系数作为量化衡量关联任务相似程度的依据.然后,根据多源判定条件确定是否采用多源集成迁移,并将相似度归一化后依次分配给待迁移源决策树作为迁移权值.最后,对源决策树进行集成迁移以辅助目标任务实现决策.基于 UCI 机器学习库的仿真结果说明,与多源迁移加权求和算法(Weighted sum rule, WSR)和 MS-TrAdaBoost 相比,STDT 能够在保证决策精度的前提下实现更为快速的迁移.

关键词 迁移学习, 决策树, 相似度, 亲和系数

引用格式 王雪松, 潘杰, 程玉虎, 曹戈. 基于相似度衡量的决策树自适应迁移. 自动化学报, 2013, 39(12): 2186–2192

DOI 10.3724/SP.J.1004.2013.02186

Self-adaptive Transfer for Decision Trees Based on Similarity Metric

WANG Xue-Song¹ PAN Jie¹ CHENG Yu-Hu¹
CAO Ge¹

Abstract Negative transfer, transfer opportunity and transfer method are the most key problems affecting the learning performance of transfer learning. In order to solve these problems, a self-adaptive transfer for decision trees based on a similarity metric (STDT) is proposed. At first, according to whether the source task datasets to be allowed to access, a prediction probability based on constituents or paths is adaptively used to calculate the affinity coefficient between decision trees, which can quantify the similarity degree of related tasks. Secondly, a judgment condition of multi-sources is used to determine whether the multi-source integrated transfer is adopted. If do, the similarity degrees are normalized, which can be viewed as transfer weights assigned to source decision trees to be transferred. At last, the source decision trees are transferred to assist the target task in making decisions. Simulation results on UCI and text classification datasets illustrate that, compared with multi-source transfer algorithms, i.e., weighted sum rule (WSR) and MS-TrAdaBoost, the proposed STDT has a faster transfer speed with the assurance of high decision accuracy.

Key words Transfer learning, decision tree, similarity metric, affinity coefficient

Citation Wang Xue-Song, Pan Jie, Cheng Yu-Hu, Cao Ge. Self-adaptive transfer for decision trees based on similarity metric. *Acta Automatica Sinica*, 2013, 39(12): 2186–2192

收稿日期 2012-03-26 录用日期 2012-09-18
Manuscript received March 26, 2012; accepted September 18, 2012
国家自然科学基金(61072094, 61273143), 教育部博士点基金(20110095110016, 20120095110025), 江苏省研究生科研创新计划(CXZZ12_0932)资助
Supported by National Natural Science Foundation of China (61072094, 61273143), Special Grade of the Financial Support

迁移学习基于大脑对新涉猎知识的学习特点,借鉴人类与部分高智能动物在学习过程中的类比迁移机制,强调利用大量已掌握的和经验解决当前的新问题,可以在很大程度上降低学习新任务的难度.对于涉及海量数据处理与挖掘以及多任务优化与控制的信息产业、交通运输业与机器人动力学系统等行业或研究领域,均可利用迁移学习来实现问题的建模,因而迁移学习具有广泛的应用前景.然而,迁移学习在实际问题的应用中将不可避免地遇到两个难题,即何时迁移与如何迁移,迁移时机与方法的不当不仅无法促进当前任务的学习,反而会造成干扰,即产生“负迁移”,这将导致迁移学习在实际应用中的困难^[1].

在迁移学习领域,为解决“负迁移”问题,提高迁移效率,主要采用以下三种方法:实例迁移、特征迁移与模型迁移.实例迁移是最基本也是最直观的迁移学习方法,主要用来解决源任务与目标任务拥有相同或相近数据分布的问题,其核心思想是尽管源样本与目标样本分布不尽相同,仍然存在可供目标任务利用的有效样本实例,那么对这些样本实例妥善处理并辅助构建目标学习器将能够实现有效迁移.基于实例迁移的经典方法有 Transductive 方法^[2]、TrAdaBoost 算法^[3]以及在其基础上拓展为多源的 MS-TrAdaBoost 算法^[4].这类算法的优点是能够高效地利用源任务的知识 and 经验,辅助目标任务达成令人满意的精度,然而其对于源样本不断筛选与更新权值的迭代进程也使其拥有远远高于其他迁移算法的时间复杂度.另外,洪佳明等^[5]基于领域弱相似性的概念,提出了一种 TrSVM 算法,其能够将相似性约束条件与目标分类器联系起来,在训练过程中有效利用了相关领域的大量数据.Zadrozny^[6]则是将抽样误差理论规范至分类器学习中,并提出了在其影响下的各学习器偏差矫正方法.

相比于实例迁移,特征迁移具有更高的灵活性,其舍弃了源任务与目标任务间的数据一致性分布假设,降低了关联性要求,在很大程度上拓宽了待解决任务空间.Torrey 等^[7]提出的 RMT-D (Relational macro transfer via demonstration) 算法,将不同任务间的共同特征总结为关联宏(Relational macros, RM),通过 RM 来实现任务间的特征迁移,并将该技术用于 RoboCup 仿真足球领域.为了解决自然语言领域内的特征空间结构化迁移问题,Arnold 等^[8]提出一种分层优先级结构,将整体特征空间层次化、离散化,便于实现局部特征迁移以提高迁移的可靠性.而 Wang 等^[9]则是从辅助域与目标域的结构相似度入手,提取不同域的结构特征,并将其映射到再生核希尔伯特空间(Reproducing kernel Hilbert space, RKHS),在 RKHS 进一步估计相似特征的结构依赖性,为合理迁移提供依据.以上算法均以特征为纽带以实现迁移,实际问题中,任务间的共同特征往往难以归纳,同时这种有针对性的特征归纳迁移在面临不同特征空间的跨领域迁移问题时,将遭遇无共同特征归纳的难题.

模型迁移允许将源任务的学习器模型直接迁移以解决目标任务,目标数据仅对源学习器做局部修正.典型的迁移模型可以是遗传算法^[10]、马尔科夫逻辑网络^[11]、高斯过程模型

from China Postdoctoral Science Foundation (20110095110016, 20120095110025), and College Graduate Research and Innovation Projects of Jiangsu Province (CXZZ12_0932)

本文责任编辑 王红卫

Recommended by Associate Editor WANG Hong-Wei

1. 中国矿业大学信息与电气工程学院 徐州 221116

1. School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221116

型^[12] 以及决策树^[13] 等. 这类迁移方式的特点是直观形象, 更符合人类大脑的类比迁移思维, 然而对于机器而言, 如何判定任务间的关联性以便于选择合适的源任务模型进行迁移是个难题. 针对这个问题, 本文提出一种基于相似度衡量的决策树自适应迁移方法 (Self-adaptive transfer for decision trees based on a similarity metric, STDT). 不同于 TrSVM 基于概率分析的领域弱相似性判定条件, 本文的相似度衡量指标对学习器结构与成分更加有针对性, 同时避免了 Wang 等在 RKHS 中映射特征结构时, 出现的特征选择和优化问题. 本文算法在本质上属于一种多源的模型迁移算法, 与实例迁移相比, 迁移进程更加快速, 且能够克服特征迁移中特征难以总结的问题. 首先, 根据源任务数据是否可以访问, 自适应地采用路径预测概率或成分预测概率进行相似度判定; 其次, 由多源判定条件确定是否进行择优迁移并将任务亲和系数归一化后分配给各源任务; 最后, 将源任务统一集成迁移, 辅助实现目标任务. 采用 UCI 机器学习库以及 20 个新闻组 Newsgroups 的数据进行了仿真, 结果证明了算法的有效性.

1 决策树相似度衡量

给定数据集 D , 其拥有属性空间 \mathbf{R}^n , n 为属性个数. 决策树 DT 将 \mathbf{R}^n 划分为 Q 个区域, 每个区域 r_m 对应类标签 $r_m.c.l$. 决策树 DT 的作用相当于一个分段常值函数 $f_{DT} : \mathbf{x} \rightarrow r_m.c.l$, 将样本 $\mathbf{x} \in D$ 映射到对应的区域 r_m 中, 并输出相应的标签值 $r_m.c.l$.

一般来说, 决策树预测区域 r , 具有路径结构与成分结构两种表达形式, 其路径结构 $r.p$ 采用下式描述:

$$r.p = \{\cap d(a_v), v = 1, 2, \dots, K_r\} \quad (1)$$

其中, $d(a_v)$ 表示属性 a_v 在区域 r 内的取值范围, K_r 为根节点到区域 r 路径上的节点总数, 符号 \cap 表示各属性区间为相交的关系. 路径结构 $r.p$ 反映的是预测区域 r 与决策树 DT 间的结构联系, 给出了根节点到区域 r 所包含的属性以及规则集合.

为描述区域 r 中所包含数据集 D 的内容, 给出 r 的成分结构如下:

$$r.c = \{\text{num}(k_1), \text{num}(k_2), \dots, \text{num}(k_J)\} \quad (2)$$

其中, J 为类别总数, $\text{num}(k_1), \text{num}(k_2), \dots, \text{num}(k_J)$ 为进入区域 r 且分别属于 k_1, k_2, \dots, k_J 类的样本个数. 成分结构 $r.c$ 反映的是预测区域 r 与相应数据集 D 间的成分关系.

对于具有相关性且结构不同的决策树 DT_1 与 DT_2 而言, 可以通过样本预测概率的亲和系数来描述相似程度. 预测概率 $P(r)$ 根据训练数据集是否可以访问, 分为路径预测概率 $P(r.p)$ 和成分预测概率 $P(r.c)$, 其预测分量表达式如式 (3) 和式 (5) 所示^[11]:

$$P(r_m.p) = \frac{V(r_m.p)}{\sum_{l=1}^Q V(r_l.p)} \quad (3)$$

$$V(r_m.p) = \prod_{v=1}^{K_{r_m}} \frac{|d(a_v)|}{|\text{dom}(a_v)|} \quad (4)$$

$$P(r_m.c) = \frac{|r_m.c|}{\sum_{l=1}^Q |r_l.c|} \quad (5)$$

其中, 式 (4) 为区域 r_m 在属性空间 \mathbf{R}^n 归一化后的超体积, $|\text{dom}(a_v)| = \max(a_v) - \min(a_v)$, 表示属性 a_v 的全局取值范围, $|d(a_v)| = \max_{r_m}(a_v) - \min_{r_m}(a_v)$ 为属性 a_v 在区域 r_m 内的取值范围; 式 (5) 中 $|r_m.c| = \sum_{\rho=1}^J \text{num}(k_\rho)$ 为区域 r_m 中各类样本总和.

需要说明的是, 式 (3) 所得的路径预测概率 $P(r.p)$ 必须满足属性一致性分布假设, 否则会出现较大偏差. 因此, 通常情况下, 应尽可能采用式 (5) 获取预测概率 $P(r)$, 除非原训练集 D 已不可访问. 另外, 式 (3) 与式 (5) 给出的仅是预测分量, 完整的预测概率表达式为 $P(r) = \{P(r_m) | m = 1, 2, \dots, Q\}$.

获取 $P(r)$ 后, 采用下式可以计算得到不同决策树间的相似度^[14]:

$$S(DT_1, DT_2) = s(P_{DT_1}(r), P_{DT_2}(r)) = \sum_{m=1}^Q [P_{DT_1}(r_m) \cdot P_{DT_2}(r_m)] \quad (6)$$

其中, $s(\cdot, \cdot)$ 为亲和系数, 展开式如式 (6) 右侧所示, 其反映的是不同概率分布的相近程度, 且有 $0 < s(\cdot, \cdot) \leq 1$. 易知, $S(DT_1, DT_2)$ 具有相同的取值范围 (0, 1], 并且 DT_1 与 DT_2 的预测概率越相似, $S(DT_1, DT_2)$ 取值越接近 1; 否则, $S(DT_1, DT_2)$ 取值越接近 0. 当且仅当 $P_{DT_1}(r) = P_{DT_2}(r)$ 时, 有 $S(DT_1, DT_2) = 1$.

2 多源决策树自适应迁移及误差分析

对于多源迁移学习 $KT : S_1 \times S_2 \times \dots \times S_N \rightarrow T$ 而言, S_i ($i = 1, 2, \dots, N$) 为源任务, T 为目标任务, 不同 S_i 可获得相应源决策树 DT_i , 算法系统结构框图如图 1 所示, 总共分为 3 个阶段, 即源域决策树训练、相似度判定、多源集成迁移. 其中, 第 2 与第 3 阶段为算法的核心部分, 相似度判定机制根据源域数据集是否能够访问又分为基于成分与基于路径两种方法, 为图 1 中第 2 阶段判定框下左右两条分支, 分别简记为 STDT-C 与 STDT-P.

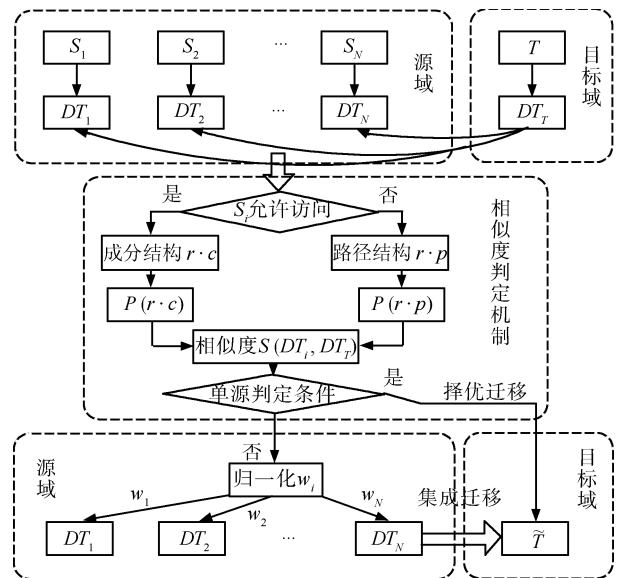


图 1 多源决策树自适应迁移系统结构框图
Fig. 1 Sketch map of self-adaptive transfer for multi-source decision trees

第 3 阶段实现的前提是进行如下单源迁移条件判定:

$$\max_i [S(DT_i, DT_T)] - \text{avg}_i [S(DT_i, DT_T)] > \theta \quad (7)$$

其中, $\text{avg}[\cdot]$ 表示取平均值, $\theta \in (0, 1)$ 为阈值参数, 其限定了进行单源择优迁移的门槛, 其值越接近 1, 则门槛越高, 即更偏向于执行多源集成迁移, 其值越接近 0, 则偏向于执行单源择优迁移, 通常情况取 $\theta = 0.2$. 若满足式 (7), 即表明源域中存在与目标任务相似度远远高于其同类的源任务, 此时, 进行单源择优迁移, 即选择 $\arg \max_i [S(DT_i, DT_T)]$ 作为迁移源辅助决策, 否则进行多源集成迁移.

采用决策树 DT 实现的各源任务 S_i 具有如下数学描述形式^[15]:

$$g(k|\mathbf{x}) = P(k|\mathbf{x}) + e_k(\mathbf{x}) \quad (8)$$

其中, $P(k|\mathbf{x})$ 为样本 \mathbf{x} 属于第 k 类的理想后验概率, $e_k(\mathbf{x})$ 为源任务 S_i 的估计误差, $g(k|\mathbf{x})$ 表示 \mathbf{x} 属于 k 类的估计概率.

由于估计误差 $e_k(x)$ 的存在, 使得实际分类边界 x_b 偏离于理想分类边界 x_b^* , 设 $b = x_b - x_b^*$, $p(x)$ 为样本 x 的概率分布, 那么可以得到单一决策树将类别 k_α 误分为类别 k_β 的概率^[15]:

$$E = \int_{x_b^*}^{x_b^*+b} |P(k_\alpha|\mathbf{x}) - P(k_\beta|\mathbf{x})| p(\mathbf{x}) d(\mathbf{x}) \quad (9)$$

其中, $\alpha, \beta \in [1, J]$ 且 $\alpha \neq \beta$.

通常情况下, b 值较小, 可将区间 $[x_b^*, x_b^* + b]$ 内的贝叶斯概率分布 $P(k|x)$ 以及样本分布作近似的线性化处理, 有 $P(k|x_b^* + b) \approx b \cdot P'(k|x_b^* + b) + P(k|x_b^*)$, $p(x) \approx p(x_b^*)$, 代入式 (9), 化简得:

$$E = \frac{p(x_b^*)}{2} b^2 \mu = \frac{p(x_b^*) \mu}{2} (\beta_b^2 + \sigma_b^2) \quad (10)$$

其中, β_b 与 σ_b^2 分别是 b 的偏差与方差, $\mu = P'(k_\alpha|x_b^* + b) - P'(k_\beta|x_b^* + b)$.

为实现多个源任务 S_i 对目标任务 T 的迁移, 考虑采用线性组合的方法对 T 进行辅助决策, 即:

$$g_T(k|\mathbf{x}) = \sum_{i=1}^N w_i g_i(k|\mathbf{x}) = P(k|\mathbf{x}) + e_k^T(\mathbf{x}) \quad (11)$$

对比式 (8) 可知, $e_k^T(\mathbf{x}) = \sum_{i=1}^N w_i e_k^i(\mathbf{x})$.

易知, 多源迁移分类器的误分类概率具有与式 (10) 相同的形式:

$$E_T = \frac{p(x_{bt}^*) \mu}{2} (\beta_{bt}^2 + \sigma_{bt}^2) \quad (12)$$

其中, x_{bt}^* 为多源组合分类器的理想分类边界, β_{bt} 、 σ_{bt}^2 为相应的边界偏移量 b_t 的偏差与方差.

由于实际分类边界 x_{bt} 上的样本点属于其相邻类别 k_α 与 k_β 的概率相等, 即 $g_T(k_\alpha|x_{bt}^* + b_t) = g_T(k_\beta|x_{bt}^* + b_t)$, 结合区间 $[x_{bt}^*, x_{bt}^* + b_t]$ 内的近似线性化条件, 可得 $b_t = \mu^{-1} [e_{k_\alpha}^T(x_{bt}) - e_{k_\beta}^T(x_{bt})]$, 因而有^[15]:

$$\begin{aligned} \beta_{bt}^2 &= \frac{1}{\mu^2} \sum_{i=1}^N w_i^2 (\beta_{k_\alpha}^i - \beta_{k_\beta}^i)^2 + \\ &\frac{1}{\mu^2} \sum_{i=1}^N \sum_{j \neq i} w_i w_j (\beta_{k_\alpha}^i - \beta_{k_\beta}^i) (\beta_{k_\alpha}^j - \beta_{k_\beta}^j) \end{aligned} \quad (13)$$

$$\begin{aligned} \sigma_{bt}^2 &= \frac{1}{\mu^2} \sum_{i=1}^N w_i^2 [(\sigma_{k_\alpha}^i)^2 + (\sigma_{k_\beta}^i)^2] + \\ &\frac{1}{\mu^2} \sum_{i=1}^N \sum_{j \neq i} w_i w_j (\rho_{k_\alpha}^{ij} \sigma_{k_\alpha}^i \sigma_{k_\alpha}^j + \rho_{k_\beta}^{ij} \sigma_{k_\beta}^i \sigma_{k_\beta}^j) \end{aligned} \quad (14)$$

其中, $\beta_{k_\alpha}^i$ 、 $\beta_{k_\beta}^i$ 分别表示源任务 S_i 估计误差 $e_{k_\alpha}^i(\mathbf{x})$ 与 $e_{k_\beta}^i(\mathbf{x})$ 的偏差; $\rho_{k_\alpha}^{ij}$ 、 $\rho_{k_\beta}^{ij}$ 表征 $e_{k_\alpha}^i(\mathbf{x})$ 与 $e_{k_\alpha}^j(\mathbf{x})$ 以及 $e_{k_\beta}^i(\mathbf{x})$ 与 $e_{k_\beta}^j(\mathbf{x})$ 间的相关系数; $(\sigma_{k_\alpha}^i)^2$ 、 $(\sigma_{k_\beta}^i)^2$ 则表示 $e_{k_\alpha}^i(\mathbf{x})$ 与 $e_{k_\beta}^i(\mathbf{x})$ 的方差.

假设源任务 S_i 估计误差 $e_k^i(\mathbf{x})$ 无偏且完全无关, 即 $\beta_{k_\alpha}^i = \beta_{k_\beta}^i = \beta_{k_\alpha}^j = \beta_{k_\beta}^j = 0$, $\rho_{k_\alpha}^{ij} = \rho_{k_\beta}^{ij} = 0$. 综合式 (12)~(14), 可得:

$$E_T = \frac{p(x_{bt}^*)}{2\mu} \sum_{i=1}^N w_i^2 [(\sigma_{k_\alpha}^i)^2 + (\sigma_{k_\beta}^i)^2] \quad (15)$$

迁移权值 w_i 的确定, 采用不同源决策树 DT_i 与目标决策树 DT_T 的相似程度来进行衡量. 考虑到 $\sum_{i=1}^N w_i = 1$, 有:

$$w_i = \left[\sum_{q=1}^N S(DT_q, DT_T) \right]^{-1} S(DT_i, DT_T) \quad (16)$$

将式 (6) 代入式 (15), 可得多源迁移学习的误分类概率:

$$\begin{aligned} E_T &= \frac{p(x_{bt}^*)}{2\mu} \left[\sum_{q=1}^N S(DT_q, DT_T) \right]^{-2} \\ &\sum_{i=1}^N S^2(DT_i, DT_T) [(\sigma_{k_\alpha}^i)^2 + (\sigma_{k_\beta}^i)^2] \end{aligned} \quad (17)$$

3 算法流程

步骤 1. 先验知识: 由源域中各数据集 S_i 训练的源决策树 DT_i , 以及目标域中的训练集 T 及目标决策任务 DT_T .

步骤 2. 依次判定 DT_T 与各源任务 DT_i 的相似度 $S(DT_T, DT_i)$, 若源数据集 S_i 允许访问, 采用式 (2) 计算区域 r 成分结构 $r.c$, 进一步通过式 (5) 获取成分预测概率 $P(r.c)$, 否则, 利用式 (1) 得到的路径结构 $r.p$, 求解式 (3) 给出的路径预测概率 $P(r.p)$.

步骤 3. 根据式 (7), 判定是否为单源迁移. 若是, 直接选取 $S(DT_T, DT_i)$ 最大的 DT_i 进行迁移, 对目标任务进行决策, 结束; 否则执行步骤 4.

步骤 4. 将各相似度 $S(DT_T, DT_i)$ 归一化, 得迁移权值 w_i , 依次分配给各源决策树.

步骤 5. 采用线性组合的方法进行迁移集成, 即目标任务决策树 $DT_T = \sum_{i=1}^N w_i \cdot DT_i$, 结束.

4 仿真研究

4.1 UCI 数据集

为验证所提算法的有效性, 采用 UCI 机器学习库中的 16 个数据集进行仿真, 数据集尺寸与属性个数如表 1 所示. 数据集分属 4 个不同领域, 即生命科学: Iris, Mushroom, Stalog (Heart), Ecoli, Acute Inflammation, Haberman's survival, Mammographic mass, SPECT heart; 社会科学: Balance scale, Nursery, Hayes-roth, Teaching assistant evaluation, Car evaluation, MONK's problem; 竞技体育: Chess (KR

vs KP); 物质结构: Wine. 其中, 属性跨度: 3 ~ 36, 类别跨度: 2 ~ 8, 样本跨度: 120 ~ 12960.

4.2 精度与复杂度分析

将本文提出的基于成分相似度判定的 STDT-C 算法、基于路径相似度判定的 STDT-P 算法与经典的 MS-TrAdaBoost 算法^[4] 以及决策树迁移 (Transfer in decision trees, TDT) 算法^[13] 进行对比, 仿真结果如表 2 所示. 为保证不同算法相互间的可比性, 4 种算法均采用 C4.5 决策树作为基学习器, 尽管与 Yao 在文献 [4] 中所采用的线性 SVM 不同, 然而 MS-TrAdaBoost 算法本身更注重的是样本权重与弱分类器权值的迭代更新, 对基学习器并无特殊的要求, 采用 C4.5 替换线性 SVM 并不影响 MS-TrAdaBoost 的本质. 同样地, Lee 等在文献 [13] 中采用了 ID3 作为基学习器, 仅能够处理离散属性问题, 改用 C4.5 后提高了其对连续属性问题的应对能力, 却不影响其在二次循环下对决策树叶节点的增补与修正.

由表 2 可以看出, 4 种算法对于分类精度而言, 以 STDT 与 MS-TrAdaBoost 算法为优, 而对于复杂度而言, 则是 STDT 优于 MS-TrAdaBoost, 两相比较下前者的仿真耗时

远远小于后者. 事实上, MS-TrAdaBoost 算法的复杂度不仅依赖于源任务的个数 N , 还与其最大迭代次数 M 有关, 精准的分类效果必然依靠设定较大的 M 值. Yao 等在文献 [4] 中就曾分析指出, MS-TrAdaBoost 算法的计算复杂度为 $C_h O(MN) + C_w O(Mn_s)$, 其中, C_h 与 C_w 分别为训练单一基学习器以及更新一次权值的复杂度, n_s 为迁移空间样本总数, 可以看出 M 值的选取对算法的复杂度起着至关重要的影响, 而大部分时候需要选取较大的 M 值, 即以时间为代价来换取分类精度. 对于 TDT 算法, Lee 等在文献 [13] 中给出了其复杂度的表达式 $O(d_s^2 n_{\max} + b_s^{d_s}) + O((d_t - d_s)d_t n_{\max})$, 其中, d_s 与 d_t 分别为源数据集与目标数据集的属性个数, n_{\max} 为源任务训练样本数与目标任务训练样本数中的较大者, b_s 则是源任务属性最大值. 显然, 从本质上来说, TDT 作为一种单源的二次迭代修正迁移算法, 复杂度必然小于多源多迭代的 MS-TrAdaBoost 算法. 两相比较下也可以看出, $O(d_s^2 n_{\max} + b_s^{d_s}) \approx C_h \ll C_h O(MN)$, 又由于 $O(n_s) \approx O(Nn_{\max}) > O(n_{\max})$, $O(M) \approx O(d_t^2)$, 因而有 $O((d_t - d_s)d_t n_{\max}) \approx O(d_t^2 n_{\max}) \ll C_w O(Mn_s)$, 故 TDT 算法的复杂度远远小于 MS-TrAdaBoost 算法, 体现在统计结果上即为表 2 中所呈现的两类算法在仿真耗时上的差异.

表 1 UCI 数据集说明
Table 1 UCI dataset

数据集	属性数	类别数	样本数	数据集	属性数	类别数	样本数
Iris	4	3	150	Hayes-roth	5	3	160
Balance scale	4	3	625	Ecoli	8	8	336
Car evolution	6	4	1 728	Acute inflammations	6	2	120
Chess (KR vs KP)	36	2	3 196	Haberman's survival	3	2	306
Mushroom	22	2	8 124	MONK's problems	7	2	432
Nursery	8	5	12 960	Mammographic mass	6	2	961
Statlog (Heart)	13	2	270	SPECT heart	22	2	267
Wine	13	3	178	TAE	5	3	151

表 2 UCI 数据集分类性能对比
Table 2 Performance comparison of different algorithms for UCI dataset

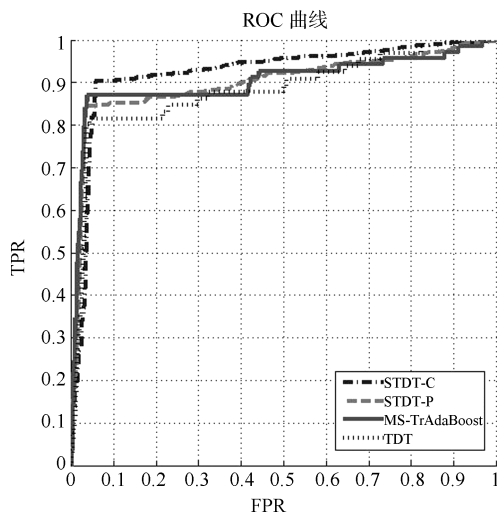
UCI 数据集	STDT-C		STDT-P		MS-TrAdaBoost		TDT	
	精度	耗时 (s)	精度	耗时 (s)	精度	耗时 (s)	精度	耗时 (s)
Iris	0.8558	0.0156	0.8423	0.0161	0.9395	1.5333	0.7873	0.0163
Balance scale	0.8035	0.0079	0.7554	0.0078	0.7955	0.7085	0.6631	0.0083
Car evaluation	0.7787	0.0210	0.7404	0.0211	0.7416	1.9778	0.7005	0.0217
Chess (KR vs KP)	0.9258	0.3450	0.9220	0.3390	0.9261	32.8913	0.6338	0.3435
Mushroom	0.7127	0.2857	0.6959	0.2678	0.7391	27.9687	0.6928	0.2868
Nursery	0.8921	0.1072	0.8922	0.1055	0.8403	10.9889	0.8281	0.1080
Statlog (Heart)	0.7840	0.0906	0.7807	0.0897	0.8580	8.7501	0.7576	0.0917
Wine	0.8136	0.3976	0.7785	0.3947	0.7882	9.6830	0.6536	0.3992
Hayes-roth	0.7083	0.0188	0.7531	0.0183	0.7023	1.7063	0.7634	0.0195
Ecoli	0.9014	0.0626	0.8913	0.0621	0.8524	6.1985	0.6448	0.0534
Acute inflammations	0.7563	0.0144	0.7899	0.0136	0.7995	1.2419	0.4118	0.0147
Haberman's survival	0.7148	0.0137	0.7311	0.0133	0.6984	1.3438	0.7377	0.0144
MONK's problems	0.8455	0.0195	0.7992	0.0194	0.8325	1.7610	0.5041	0.0200
Mammographic mass	0.8010	0.0154	0.8166	0.0151	0.8022	1.4095	0.6128	0.0157
SPECT heart	0.7595	0.0461	0.7962	0.0450	0.7468	4.3024	0.4937	0.0476
TAE	0.9260	0.0203	0.9334	0.0202	0.8933	1.9343	0.6400	0.0214

相比之下, STDT 则是综合了这两类算法的优点, 在复杂度上, STDT-C 与 STDT-P 分别为 $O((d_{\max}^2 n_{\max} + b_s^{d_{\max}})N) + C_c O(Nn_{\max})$ 与 $O((d_{\max}^2 n_{\max} + b_s^{d_{\max}})N) + C_p O(Nd_{\max})$, 其中, d_{\max} 为任务空间最大属性数, C_c 与 C_p 分别为计算单一成分预测概率 $P(r \cdot c)$ 与路径预测概率 $P(r \cdot p)$ 的复杂度, 其差异仅在于相似度判定的机理不同. 明显地, $O((d_{\max}^2 n_{\max} + b_s^{d_{\max}})N) \approx C_h O(N) > O(d_s^2 n_{\max} + b_s^{d_s})$, $C_c \approx C_p \approx C_w$, 而 $O(N) \approx O((d_t - d_s)d_t)$, 故 STDT 的复杂度优于 MS-TrAdaBoost 而劣于 TDT. 迁移精度方面, STDT 算法选择了多源机制, 与 TDT 相比精度更高, 同时克服了其对源任务选取的依赖性, 降低了负迁移的概率. 尽管不同于 MS-TrAdaBoost 的权值更新方法, STDT 的相似度判定机制也能很好地对多个迁移源分配权值, 从而保证了迁移精度.

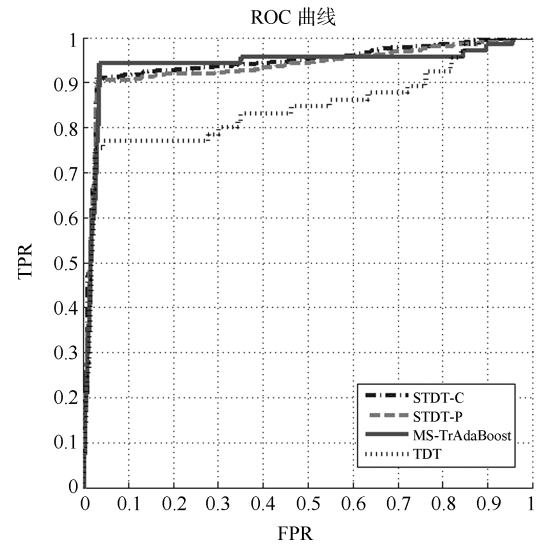
4.3 ROC 曲线与稳定性分析

为了从更细微的角度观察算法分类性能, 图 2 给出了 STDT-C、STDT-P、MS-TrAdaBoost 与 TDT 对于数据集 Balance scale 不同类别的受试者操作特性 (Receiver operating characteristics, ROC) 曲线. 一般来说, 对于 n 类的问题 ROC 空间维数将达到 $n^2 - n$, 当 $n > 2$ 时, 很难直观地呈现其仿真特性. 为解决这个问题, 针对 Balance scale 的每一类, 均生成一幅 ROC 曲线, 如图 2(a) ~ (c) 所示, 其中横坐标误检率 (False positive rate) 为假正类率, 即错分为正类的负类样本在全部负类中所占比例, 纵坐标 TPR (True positive rate) 为真正类率, 表示分类正确的正类样本在全部正类中的比例.

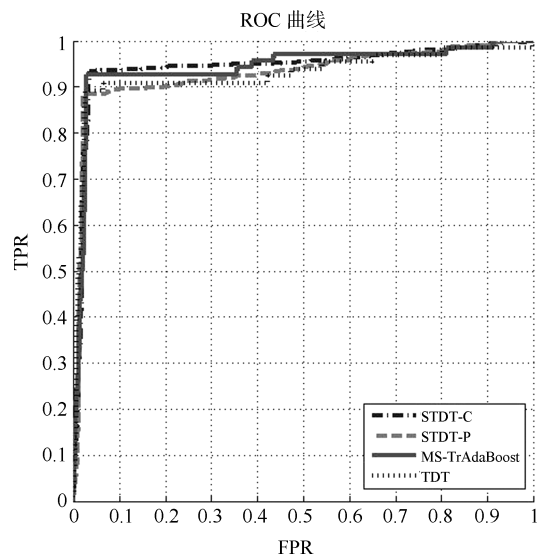
对于 ROC 曲线而言, 其越靠近左上, 表明分类的效果越好, 即能够以尽可能小的错分负类代价换取较高的正类划分率. 由图 2 可以看出, 在对 Balance scale 第 2 类和第 3 类的识别效果上, STDT-C 与 MS-TrAdaBoost 大致相同, 优于 STDT-P 与 TDT 算法, 而在第 1 类的识别上, 则是 STDT-C 优于其余三种算法, 综合而言, 在数据集 Balance scale 的分类效果上, 算法 STDT-C 为最优, 这也与表 2 呈现的数据相符. 对于算法 TDT, 由图 2 可以发现, 其对于不同类别的识别效果并不稳定, 在 Balance scale 第 1 类与第 3 类的识别上较好, 而在第 2 类上很差, 体现了单源迁移的局限性.



(a) Balance scale 第 1 类
(a) The 1st class of Balance scale



(b) Balance scale 第 2 类
(b) The 2nd class of Balance scale



(c) Balance scale 第 3 类
(c) The 3rd class of Balance scale

图 2 各算法不同类别下 ROC 曲线对比

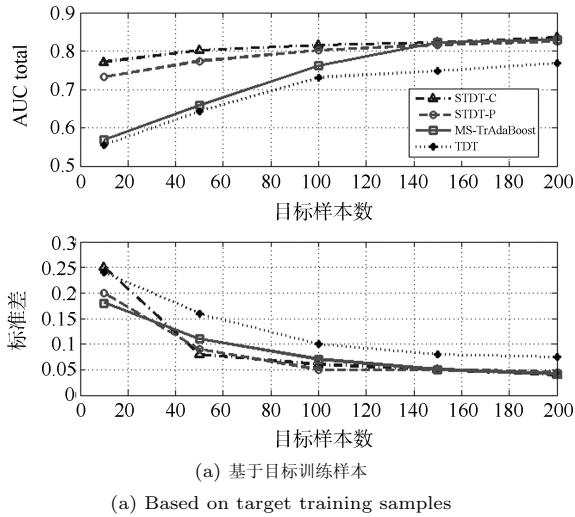
Fig. 2 ROC performance comparison of algorithms for different classes

为了更直观地观察不同算法的分类性能, 此处采用 AUC 指标予以衡量. 所谓 AUC, 即指 ROC 曲线下的面积, 对于多分类问题, 总体的 AUC 指标为每一类 AUC 的加权和, 即^[16]:

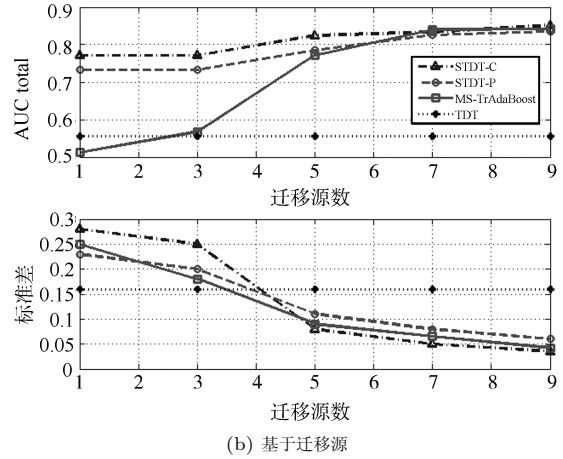
$$AUC_{\text{total}} = \sum_{k_i \in K} p(k_i) AUC(k_i) \quad (18)$$

其中, K 为类集合, $p(k_i)$ 为第 k_i 类样本所占比例, $AUC(k_i)$ 为第 k_i 类 ROC 曲线的 AUC 指标. 图 3 分别针对目标训练样本数以及迁移源数这两个不同角度, 给出了不同算法对于 AUC 以及 AUC 标准差两个指标的性能对比.

图 3(a) 中, 目标训练样本数为 $n_T = \{10, 50, 100, 150, 200\}$, 迁移源数为 $N = 3$, 图 3(b) 中, 迁移源数 $N = \{1, 3, 5, 7, 9\}$, 训练样本数为 $n_T = 10$. 由图 3(a) 可以看出, 随着 n_T 的增加, 4 类算法的 AUC 均呈上升趋势, 最终在到达 200 个训练样本时, 算法 STDT-C、STDT-P、MS-TrAdaBoost 的性能趋于一致. 尽管如此, 在 $n_T \in [10, 100]$ 阶段, STDT-C 与 STDT-P 的算法性能远远优于 MS-TrAdaBoost, 也就是说, 当目标训练样本数较少时, 本文所提出的基于相似度判定的集成迁移机制优于 MS-TrAdaBoost 的弱分类器样本权重更新机制. 另外, 从 AUC 标准差的变化趋势来看, 基于多源迁移的三类算法好于单源迁移的 TDT 算法, 说明了多源迁移能够在一定程度上提高算法的稳定性. 图 3(b) 是基于不同迁移源数的 4 种算法 AUC 性能指标对比, 可以看出, 经典的多源迁移算法 MS-TrAdaBoost 在迁移源数达到下限 $N = 1$ 时, 表现出了较差的性能, 而 STDT-C 与 STDT-P 算法的相似度判定机制则起到了良好的筛选性, 保证了算法在迁移知识极少的情況下仍然能保持令人满意的分类效果. 在 AUC 标准差的对比上, 与基于目标样本数的曲线变化趋势相同, 更多的迁移源保证了更加稳定的算法性能. 不同的是, 由内容拟合的相似度指标在迁移源数目较少时波动较大, 使得 STDT-C 在曲线的前半部分呈现出高于其余算法的 AUC 标准差, 这种情况在迁移源增多时得到了较好的改善, 当 $N > 7$ 时, 算法 STDT-C 成功地将 AUC 标准差控制在 0.05 以下, 使算法的分类精度保持了极高的稳定性.



(a) Based on target training samples



(b) Based on transfer sources

图 3 各算法 AUC 指标与 AUC 标准差对比 (Balance scale)
Fig. 3 Comparison of AUC and AUC standard deviation of different algorithms (Balance scale)

4.4 文本分类数据集

本节采用 20 个 Newsgroups 数据集进行相关算法的仿真对比. 数据集采集自近 20000 个新闻组文档, 涉及 20 个不同领域的内容, 如表 3 所示. 全部新闻组共分为 6 个不同的范畴, 即计算机: comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x; 记录: rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey; 科学: sci.crypt, sci.electronics, sci.med, sci.space; 销售: misc.forsale; 演讲: talk.politics.misc, talk.politics.guns, talk.politics.mideast; 宗教: talk.religion.misc, alt.atheism, soc.religion.christian.

表 3 给出了几种不同集成迁移算法的性能对比. 其中, STDT-C 是本文提出的基于成分相似度的自适应决策树迁移算法, 第 2 种是由 Li 等^[17] 提出的加权求和算法 (Weighted sum rule, WSR), 表 3 中出现的 MS-TrAdaBoost 采用 Yao 在原文中使用的支持向量机 (Support vector machine, SVM) 作为基分类器. 目标任务从 6 个不同领域中任选其一, 迁移源为相应领域内的其余任务. 同时由于 misc.forsale 为其领域内的唯一数据集, 因而采用了科学领域的文本集 sci.med 对其进行了替换. 由表 3 中数据可以看出,

表 3 文本分类性能对比

Table 3 Performance comparison of different algorithms for text classification

目标任务	迁移源数	STDT-C		WSR		MS-TrAdaBoost	
		精度	耗时 (s)	精度	耗时 (s)	精度	耗时 (s)
comp.graphics	4	0.9155	0.4631	0.8690	1.6243	0.9012	4.1318
rec.motorcycles	3	0.9112	0.3232	0.8622	1.3154	0.8936	3.2381
sci.crypt	3	0.8947	0.2987	0.9114	1.2194	0.9232	3.0854
sci.med	3	0.9245	0.3119	0.9392	1.1875	0.9485	3.2198
talk.politics.misc	2	0.9613	0.2019	0.9637	0.8124	0.9371	1.9882
alt.atheism	2	0.9193	0.2238	0.9207	0.8447	0.9306	2.6213

三类集成学习算法在分类精度方面相差不大, MS-TrAdaBoost 略优于 STDT-C, 而在分类耗时方面则有显著差异, STDT-C 是非参数依赖的无迭代算法, MS-TrAdaBoost 由于其循环迭代操作, 使得其耗时最大, 远远超过前两类算法。另外, 尽管 WSR 也不需迭代, 然而其在计算最优权值时引入了新的优化问题, 使得问题的求解过程变得复杂。

5 结论

提出一种相似度衡量的决策树自适应迁移方法, 根据预测概率的不同分为基于成分的 STDT-C 算法与基于路径的 STDT-P 算法。主要有以下特点: 1) 对迁移源数以及源任务数据集的自适应性, 既可以进行多源任务的集成迁移, 也可以实现单源任务的择优选择迁移, 并针对源数据集是否可以访问采用不同的概率预测方式; 2) 相似度判定机制使得迁移更加高效也更有针对性, 在很大程度上降低了负迁移的可能性; 3) 利用了模型迁移高效快速的特点, 避免了对源数据反复筛选迭代更新的过程, 提高了迁移时间效率, 与经典的多源迁移算法 MS-TrAdaBoost 相比拥有更低的时间复杂度; 4) STDT-C 允许访问源任务数据集, 使其预测概率更加准确, 相似度计算更为合理, 从而保证更好的迁移精度; 5) 对于源数据空间不可访问的情形, 可以采用 STDT-P 算法, 其基于路径判定预测概率的方式更适合于解决数据空间符合属性一致性分布假设的问题; 6) 与 MS-TrAdaBoost 和 TDT 相比, 当迁移源数较少时, STDT 拥有更小的标准差, 克服了迁移源数目造成的算法不稳定性。

References

- Pan S J, Yang Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2010, **22**(10): 1345–1359
 - Ceci M, Appice A, Barile N, Malerba D. Transductive learning from relational data. In: Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition. Leipzig, Germany: Springer-Verlag, 2007. 324–338
 - Dai W Y, Yang Q, Xue G R, Yu Y. Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning. Corvallis, USA: ACM, 2007. 193–200
 - Yao Y, Doretto G. Boosting for transfer learning with multiple sources. In: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, USA: IEEE, 2010. 1855–1862
 - Hong Jia-Ming, Yin Jian, Huang Yun, Liu Yu-Bao, Wang Jia-Hai. TrSVM: a transfer learning algorithm using domain similarity. *Journal of Computer Research and Development*, 2011, **48**(10): 1823–1830
(洪佳明, 印鉴, 黄云, 刘玉葆, 王甲海. TrSVM: 一种基于领域相似性的迁移学习算法. *计算机研究与发展*, 2011, **48**(10): 1823–1830)
 - Zadrozny B. Learning and evaluating classifiers under sample selection bias. In: Proceedings of the 21st International Conference on Machine Learning. Banff, Canada: ACM, 2004. 903–910
 - Torrey L, Shavlik J, Walker T, Malin R. Relational macros for transfer in reinforcement learning. In: Proceedings of the 17th International Conference on Inductive Logic Programming. Corvallis, USA: Springer-Verlag, 2008. 254–268
 - Arnold A, Nallapati R, Cohen W W. Exploiting feature hierarchy for transfer learning in named entity recognition. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Columbus, USA: ACL, 2008. 245–253
 - Wang H Y, Yang Q. Transfer learning by structural analogy. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence and the 23rd Innovative Applications of Artificial Intelligence Conference. San Francisco, USA: AAAI Press, 2011. 513–518
 - Koçer B, Arslan A. Genetic transfer learning. *Expert Systems with Applications*, 2010, **37**(10): 6997–7002
 - Mihalkova L, Huynh T, Mooney R J. Mapping and revising Markov logic networks for transfer learning. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence and the 19th Innovative Applications of Artificial Intelligence Conference. Vancouver, Canada: AAAI, 2007. 608–614
 - Yu K, Chu W. Gaussian process models for link analysis and transfer learning. In: Proceedings of the 2007 Annual Conference on Neural Information Processing Systems. Vancouver, Canada: Curran Associates, 2007. 1–8
 - Lee J W, Giraud C C. Transfer learning in decision trees. In: Proceedings of the 2007 International Joint Conference on Neural Networks. Orlando, USA: IEEE, 2007. 726–731
 - Ntoutsi I, Kalousis A, Theodoridis Y. A general framework for estimating similarity of datasets and decision trees: exploring semantic similarity of decision trees. In: Proceedings of the 8th SIAM International Conference on Data Mining. Atlanta, USA: Society for Industrial and Applied Mathematics Publications, 2008. 810–821
 - Fumera G, Roli F. A theoretical and experimental analysis of linear combiners for multiple classifier systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, **27**(6): 942–956
 - Fawcett T. An introduction to ROC analysis. *Pattern Recognition Letters*, 2006, **27**(8): 861–874
 - Li S S, Huang C R, Zong C Q. Multi-domain sentiment classification with classifier combination. *Journal of Computer Science and Technology*, 2011, **26**(1): 25–33
- 王雪松 中国矿业大学教授。主要研究方向为机器学习, 生物信息学。本文通信作者。E-mail: wangxuesongcumt@163.com
(WANG Xue-Song Professor at China University of Mining and Technology. Her research interest covers machine learning and bioinformatics. Corresponding author of this paper.)
- 潘 杰 中国矿业大学博士研究生。主要研究方向为迁移学习。E-mail: panjie1616@126.com
(PAN Jie Ph. D. candidate at China University of Mining and Technology. His main research interest is transfer learning.)
- 程玉虎 中国矿业大学教授。主要研究方向为机器学习, 智能优化与控制。E-mail: chengyuhu@163.com
(CHENG Yu-Hu Professor at China University of Mining and Technology. His research interest covers machine learning, intelligent optimization and control.)
- 曹 戈 中国矿业大学硕士研究生。主要研究方向为迁移学习。E-mail: caogexz@163.com
(CAO Ge Master student at China University of Mining and Technology. His main research interest is transfer learning.)