

基于平衡概率分布和实例的迁移学习算法

黄 露, 曾庆山

(郑州大学 电气工程学院 河南 郑州 450001)

摘要: 在联合匹配边缘概率和条件概率分布以减小源域与目标域的差异性时,存在由类不平衡导致模型泛化性能差的问题,从而提出了基于平衡概率分布和实例的迁移学习算法。通过基于核的主成分分析方法将特征数据映射到低维子空间,在子空间中对源域与目标域的边缘分布和条件分布进行联合适配,利用平衡因子动态调节每个分布的重要性,采用加权条件概率分布自适应地改变每个类的权重,同时融合实例更新策略,进一步提升模型的泛化性能。在字符和对象识别数据集上进行了多组对比实验,表明该算法有效地提高了图像分类的准确率。

关键词: 迁移学习; 平衡分布; 类不平衡; 实例更新; 领域自适应

中图分类号: TP3

文献标志码: A

文章编号: 1671-6841(2020)03-0055-07

DOI: 10.13705/j.issn.1671-6841.2019439

0 引言

我们正处在一个飞速发展的大数据时代,每天各行各业都产生海量的图像数据。数据规模的不断增大,使得机器学习的模型能够持续不断地进行训练和更新,从而提升模型的性能。传统的机器学习和图像处理中,通常假设训练集和测试数据集遵循相同的分布,而在实际视觉应用中相同分布假设很难成立,诸如姿势、光照、模糊和分辨率等许多因素都会导致特征分布发生改变,而重新标注数据工作量较大,且成本较高,也就形成了大量的不同分布的训练数据,如果弃之不用则会造成浪费。如何充分有效地利用这些不同分布的训练数据,成为计算机视觉研究中的一个具有挑战性的问题。而迁移学习是针对此类问题的一种有效解决方法,能够将知识从标记的源域转移到目标域,用来自旧域的标记图像来学习用于新域的精确分类器。

目前,迁移学习已经成为人工智能领域的一个研究热点。其基本方法可以归纳为 4 类^[1],即基于特征、基于样本、基于模型及基于关系的迁移。其中基于特征的迁移学习方法是指通过特征变换的方法,来尽可能地缩小源域与目标域之间的分布差异,实现知识跨域的迁移^[2-8]。文献[2]提出迁移主成分分析(transfer component analysis, TCA),通过特征映射得到新的特征表示,以最大均值差异(maximum mean discrepancy, MMD)作为度量准则,将领域间的边缘分布差异最小化。由于 TCA 仅对域间边缘分布进行适配,故而有较大的应用局限性。文献[3]提出的联合分布自适应(joint distribution adaptation, JDA)在 TCA 的基础上增加对源域和目标域的条件概率进行适配,联合选择特征和保留结构性质,将域间差异进一步缩小。基于样本的迁移方法通常对样本实例进行加权^[9-10],以此来削弱源域中与目标任务无关的样本的影响,不足之处是容易推导泛化误差上界,应用的局限性较大。基于模型的迁移方法则是利用不同域之间能够共享的参数信息,来实现源域到目标域的迁移。而基于关系的迁移学习方法关注的是不同域的样本实例之间的关系,目前相关方面的研究较少。

本文提出的基于平衡概率分布和实例的迁移学习算法(balanced distribution adaptation and instance based transfer learning algorithm, BDAITL)是一种混合算法,结合了上述的基于特征和样本实例这两种基本的迁移算法。在多个真实数据集上进行的多组相关实验表明, BDAITL 算法模型泛化性能良好。

收稿日期:2019-09-25

基金项目:河南省高校科技创新团队支持计划项目(17IRTSTHN013)。

作者简介:黄露(1994—),女,河南驻马店人,硕士研究生,主要从事智能控制理论、机器学习研究, E-mail:1751037268@qq.com;通信作者:曾庆山(1963—),男,湖北武汉人,教授,主要从事智能控制理论、复杂系统的建模研究, E-mail:huanglulu823@163.com。

1 问题描述

迁移学习就是把源域中学习到的知识迁移到目标域,帮助目标域进行模型训练。领域和任务是迁移学习的两个基本概念。下面从领域和任务的定义方面,对要解决的问题进行描述^[1]。

定义 1 领域 D 是迁移学习中进行学习的主体,由特征空间 \mathcal{X} 和边缘概率分布 $P(\mathbf{X})$ 组成,可以表示为 $D = \{\mathcal{X}, P(\mathbf{X})\}$, 其中:特征矩阵 $\mathbf{X} = \{x_1, x_2, \dots, x_n\} \in \mathcal{X}$ 。领域与领域之间的不同一般有两种情况,特征空间不同或边缘概率分布不同。

定义 2 给定一个领域 D , 任务 T 定义为由类别空间 Y 和一个预测函数 $f(x)$ 构成,表示为 $T = \{Y, f(x)\}$, 其中类别标签 $y \in Y$ 。

问题 1 给定一个有完整标注的源领域 $D_s = \{x_i, y_i\}_{i=1}^{n_s}$ 和源任务 T_s 。一个没有任何标注的目标领域 $D_t = \{x_j\}_{j=1}^{n_t}$ 和目标任务 T_t 。假设 D_s 和 D_t 有相同的特征空间和类别空间:即 $\mathcal{X}_s = \mathcal{X}_t, Y_s = Y_t$; 以及不同的分布:即边缘概率分布 $P(\mathbf{X}_s) \neq P(\mathbf{X}_t)$ 、条件概率分布 $P(y_s/x_s) \neq P(y_t/x_t)$ 。迁移学习最终的目标是,迁移 D_s 和 T_s 中的知识以帮助 D_t 和 T_t 训练预测函数 $f(x)$, 提升模型的性能。

2 基于平衡概率分布和实例的迁移学习算法

BDAITL 算法从特征和样本实例两个层面进行知识的迁移。首先,使用基于核的主成分分析法(Kernel principal component analysis, KPCA),采用非线性映射将源域与目标域的高维数据映射到一个低维子特征空间。然后,在子空间内采用 MMD 方法联合匹配域间的边缘分布和条件分布。与 JDA 直接忽略两者之间重要性不同的是,BDAITL 算法采用平衡因子来评估每个分布的重要性^[4]。另外,JDA 在适配条件分布时,由于目标域无标签,无法直接建模,采用了类条件概率来近似、隐含地假设每个域中该类的概率是相似的,而实际应用中通常是不成立的。而 BDAITL 算法在适配条件分布时,充分考虑类不平衡问题,采用加权来平衡每个域类别比例,得出了更为稳健的近似。最后,考虑源域中并不是所有的样本实例都与目标任务的训练有关,采用 $L_{2,1}$ 范数将行稀疏性引入变换矩阵 \mathbf{A} , 选择源域中相关性高的实例进行目标任务模型的训练。BDAITL 算法的具体过程在下文介绍。

2.1 问题建模

首先,针对源域和目标域特征维数过高的问题,对其进行降维重构,最大限度地最小化领域间的分布差异,从而利于判别信息从源域到目标域的迁移。记 $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t] = [x_1, x_2, \dots, x_n] \in \mathbf{R}^{m \times n}$ 表示源域和目标域的所有样本组成的矩阵,中心矩阵表示为 $\mathbf{H} = \mathbf{I} - (1/n)\mathbf{1}$, 其中: m 表示样本维数; $n = n_s + n_t$ 表示样本总数; $\mathbf{1} \in \mathbf{R}^{n \times n}$ 表示元素全为 1 的矩阵。PCA 的优化目标是找到正交变换 $\mathbf{V} \in \mathbf{R}^{m \times q}$, 使样本的协方差矩阵 $\mathbf{X}\mathbf{H}\mathbf{X}^T$ 最大化,即

$$\max \text{tr}(\mathbf{V}^T \mathbf{X}\mathbf{H}\mathbf{X}^T \mathbf{V}), \text{ s. t. } \mathbf{V}^T \mathbf{V} = \mathbf{I}, \quad (1)$$

其中: q 为降维后特征子空间基向量的个数;新的特征表示为 $\mathbf{Z} = \mathbf{V}^T \mathbf{X}$ 。

本文使用 KPCA 方法对源域和目标域数据降维。利用 KPCA 方法,应用核映射 $\mathbf{X} \rightarrow \Psi(\mathbf{X})$ 对 PCA 进行非线性推广,获取数据的非线性特征,相应的核矩阵为 $\mathbf{K} = \Psi(\mathbf{t})^T \Psi(\mathbf{t}) \in \mathbf{R}^{n \times n}$, 对式(1)进行核化后可得

$$\max \text{tr}(\mathbf{A}^T \mathbf{K}\mathbf{H}\mathbf{K}^T \mathbf{A}), \text{ s. t. } \mathbf{A}^T \mathbf{A} = \mathbf{I}, \quad (2)$$

其中: $\mathbf{A} \in \mathbf{R}^{n \times q}$ 是变换矩阵;核化后的特征表示为 $\mathbf{Z} = \mathbf{A}^T \mathbf{K}$ 。

其次,平衡概率分布。迁移学习需要解决的一个主要问题是减小源域与目标域之间的分布差异,包括边缘分布和条件分布,将不同的数据分布的距离拉近。本文采用 MMD 方法来最小化源域与目标域之间的边缘分布 $P(\mathbf{X}_s)$ 、 $P(\mathbf{X}_t)$ 以及条件分布 $P(y_s/x_s)$ 、 $P(y_t/x_t)$ 的距离。即

$$\begin{aligned} D(D_s, D_t) &= (1 - \mu) \|P(\mathbf{X}_s) - P(\mathbf{X}_t)\| + \mu \|P(y_s/x_s) - P(y_t/x_t)\| = \\ &= (1 - \mu) \text{MMD}_H^2(P(\mathbf{X}_s), P(\mathbf{X}_t)) + \mu \text{MMD}_H^2(P(y_s/x_s), P(y_t/x_t)), \end{aligned} \quad (3)$$

其中: $\mu \in [0, 1]$ 是平衡因子。当 $\mu \rightarrow 0$ 时,表示源域和目标域数据本身存在较大的差异性,边缘分布更重

要;当 $\mu = 0$ 时,即为TCA;当 $\mu \rightarrow 1$ 时,表示域间数据集有较高的相似性,条件分布适配更为重要;当 $\mu = 0.5$ 时,即为JDA。也就是说,平衡因子根据实际数据分布的情况,来动态调节每个分布的重要性。源域与目标域边缘概率分布的MMD距离计算如下, M_o 是MMD矩阵,

$$MMD_H^2(P(X_s), P(X_t)) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} A^T k_i - \frac{1}{n_t} \sum_{j=n_s+1}^{n_s+n_t} A^T k_j \right\|_H^2 = \text{tr}(A^T K M_o K^T A), \quad (4)$$

$$M_o(i, j) = \begin{cases} 1/(n_s)^2, & x_i \in D_s, x_j \in D_s, \\ 1/(n_t)^2, & x_i \in D_t, x_j \in D_t, \\ -1/n_s n_t, & \text{其他。} \end{cases} \quad (5)$$

适配源域与目标域的条件概率分布时,采用加权来平衡每个域类别比例。具体为

$$\left\| P\left(\frac{y_s}{x_s}\right) - P\left(\frac{y_t}{x_t}\right) \right\|_H^2 = \left\| \frac{P(y_s)}{P(x_s)} P\left(\frac{x_s}{y_s}\right) - \frac{P(y_t)}{P(x_t)} P\left(\frac{x_t}{y_t}\right) \right\|_H^2 = \left\| \alpha_s P\left(\frac{x_s}{y_s}\right) - \alpha_t P\left(\frac{x_t}{y_t}\right) \right\|_H^2, \quad (6)$$

其中: α_s, α_t 表示权值。故源域与目标域条件概率分布的MMD距离计算为

$$MMD_H^2(P\left(\frac{y_s}{x_s}\right), P\left(\frac{y_t}{x_t}\right)) = \sum_{c=1}^C \left\| \frac{\alpha_s^{(c)}}{n_s^{(c)}} \sum_{x_i \in D_s^{(c)}} A^T k_i - \frac{\alpha_t^{(c)}}{n_t^{(c)}} \sum_{x_j \in D_t^{(c)}} A^T k_j \right\|_H^2 = \sum_{c=1}^C \text{tr}(A^T K M_c K^T A), \quad (7)$$

其中: $c \in (1, 2, \dots, C)$ 表示样本类别; $D_s^{(c)}, D_t^{(c)}$ 和 $n_s^{(c)}, n_t^{(c)}$ 分别表示源域和目标域中属于类别 c 的样本集合和样本数; M_c 为每一类别的加权MMD矩阵,

$$M_c(i, j) = \begin{cases} P(y_s^{(c)})/n_s^{(c)} n_s^{(c)}, & x_i \in D_s^{(c)}, x_j \in D_s^{(c)}, \\ P(y_t^{(c)})/n_t^{(c)} n_t^{(c)}, & x_i \in D_t^{(c)}, x_j \in D_t^{(c)}, \\ -\sqrt{P(y_s^{(c)})P(y_t^{(c)})}/n_s^{(c)} n_t^{(c)}, & x_i \in D_s^{(c)}, x_j \in D_t^{(c)} \text{ 或 } x_i \in D_t^{(c)}, x_j \in D_s^{(c)}, \\ 0, & \text{其他。} \end{cases} \quad (8)$$

综合式(2)、式(3)、式(7)和式(8),可得源域和目标域的平衡概率分布

$$D(D_s, D_t) = (1 - \mu) \text{tr}(A^T K M_o K^T A) + \mu \sum_{c=1}^C \text{tr}(A^T K M_c K^T A) = (1 - \mu) \text{tr}(A^T K M_o K^T A) + \mu \text{tr}(A^T K W_c K^T A), \quad (9)$$

其中: $W_c = \sum_{c=1}^C M_c$ 。

最后,实例更新。源域中通常会存在一些特殊的样本实例,对于训练目标域的分类模型是没有用的。由于变换矩阵 A 的每一行都对应一个实例,基于它们与目标实例的相关性,行稀疏性基本上可以促进实例的自适应加权,实现更新学习。故本文对变换矩阵中与源域相关的部分 A_s 引入 $L_{2,1}$ 范数约束,同时对与目标域相关的部分 A_t 施加 F 范数约束,以保证模型是良好定义的。即

$$\|A_s\|_{2,1} + \|A_t\|_F^2. \quad (10)$$

通过最小化式(10)使得式(2)最大化,与目标实例相关(不相关)的源域实例被自适应地重新加权,在新的特征表示 $Z = A^T K$ 中具有更大(更少)的重要性。

综上所述,可得本文的最终优化目标

$$\min (1 - \mu) \text{tr}(A^T K M_o K^T A) + \mu \text{tr}(A^T K W_c K^T A) + \lambda (\|A_s\|_{2,1} + \|A_t\|_F^2) \quad \text{s.t. } A^T K H K^T A = I, \quad (11)$$

其中: λ 是权衡特征匹配和实例重新加权的正则化参数,能够控制模型复杂度并保证模型正定。

2.2 目标优化

式(11)所示目标函数是一个带有约束的最优化问题,利用Lagrange法进行求解,记

$$L = (1 - \mu) \text{tr}(A^T K M_o K^T A) + \mu \text{tr}(A^T K W_c K^T A) + \lambda (\|A_s\|_{2,1} + \|A_t\|_F^2) - \text{tr}((I - A^T K H K^T A) \Phi)$$

为式(11)的Lagrange函数, Φ 为Lagrange乘子。 $\partial L / \partial A = (K((1 - \mu)M_o + \mu W_c)K^T + \lambda G)A - K H K^T A \Phi$,令 $\partial L / \partial A = 0$ 可得, $(K((1 - \mu)M_o + \mu W_c)K^T + \lambda G)A = K H K^T A \Phi$ 。由于在零点并不是平滑的,故子梯度的计算为 $\partial(\|A_s\|_{2,1} + \|A_t\|_F^2) / \partial A = 2GA$,其中: G 是一个子梯度矩阵,且

$$G_{ii} = \begin{cases} 1/(2 \| \mathbf{a}^i \|), & x_i \in D_s, \mathbf{a}^i \neq \mathbf{0}, \\ 0, & x_i \in D_s, \mathbf{a}^i = \mathbf{0}, \\ 1, & x_i \in D_t, \end{cases}$$

其中: \mathbf{a}^i 是矩阵 \mathbf{A} 的第 i 行。这样将求解变换矩阵 \mathbf{A} 归结为求解特征分解, 得到 q 个最小的特征向量。

3 实验结果及分析

3.1 实验数据集

为了研究和测试算法的性能, 在不同的数据集上进行测试实验。USPS 和 MNIST 是包含 0~9 的手写数字的标准数字识别数据集, 分别包含训练图像 60 000 幅和 7 291 幅以及测试图像 10 000 幅和 2 007 幅, 示例如图 1 所示。office 由 3 个对象域组成: amazon(在线电商图像)、webcam(网络摄像头拍摄的低解析度图像)、DSLR(单反相机拍摄的高清晰度图像), 共有 4 652 幅图像, 31 个类别。caltech-256 是对象识别的基准数据集, 共有 30 607 幅图像, 256 个类别, 示例如图 2 所示。

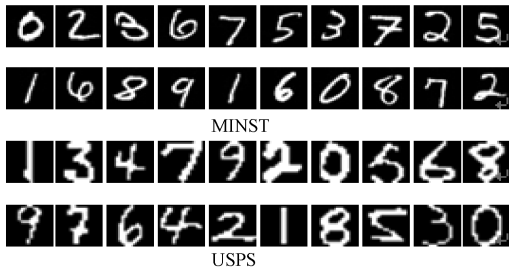


图 1 MNIST 和 USPS 数据集图片示例

Figure 1 Example of MNIST and USPS dataset

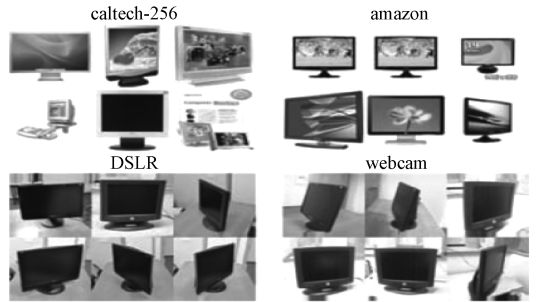


图 2 office 和 caltech-256 数据集图片示例

Figure 2 Example of office and caltech-256 dataset

本文实验采用文献[5]中的方法预处理数据集 MNIST 和 USPS, 以及文献[6]中方法的预处理数据集 office 和 caltech-256。其统计信息如表 1 所示, 数据子集 M 和 U 分别作为源域和目标域, 可构建 $M \rightarrow U$ 、 $U \rightarrow M$ 两个跨域迁移学习任务。数据子集 A 、 W 、 D 和 C 中任意两个作为源域和目标域, 可构建 12 个跨域迁移学习任务, 记为: $D \rightarrow W$ 、 $D \rightarrow C$ 、 \dots 、 $A \rightarrow C$ 。

表 1 实验数据子集的统计信息

Table 1 Dataset used in the experiment

数据集	样本数	特征维数	类别数	子集
MNIST	2 000	256	10	M
USPS	1 800	256	10	U
office+caltech-256	2 533	800	10	A, W, D, C

3.2 实验结果分析

实验验证环节, 将 BDAITL 方法与用于图像分类问题的 6 种相关方法进行了比较, 即最近邻算法(nearest neighbor, NN)、主成分分析法(principal component analysis, PCA)、TCA、基于核的测地流形法(geodesic flow kernel, GFK)、JDA 以及转移联合匹配方法(transfer joint matching, TJM)。评价准则是目标域中的样本分类准确率(accuracy), 具体计算为

$$accuracy = |x: x \in D_t \cap \hat{y}(x) = y(x)| / |x: x \in D_t|,$$

其中: x 表示目标域中的测试样本; $y(x)$ 表示其真实标签; $\hat{y}(x)$ 表示其预测标签。实验结果如表 2 所示, 分别设置 $q = 40$ 、 $\lambda = 1$ 、迭代次数 $t = 10$ 。

如表 2 所示, BDAITL 算法的分类准确率相较于传统方法 NN 和 PCA 有明显的提升。与经典迁移学习算法 TCA、GFK、JDA、TJM 相比, BDAITL 算法的分类准确率在大部分的跨域学习任务中有较大幅度的提高,

表 2 7 种算法在 14 个迁移任务中的平均准确率

Table 2 Accuracy comparison of 7 algorithms on 14 cross-domain tasks

数据集	平均准确率/%						
	NN	PCA	TCA	GFK	JDA	TJM	BDAITL
$D \rightarrow W$	63.39	75.93	86.44	75.59	89.49	85.42	91.19
$D \rightarrow C$	26.27	29.65	32.50	30.28	30.99	31.43	33.57
$D \rightarrow A$	28.50	32.05	31.52	32.05	32.25	32.61	34.76
$W \rightarrow D$	25.84	77.07	85.99	80.89	89.17	89.73	92.99
$W \rightarrow C$	19.86	26.36	27.16	30.72	31.17	30.19	33.93
$W \rightarrow A$	22.96	31.00	30.69	29.75	32.78	29.26	33.51
$C \rightarrow W$	25.76	32.54	36.61	40.68	38.64	37.97	44.41
$C \rightarrow D$	25.48	38.22	45.86	38.85	45.22	43.31	46.50
$C \rightarrow A$	23.70	36.95	44.89	41.02	42.90	46.66	46.87
$A \rightarrow W$	29.83	35.59	37.63	38.98	37.97	40.74	43.05
$A \rightarrow D$	25.48	27.39	31.85	36.31	39.49	42.17	46.50
$A \rightarrow C$	26.00	34.73	40.78	40.25	38.36	39.45	41.23
$M \rightarrow U$	65.94	66.22	54.28	67.22	62.89	62.37	76.00
$U \rightarrow M$	44.70	44.95	52.20	46.45	57.45	52.25	63.05

其中在任务 $M \rightarrow U$ 中较其最佳基准算法(GFK)提高了 8.78%,这表明 BDAITL 算法在适配条件概率时采用加权来平衡每个域的分类比例对算法的性能提升是有效的,是平衡域之间不同类别分布的有效方法。同时实例的更新学习也能够削弱一些不相关实例的影响,一定程度上提升了算法的性能。

3.3 参数分析

在本文的 BDAITL 算法的优化模型中,设置了 3 个参数,即平衡因子 μ 、正则化参数 λ 以及子空间纬度 q 。实验中通过保持其中两个参数不变,改变第 3 个参数的值来观察其对算法性能的影响。

平衡因子 μ 可以通过分别计算两个领域数据的整体和局部的分布距离来近似给出。为了分析 μ 在不同的取值下对 BDAITL 算法性能的影响,取 $\mu \in \{0, 0.1, 0.2, \dots, 0.9\}$,实验结果如表 3 所示。从表中可以看出,不同的学习任务对于 μ 的取值敏感度不完全相同,如 $D \rightarrow W$ 、 $W \rightarrow D$ 、 $C \rightarrow D$ 、 $M \rightarrow U$ 、 $U \rightarrow M$ 分别在 0.6、0.4、0.6、0.2、0.3 时取得最大的分类准确率, μ 值越大说明适配条件概率分布越重要。它表明在不同的跨领域学习问题中,边缘分布自适应和条件分布自适应并不是同等重要的,而 μ 起到了很好的平衡作用。

表 3 μ 的取值对 BDAITL 算法准确率的影响

Table 3 Influence of μ on the accuracy of the BDAITL algorithm

数据集	准确率/%									
	$\mu=0$	$\mu=0.1$	$\mu=0.2$	$\mu=0.3$	$\mu=0.4$	$\mu=0.5$	$\mu=0.6$	$\mu=0.7$	$\mu=0.8$	$\mu=0.9$
$D \rightarrow W$	89.15	90.85	90.51	90.51	90.17	90.51	91.19	90.85	90.85	90.53
$D \rightarrow C$	32.32	32.28	32.24	32.68	32.77	32.59	33.57	32.86	32.68	32.41
$D \rightarrow A$	32.99	33.61	34.34	34.66	34.66	34.76	34.24	33.61	33.51	33.19
$W \rightarrow D$	89.81	90.45	91.08	92.36	92.99	92.36	91.72	91.08	90.45	89.17
$W \rightarrow C$	34.73	34.64	34.28	33.93	33.75	33.57	33.21	33.13	33.48	33.84
$W \rightarrow A$	31.52	32.05	32.25	31.94	32.57	32.46	32.78	33.09	33.09	33.51
$C \rightarrow W$	39.32	38.64	38.98	40.34	42.03	42.37	43.73	44.41	43.39	43.05
$C \rightarrow D$	42.68	42.68	43.31	43.31	43.95	43.95	46.50	43.95	43.95	43.31
$C \rightarrow A$	45.82	45.82	45.72	45.93	46.35	46.45	46.66	46.56	46.87	46.87
$A \rightarrow W$	41.69	41.36	41.36	42.03	42.71	43.05	42.37	41.02	40.68	40.00
$A \rightarrow D$	46.50	45.86	44.59	43.95	43.95	44.59	46.50	45.86	45.22	44.59
$A \rightarrow C$	41.14	41.23	40.69	41.05	41.05	40.87	40.78	40.34	40.52	40.61
$M \rightarrow U$	62.17	74.61	76.00	75.28	74.72	73.89	73.44	72.56	73.11	73.28
$U \rightarrow M$	49.95	61.50	62.70	63.05	61.80	61.65	61.85	61.60	61.45	61.20

表 4 是 q 分别取 20、40、60、80、100、140、180、220、260、300 时,BDAITL 算法的分类准确率的变化情况。从表中可以看出,不同的迁移学习任务在达到最优性能时,所对应的 q 是不同的,即不同任务的最优子空间纬度是不同的,如 $D \rightarrow W$ 、 $W \rightarrow D$ 、 $C \rightarrow D$ 、 $M \rightarrow U$ 、 $U \rightarrow M$ 的最优子空间纬度分别是 80、100、80、60、60。

表4 q 的取值对 BDAITL 算法准确率的影响Table 4 Influence of q on the accuracy of the BDAITL algorithm

数据集	准确率/%									
	$q=20$	$q=40$	$q=60$	$q=80$	$q=100$	$q=140$	$q=180$	$q=220$	$q=260$	$q=300$
$D \rightarrow W$	89.15	91.19	92.54	92.88	92.20	90.17	89.83	89.49	89.15	89.15
$D \rightarrow C$	33.04	33.57	33.21	33.66	33.39	32.32	32.86	32.50	32.06	31.97
$D \rightarrow A$	35.18	34.24	32.46	33.40	32.15	32.78	32.57	32.46	32.25	32.05
$W \rightarrow D$	89.17	91.72	91.08	89.17	92.36	91.72	91.08	90.45	88.54	87.26
$W \rightarrow C$	32.95	33.21	32.59	32.50	33.30	32.50	33.21	32.77	32.06	31.52
$W \rightarrow A$	33.09	32.78	33.51	32.99	34.13	33.09	32.46	33.92	34.34	33.92
$C \rightarrow W$	41.69	42.37	40.68	40.34	39.32	39.66	39.66	40.34	40.00	39.66
$C \rightarrow D$	47.77	46.50	47.13	48.41	45.86	45.22	47.13	45.22	45.22	44.59
$C \rightarrow A$	45.51	46.66	45.82	47.18	47.60	46.35	45.62	44.57	44.47	43.95
$A \rightarrow W$	46.44	42.71	39.66	38.98	39.32	37.29	36.95	35.59	36.27	35.25
$A \rightarrow D$	42.68	46.50	36.31	33.76	32.48	35.03	35.67	37.58	38.22	36.94
$A \rightarrow C$	41.14	40.78	40.69	39.54	39.36	39.18	39.27	39.08	39.00	38.82
$M \rightarrow U$	73.22	73.44	75.44	75.06	74.94	75.06	75.11	75.00	75.17	75.22
$U \rightarrow M$	59.85	61.85	62.15	61.95	61.70	61.65	61.90	61.80	61.25	61.50

正则化参数 λ 取值为 $\lambda \in \{0.001, 0.01, \dots, 100\}$ 时,对 BDAITL 算法性能的影响如表 5 所示。可以看出,由于不同的迁移任务中源域与目标域的样本实例相差较大,导致不同的迁移学习任务在 λ 的不同取值下得到最优分类性能,其中部分任务如 $D \rightarrow W$ 、 $W \rightarrow D$ 、 $C \rightarrow D$ 、 $M \rightarrow U$ 、 $U \rightarrow M$ 分别是在 0.1、10、0.1、1、1 时取得最优性能。

表5 λ 的取值对 BDAITL 算法准确率的影响Table 5 Influence of λ on the accuracy of the BDAITL algorithm

数据集	准确率/%						
	$\lambda=0.001$	$\lambda=0.01$	$\lambda=0.1$	$\lambda=1$	$\lambda=10$	$\lambda=100$	
$D \rightarrow W$	84.41	87.80	92.88	92.54	90.17	90.51	
$D \rightarrow C$	31.61	31.97	33.30	33.21	32.15	31.52	
$D \rightarrow A$	36.12	33.92	31.42	32.46	31.84	31.94	
$W \rightarrow D$	82.17	85.35	88.54	91.08	92.36	89.81	
$W \rightarrow C$	30.28	30.99	29.39	32.59	32.41	31.88	
$W \rightarrow A$	33.09	32.88	31.21	33.51	30.17	29.54	
$C \rightarrow W$	30.85	34.58	37.29	40.68	41.02	40.34	
$C \rightarrow D$	40.76	43.95	47.77	47.13	44.59	42.04	
$C \rightarrow A$	42.17	44.78	48.33	45.82	46.35	46.03	
$A \rightarrow W$	34.92	36.27	37.29	39.66	40.68	40.68	
$A \rightarrow D$	31.85	36.31	40.76	46.50	43.95	43.31	
$A \rightarrow C$	39.18	40.87	41.94	40.78	39.72	39.54	
$M \rightarrow U$	72.17	71.94	73.06	75.44	74.33	67.44	
$U \rightarrow M$	60.00	59.80	61.15	62.15	58.25	52.75	

4 总结

本文提出基于平衡概率分布和实例的迁移学习算法,融合了特征选择和实例更新两种策略。它采用平衡因子来自适应地调节边缘和条件分布适应的重要性,使用加权条件分布来处理域间的类不平衡问题,然后融合实例更新策略,进一步提升算法的性能。在 4 个图像数据集上的大量实验证明了该方法优于其他几种方法。但参数优化方面仍有改进的空间,在下一步的研究中将着重探索多参数优化方法,以期进一步提高算

法的性能。未来将继续探索迁移学习中针对类不平衡问题的处理方法,在传递式迁移学习和多源域迁移学习方向进行深入研究。

参考文献:

- [1] PAN S J, YANG Q. A survey on transfer learning[J]. IEEE transactions on knowledge and data engineering, 2010, 22(10): 1345–1359.
- [2] PAN S J, TSANG I W, KWOK J T, et al. Domain adaptation via transfer component analysis[J]. IEEE transactions on neural networks, 2011, 22(2): 199–210.
- [3] LONG M S, WANG J M, DING G G, et al. Transfer feature learning with joint distribution adaptation[C]//IEEE International Conference on Computer Vision. Sydney, 2013: 2200–2207.
- [4] WANG J D, CHEN Y Q, HAO S J, et al. Balanced distribution adaptation for transfer learning[C]//IEEE International Conference on Data Mining. New Orleans, 2017: 1129–1134.
- [5] LONG M S, WANG J M, DING G G, et al. Transfer joint matching for unsupervised domain adaptation[C]//IEEE Conference on Computer Vision and Pattern Recognition. Columbus, 2014: 1410–1417.
- [6] GONG B Q, SHI Y, SHA F, et al. Geodesic flow kernel for unsupervised domain adaptation[C]//IEEE Conference on Computer Vision and Pattern Recognition. Providence, 2012:2066–2073.
- [7] TAHMORESNEZHAD J, HASHEMI S. Visual domain adaptation via transfer feature learning[J]. Knowledge and information systems, 2017, 50(2): 585–605.
- [8] ZHANG J, LI W Q, OGUNBONA P. Joint geometrical and statistical alignment for visual domain adaptation[C]//IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017:5150–5158.
- [9] 赵鹏, 吴国琴, 刘慧婷, 等. 基于特征联合概率分布和实例的迁移学习算法[J]. 模式识别与人工智能, 2016, 29(8): 717–724.
ZHAO P, WU G Q, LIU H T, et al. Feature joint probability distribution and instance based transfer learning algorithm[J]. Pattern recognition and artificial intelligence, 2016, 29(8): 717–724.
- [10] 戴文渊. 基于实例和特征的迁移学习算法研究[D]. 上海: 上海交通大学, 2009: 8–23.
DAI W Y. Instance-based and feature-based transfer learning[D]. Shanghai: Shanghai Jiaotong University, 2009: 8–23.

Balanced Distribution Adaptation and Instance Based Transfer Learning Algorithm

HUANG Lu, ZENG Qingshan

(College of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: Aim to deal with the poor generalization ability caused by class imbalance of jointly matching the marginal probability and conditional probability distribution to reduce the domain difference, a balanced distribution adaptation and instance based transfer learning algorithm was proposed. The feature instances were mapped to the subspace with the kernel principal component analysis. In this subspace, the marginal and conditional probability distribution were jointly matched with dynamically adjusting the different importance of each distribution by a balance factor and adaptively changing the weight of each class. Thus, the difference between the source domain and target domain was reduced. Meanwhile, the instance update strategy was merged and the generalization ability of the model obtained by transfer learning was improved further. Experimental results on the digital and object recognition datasets demonstrated the validity and efficiency of the proposed algorithm.

Key words: transfer learning; balance distribution; class imbalance; instance update; domain adaptation

(责任编辑:王浩毅)