文章编号: 1001-0920(2017)11-1941-08 DOI: 10.13195/j.kzyjc.2016.1279

基于相似度学习的多源迁移算法

卞则康†, 王士同

(江南大学 数字媒体学院，江苏 无锡 214122)

摘 要: 针对与测试数据分布相同的训练数据不足,相关领域中存在大量的、与测试数据分布相近的训练数据的场景,提出一种基于相似度学习的多源迁移学习算法(SL-MSTL). 该算法在经典SVM分类模型的基础上提出一种新的迁移分类模型,增加对多源域与目标域之间的相似度学习,可以有效地利用各源域中的有用信息,提高目标域的分类效果. 实验的结果表明了SL-MSTL 算法的有效性和实用性.

关键词: 相似度学习；多源域；迁移学习；SVM；迁移分类

中图分类号: TP181 **文献标志码:** A

Similarity-learning based multi-source transfer learning algorithm

BIAN Ze-kang†, WANG Shi-tong

(School of Digital Media，Jiangnan University，Wuxi 214122，China)

Abstract: For the proplem that the training data which have the same distribution with the test data are insuficient, but a lot of training data which have the similar distribution with the test data exist in the related field, a similarity-learning based multi-source transfer learning(SL-MSTL) algorithm is proposed. A similarity-learning based classification model is proposed in contrast to the classical support vector machine(SVM) model. Compared to the SVM model, the proposed similarity-learning based model can make better use of the source information and improve the classification performance. Experimental results show the effectiveness and the practicality of the proposed algorithm.

Keywords: similarity learning；multi-source；transfer learning；SVM；transfer classification

0 引 言

相似度学习是一种常见的机器学习方法,常常被用于人工智能领域的聚类、分类、回归等数据挖掘问题中[1-3]. 相似度学习的目标是从已知的样本中学习出一个相似度函数,这个相似度函数显示了两个目标之间的相似度. 因此,如何有效地学习出目标间的相似度,并将之运用到实际应用中已经成为一个重要的研究方向.

多源迁移分类是在迁移学习的基础上提出的一种适用性更加广泛的学习框架,它强调源域之间、源域与目标域之间存在着一定的相似度和差异度,从而通过对多源域进行学习来提高目标域的分类效果[4]. 常见的迁移分类算法分为以下两类:一类是目标域含有少量的数据标签已知的数据,例如基于流行结构的MMDE算法[5-6]、多源多视角Adaboost迁移学习算法[7-8]、基于香农熵的一致性[9];另一类是目标域数据的标签完全未知,借助分类器或者源域数据进行迁移学习,如ML型模糊迁移系统[10]、基于LSSVM光滑假设[11]、基于异构一致性学习框架[12].

在大多数多源迁移学习中,一个重要的前提是源域与目标域之间存在一定的相似度,因此如何确定相似度和利用相似度进行迁移分类是一个重要的问题. 在迁移分类的框架下,利用源域与目标域的相似度学习,不仅可以提高算法的学习效率,而且可以提高目标域的分类效果.

针对上述问题,本文提出一种基于源域与目标域之间的相似度学习的多源迁移算法SL-MSTL (similarity-learning based multi-source transfer learning). SL-MSTL算法利用各源域的相似度对目标域进行分类,利用各个源域的分类器,在最大边界的框架下进行相似度学习,得到各源域与目标域的相似度,最后构建适合于目标域的新的分类模型. 将学习得到的新的分类模型应用到目标域中,以提高分类的效果和性能,并通过实验验证了这一观点.

收稿日期: 2016-10-11；**修回日期:** 2017-02-13.

基金项目: 国家自然科学基金项目(61170122, 61272210)；江苏省自然科学基金项目(BK20130155).

作者简介: 卞则康(1993−), 男, 博士生, 从事人工智能与模式识别、机器学习的研究；王士同(1964−), 男, 教授, 博士生导师, 从事人工智能与模式识别、机器学习、深度学习等研究.

†通讯作者. E-mail: bianzekang@163.com

1 相关工作

1.1 经典的SVM算法

支持向量机(SVM)[13]是一种常用的二分类模型,其基本模型定义为特征空间上间隔最大的线性分类器,其学习策略就是间隔的最大化,最终可以转化成一个凸二次规划问题的求解. 标准的SVM算法通过对已知标签的数据集$\boldsymbol{D}_l^t$进行训练得到相应的分类模型,之后利用得到的训练模型对未知标签的数据集$\boldsymbol{D}_u^t$进行分类. 对于给出的一个线性数据样本$\boldsymbol{x}$,SVM通过一个线性决策函数$f(\boldsymbol{x})=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}+b$判断其类别. 对于非线性的数据,SVM通过定义一个内核映射ϕ将低维的$\boldsymbol{x}$映射到高维空间$\phi(\boldsymbol{x})$,则相应的决策函数为$f(\boldsymbol{x})=\boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{x})+b$. SVM的最终目标是为了寻找一个最佳的超平面,使得区分数据的正负类时的泛化误差较低. 该超平面由优化以下目标函数所确定:

$$\begin{aligned}&\min_{\boldsymbol{W}}\frac{1}{2}\|\boldsymbol{W}\|_2^2+C\sum_{i=1}^{N_l^t}\varepsilon_i;\\&\text{s.t. }y_i\boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b\geqslant 1-\varepsilon_i,\ \varepsilon_i\geqslant 0,\\&\qquad\forall(\boldsymbol{x}_i,y_i)\in\boldsymbol{D}_l^t.\end{aligned}\tag{1}$$

其中:ε_i为每一个$\boldsymbol{x}_i$增加的惩罚项,C为惩罚因子.

在跨源域迁移学习中,重要的前提是源域$\boldsymbol{D}^s$和目标域$\boldsymbol{D}^t$之间有一定的相似度,目标域中分为标签已知的$\boldsymbol{D}_l^t$和标签未知的$\boldsymbol{D}_u^t$. 为了将经典的SVM算法应用到迁移学习中,人们提出了基于SVM算法的改进算法.

1.2 A-SVM算法

文献[14]提出了一种改进的SVM算法Adaptive SVM(A-SVM). A-SVM算法的最终决策函数可以表示为

$$f(\boldsymbol{x})=f^s(\boldsymbol{x})+\Delta f(\boldsymbol{x}),\tag{2}$$

$$\Delta f(\boldsymbol{x})=\boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{x})+b.\tag{3}$$

其中:$f^s(\boldsymbol{x})$为源域$\boldsymbol{D}^s$的分类器,$\Delta f(\boldsymbol{x})$为通过对已知标签的目标域数据$\boldsymbol{D}_l^t$得到的分类器. A-SVM算法的主要目的是通过对源域数据$\boldsymbol{D}^s$和目标域已知标签的数据$\boldsymbol{D}_l^t$进行学习得到一个新的决策边界,这个新的决策边界在最接近源域$\boldsymbol{D}^s$的情况下,尽可能区分目标域$\boldsymbol{D}_l^t$中数据. 决策边界定义如下:

$$\begin{aligned}&\min_{\tilde{\boldsymbol{W}}}\frac{1}{2}\|\tilde{\boldsymbol{W}}\|_2^2+C\sum_{i=1}^{N_l^t}\varepsilon_i,\ \tilde{\boldsymbol{W}}=[\boldsymbol{W}^{\mathrm{T}},b]^{\mathrm{T}};\\&\text{s.t. }y_i(f^s(\boldsymbol{x}_i)+\boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{x}_i)+b)\geqslant 1-\varepsilon_i,\\&\qquad\varepsilon_i\geqslant 0,\ \forall(\boldsymbol{x}_i,y_i)\in\boldsymbol{D}_l^t.\end{aligned}\tag{4}$$

式(4)的第1项是为了尽量减少新的决策边界与旧的决策边界之间的偏差,第2项控制目标域中训练数据的分类误差.

A-SVM算法中存在着一个关于全局约束的问题,上述新的决策边界不能太偏离源域的分类器,因此式(4)在利用目标域数据计算边界时,并不能确定是否是最大边界. 当$\boldsymbol{D}^t$和$\boldsymbol{D}^s$的相似度较低时,可能出现这一问题.

1.3 CDSVM算法

针对上述问题,文献[15]提出一种Cross-Domain SVM(CDSVM)算法. CDSVM算法的目标是在借助源域$\boldsymbol{D}^s$的相关信息下,通过对数据$\boldsymbol{D}_l^t$的学习得到最适合于$\boldsymbol{D}_u^t$的决策边界. 在CDSVM算法中,定义源域$\boldsymbol{D}^s$的支持向量$\boldsymbol{V}^s=\{(\boldsymbol{v}_1^s,y_1^s),(\boldsymbol{v}_2^s,y_2^s),\cdots,(\boldsymbol{v}_m^s,y_m^s)\}$和源域的决策函数$f^s(\boldsymbol{x})$. 因此,CDSVM算法的目标转化为借助$\boldsymbol{V}^s$对$\boldsymbol{D}_l^t$学习,得到新的决策函数,并将之应用到$\boldsymbol{D}_u^t$分类. 得到的目标函数如下:

$$\begin{aligned}&\min_{\boldsymbol{W}}\frac{1}{2}\|\boldsymbol{W}\|_2^2+C\sum_{i=1}^{|\boldsymbol{D}_l^t|}\varepsilon_i+C\sum_{j=1}^{m}\bar{\varepsilon}_j\cdot\sigma(\boldsymbol{v}_j^s,\boldsymbol{D}_l^t);\\&\text{s.t. }y_i(\boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{x}_i)-b)\geqslant 1-\varepsilon_i,\\&\qquad\varepsilon_i\geqslant 0,\ \forall(\boldsymbol{x}_i,y_i)\in\boldsymbol{D}_l^t,\\&\qquad y_j^s(\boldsymbol{W}^{\mathrm{T}}\phi(\boldsymbol{v}_j^s)-b)\geqslant 1-\bar{\varepsilon}_j,\\&\qquad\bar{\varepsilon}_j\geqslant 0,\ \forall(\boldsymbol{v}_j^s,y_j^s)\in\boldsymbol{V}^s.\end{aligned}\tag{5}$$

对于源域$\boldsymbol{D}^s$中的支持向量$\boldsymbol{V}^s$,权重函数σ定义了源域中的支持向量和数据$\boldsymbol{D}_l^t$的差异程度.

与A-SVM算法类似,CDSVM算法的目标也是通过对源域$\boldsymbol{D}^s$和$\boldsymbol{D}_l^t$的学习得到一个最适合于$\boldsymbol{D}_u^t$的新的决策边界. 但是,CDSVM算法没有过于强调新的决策边界与旧的边界相似这一全局约束条件,相反,基于局部判别的思想,判别的依据仅仅是与$\boldsymbol{D}_l^t$具有相似分布的$\boldsymbol{D}^s$中的数据. 特别地,σ定义为高斯函数形式如下:

$$\sigma(\boldsymbol{v}_j^s,\boldsymbol{D}_l^t)=\frac{1}{|\boldsymbol{D}_l^t|}\sum_{(\boldsymbol{x}_i,y_i)\in\boldsymbol{D}_l^t}\exp\{-\beta\|\boldsymbol{v}_j^s-\boldsymbol{x}_i\|_2^2\}.\tag{6}$$

β控制着支持向量$\boldsymbol{V}^s$的重要性下降速度,β值越大,源域$\boldsymbol{D}^s$中的支持向量$\boldsymbol{V}^s$对$\boldsymbol{D}_l^t$的影响越低,新的决策边界与源域$\boldsymbol{D}^s$的联系越低,算法的迁移效率越低.

上述两种算法A-SVM和CDSVM是基于经典SVM算法分类模型改进的SVM迁移算法. 两种算法通过对源域$\boldsymbol{D}^s$的分类器和目标域$\boldsymbol{D}_l^t$的分类器进行学习,得到最适合于目标域$\boldsymbol{D}_l^t$的新的决策边界,并将

其应用到$\boldsymbol{D}_u^t$中. 但是,在面对多源域时,上述两种算法无法有效地学习各个源域中的信息,对于目标域的分类效果不佳. 因此,对于多源域迁移分类,为了能够有效地学习各源域与目标域有效信息,提出了一种基于源域与目标域之间的相似度学习方法,进而在经典SVM算法的分类模型上提出一种基于相似度学习的多源迁移算法(SL-MSTL).

2　SL-MSTL算法

假设存在c个源域,多源域可以表示为

$$\boldsymbol{D}=\{\boldsymbol{D}_1,\boldsymbol{D}_2,\cdots,\boldsymbol{D}_c\},$$

每一个源域可以表示为

$$\boldsymbol{D}_j=\{(\boldsymbol{x}_i,y_i)\in\boldsymbol{X}\times\boldsymbol{Y}\},\ j=1,2,\cdots,c.$$

其中: $\boldsymbol{X}\in\boldsymbol{R}^d$为源域中样本的特征空间,$\boldsymbol{Y}$为样本的标签向量,$(\boldsymbol{x}_i,y_i)$为相应源域中的一个样本,每个样本的标签$y_i\in\{1,-1\}$,每个源域的分类器可以表示为$f_j(\boldsymbol{x})=\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{x}+b_j$. 目标域$\boldsymbol{D}^t=\boldsymbol{D}_l^t\bigcup\boldsymbol{D}_u^t$,$\boldsymbol{D}_l^t$表示标签已知的少数样本集合,$\boldsymbol{D}_u^t$表示标签未知的样本数据集. $\boldsymbol{D}_l^t=\{(\boldsymbol{x'}_i,y'_i)\in\boldsymbol{X'}\times\boldsymbol{Y'}\}$,其中$i=1,2,\cdots,N$. $\boldsymbol{X'}\in\boldsymbol{R}^d$为目标域$\boldsymbol{D}_l^t$中数据的特征空间,$\boldsymbol{Y'}$为$\boldsymbol{D}_l^t$中的标签向量,$(\boldsymbol{x'}_i,y'_i)$为$\boldsymbol{D}_l^t$中的一个样本,且$\boldsymbol{D}_l^t$中的正负类数据数量一致,测试数据集表示为$\boldsymbol{D}_u^t=\{\boldsymbol{x}''_1,\boldsymbol{x}''_2,\cdots,\boldsymbol{x}''_M\}$.

根据多源迁移学习的策略,通过对每个源域加入相应的权值表示源域与目标域的相似程度,提出多源迁移学习模型,其目标函数为

$$\begin{aligned}&\min_{\omega_i,\beta}J=\\&\frac{1}{2}\sum_{i=1}^{c}\omega_i^2-C\sum_{k=1}^{N}y'_k\Big(\Big(\sum_{i=1}^{c}\omega_i\boldsymbol{W}_i^{\mathrm{T}}\Big)\boldsymbol{x}'_k-\beta\Big).\\&\text{s.t. }y'_k\Big(\Big(\sum_{i=1}^{c}\omega_i\boldsymbol{W}_i^{\mathrm{T}}\Big)\boldsymbol{x}'_k-\beta\Big)>0,\\&\qquad k=1,2,\cdots,N;\\&\sum_{i=1}^{c}\omega_i=1,\ 0\leqslant\omega_i\leqslant1,\ i=1,2,\cdots,c.\end{aligned}\tag{7}$$

其中$\boldsymbol{W}_i^{\mathrm{T}}$表示第$i$个源域对应的分类线投影向量. 对上述目标函数进行优化,得到拉格朗日函数如下:

$$\begin{aligned}&L=\\&\frac{1}{2}\sum_{i=1}^{c}\omega_i^2-C\sum_{k=1}^{N}y'_k\Big(\Big(\sum_{i=1}^{c}\omega_i\boldsymbol{W}_i^{\mathrm{T}}\Big)x'{}_k-\beta\Big)+\\&\lambda\Big(1-\sum_{i=1}^{c}\omega_i\Big)+\sum_{i=1}^{c}\phi_i\omega_i,\end{aligned}\tag{8}$$

其中ϕ_i和λ为拉格朗日乘子. KKT条件为

$$\begin{cases}\dfrac{\partial L}{\partial\omega_i}=0,\\ \phi_i\geqslant0,\\ \phi_i\omega_i=0.\end{cases}\tag{9}$$

由式(9)可见,无法求得ω_i,因此可以先舍弃ω_i非负的情况. 新的拉格朗日函数如下:

$$\begin{aligned}&L=\\&\frac{1}{2}\sum_{i=1}^{c}\omega_i^2-C\sum_{k=1}^{N}y'_k\Big(\Big(\sum_{i=1}^{c}\omega_i\boldsymbol{W}_i^{\mathrm{T}}\Big)\boldsymbol{x}'_k-\beta\Big)+\\&\lambda\Big(1-\sum_{i=1}^{c}\omega_i\Big).\end{aligned}\tag{10}$$

相应的导函数为

$$\begin{cases}\dfrac{\partial L}{\partial\omega_i}=0,\\ \dfrac{\partial L}{\partial\beta}=0,\\ \dfrac{\partial L}{\partial\lambda}=0.\end{cases}\tag{11}$$

求得ω_i的结果为

$$\omega_i=\frac{1}{c}+C\sum_{k=1}^{N}y'_k\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{x}'_k-\frac{C}{c}\sum_{k=1}^{N}y'_k\sum_{j=1}^{c}\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{x}'_k.\tag{12}$$

由式(12)可见,ω_i也可能出现负值,因此可以将式(12)改写为

$$\omega_i=\begin{cases}0,\ i\in\boldsymbol{c}^-;\\ \dfrac{1}{|\boldsymbol{c}^+|}+CF_i,\ i\in\boldsymbol{c}^+.\end{cases}\tag{13}$$

其中

$$F_i=\sum_{k\in\boldsymbol{N}^+}y'_k\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{x}'_k-\frac{1}{|\boldsymbol{c}^+|}\sum_{k\in\boldsymbol{N}^+}y'_k\sum_{j\in\boldsymbol{c}^+}\boldsymbol{W}_j^{\mathrm{T}}\boldsymbol{x}'_k,\tag{14}$$

$\boldsymbol{c}^+$表示所有使ω_i为正的i的集合,$\boldsymbol{c}^-$表示所有使ω_i为非正的i的集合. 定义

$$\boldsymbol{N}^+=\Big\{k\in\boldsymbol{N}:y'_k\Big(\Big(\sum_{\mathrm{i}=1}^{\boldsymbol{c}^+}\omega_i\boldsymbol{W}_i^{\mathrm{T}}\Big)\boldsymbol{x}'_k-\beta\Big)>0\Big\},$$

$|\boldsymbol{c}^+|$和$|\boldsymbol{c}^-|$表示集合的大小.

使用阈值下降的方法求解β. 定义β为

$$\nabla_\beta J=C\sum_{k\in\boldsymbol{N}^+}y'_k.\tag{15}$$

由梯度下降法得到新的解为

$$\beta'=\beta-\gamma\nabla_\beta J,\tag{16}$$

其中γ表示梯度下降的速率,设置$\gamma=1/t$.

求解集合$\boldsymbol{c}^+$和$\boldsymbol{c}^-$的算法如下.

算法1　求解集合$\boldsymbol{c}^+$和$\boldsymbol{c}^-$.

Step 1: 初始化. $\boldsymbol{c}_0^+=\varnothing,\boldsymbol{c}_0^-=\{1,2,\cdots,c\},n=$

0.

Step 2: $n=n+1, \boldsymbol{c}_n^+=\boldsymbol{c}_{n-1}^+ +\{i\}, \boldsymbol{c}_n^-=\boldsymbol{c}_{n-1}^- -\{i\}$,其中$i=\arg\max\limits_{i\in\boldsymbol{c}_{n-1}^-}\{F_i\}$.

Step 3: 通过式(13)计算ω_j,并判断其是否大于0,$j=\arg\max\limits_{i\in\boldsymbol{c}_n^+}\{F_i\}$. 如果$\omega_j>0$,则返回Step 2,否则设置$\boldsymbol{c}_n^+=\boldsymbol{c}_{n-1}^+, \boldsymbol{c}_n^-=\boldsymbol{c}_{n-1}^-$,并终止.

求解$\boldsymbol{\omega}$的具体算法过程如下.

算法2 求解权值向量$\boldsymbol{\omega}$.

输入: 数据矩阵$\boldsymbol{D}_l^t=\{(\boldsymbol{x}_i', y_i')\in\boldsymbol{X}'\times\boldsymbol{Y}'\}, i=1,2,\cdots,N, y_i\in\{1,-1\}$;各源域分类器$\{\boldsymbol{W}_1,\boldsymbol{W}_2,\cdots,\boldsymbol{W}_c\},\{b_1,b_2,\cdots,b_c\}$;惩罚因子$C$.

输出: 源域权值$\boldsymbol{\omega}$,阈值β.

Step 1: 初始化. ${\omega_1}^{(0)}={\omega_2}^{(0)}=\cdots={\omega_c}^{(0)}=1/c, \beta=\beta^{(0)}=\sum\limits_{i=1}^{c}\omega_i^{(0)}\cdot b_i$.

Step 2: 设置迭代步数$t=1$.

Step 3: 循环,直至收敛.

Step 3.1: 更新学习率$\gamma=1/t$, $t=t+1$;

Step 3.2: 更新训练子集

$$\boldsymbol{N}^+=\Big\{k\in\boldsymbol{N}: y_k'\Big(\Big(\sum_{i=1}^{\boldsymbol{c}^+}\omega_i\boldsymbol{W}_i^{\mathrm{T}}\Big)\boldsymbol{x}_k'-\beta\Big)>0\Big\},$$

利用式(15)计算梯度$\nabla_\beta J$;

Step 3.3: 更新阈值$\beta'=\beta-\gamma\nabla_\beta J$;

Step 3.4: 更新集合$\boldsymbol{c}^+$、$\boldsymbol{c}^-$和算法1;

Step 3.5: 更新$\boldsymbol{\omega}$.

通过对多源域和一部分带有标签的目标域数据的学习,得到适用于目标域的分类模型. 利用新的模型对测试数据进行分类,具体过程如下.

算法3 目标域分类.

输入: 待测数据矩阵$\boldsymbol{D}_u^t=\{\boldsymbol{x}_1'', \boldsymbol{x}_2'', \cdots, \boldsymbol{x}_M''\}$,各源域权值$\boldsymbol{\omega}=\{\omega_1,\omega_2,\cdots,\omega_c\}$,各源域相应的分类器投影向量$\boldsymbol{W}=\{\boldsymbol{W}_1,\boldsymbol{W}_2,\cdots,\boldsymbol{W}_c\},\beta$;

输出: 待检测数据的标签label.

Step 1: 利用训练得到的模型对每个数据进行如下运算:

$$y(m)=\Big(\sum_{i=1}^{c}\omega_i\boldsymbol{W}_i^{\mathrm{T}}\Big)\boldsymbol{x}_m''-\beta,$$

$$m=1,2,\cdots,M.$$

Step 2: 若$y(m)>0$,则label$=1$,否则label$=-1$.

3 实验分析

通过人工数据集和真实数据集验证算法的有效性和实用性,所有的实验数据集都是二分类数据集.

3.1 实验准备和说明

共安排3组对比实验,并与现有的经典算法进行对比,有SVM、CDSVM、TrAdaBoost[16]. 为了更好地验证算法的有效性和实用性,使用人工数据集和真实数据集. 各算法的参数设置如下: CDSVM中的$C=10$, $\beta=10$, SL-MSTL中的$C=10^{-4}$.

由于现有的真实数据集很少为跨领域算法设定,需要对数据集进行处理. 实验数据集包括: 3个源域,1个目标域,源域为大数据域,目标域为稀有域,属于小数据域. 其中目标域的10%数据集是带标签的训练数据集和90%被隐去标签的测试数据集,所有数据集都取自人工数据集或真实数据集,且每组域只含有正负两类数据,在构造数据域时要注意源域与目标域之间要有相似度,即源域与目标域的分布相似. 在数据的准备中,主要分为代表正类的父类$\boldsymbol{A}=\{\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_n\}$和代表负类的父类$\boldsymbol{B}=\{\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_m\}$,其中每个父类中包含若干个子类,子类之间属性相似但分布不同. 通过在A类和B类中随机选取相应的子类构造3组源域,每组源域可以表示为$\boldsymbol{C}=\{\boldsymbol{A}_1,\boldsymbol{A}_2,\cdots,\boldsymbol{A}_i\}\bigcup\{\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_j\}$,每个源域中的正负类子类个数随机. 目标域$\boldsymbol{D}^t$由余下子类中各选取一个子类构成,其中选取10%数据作为标签已知训练数据,训练数据中正负类数据个数必须一致,余下数据作为测试数据. 实验结果采用均值和标准差的形式,每组实验运行20次.

为了对算法的分类性能做出评估,本文使用数据的分类正确率作为评价标准,分类正确率具体表示为: 分类正确的个数与测试样本总数的比值. 由于对比算法基本上都是单源迁移算法,实验中将多个源域合并为一个源域作为算法的源域.

3.2 人工数据集

为了验证算法的有效性,实验使用自己构造的人工数据集. 本文采用人为模拟的二维数据集,采用二维曲线模拟生成数据,生成函数$y=Ax^3+Bx^2+Cx+D$,其中A, B, C是针对多源域和目标域而设定的参数,通过不同D值区分数据的正负类,$\boldsymbol{D}^+$设定了正类,$\boldsymbol{D}^-$设定了负类. 具体设定如表1所示.

表1 二维曲线参数值设定

源域/目标域	A	B	C	$\boldsymbol{D}^+$	$\boldsymbol{D}^-$	x取值
源域1	0.1	0.2	0.1	[1,3]	[−1,1]	[−6,2]
源域2	−0.2	−0.1	0.1	[−1,1]	[−3,−1]	[−4,2]
源域3	−0.1	0.2	0.2	[1.5,3.5]	[−0.5,1.5]	[−2,5]
目标域	−0.1	−0.1	0.1	[−0.5,1.5]	[−2.5,−0.5]	[−2,2]

本次实验共设置3个大源域和1个目标域,每个源域的数据个数为2 000,目标域的个数为200. 按照上述方式构造的源域和目标域,由于其构造函数之间有一定的相似性,在相应系数的设定上存在着不同,因此它们之间有一定的相似度和差异度,此数据集可以用作本文的实验. 为了改变各源域与目标域之间的相似度,本文对数据作出了旋转和加噪声处理,噪声是高斯白噪声,对源域的旋转角度分别设定为$\{5,-5,10\}$,实验中数据集添加的噪声是Matlab自带的噪声函数awgn,设定的噪声参数分别为$\{5\,\mathrm{db},10\,\mathrm{db}\}$. 具体操作参数如表2所示.

表3为人工数据集实验结果. 实验结果表明,在对源域不同变换的情况下,本文所提出的SL-MSTL迁移算法的分类效果比传统的SVM算法、CDSVM算法、TrAdaBoost算法效果要好; 在相同旋转角度的情况下,随着信噪比的增加,算法的分类效果降低,但是由于传统SVM算法没有进行迁移学习,容易受到影响,分类的平均准确率下降较快,如MD2、MD6和MD4、MD5组; 本文提出的SL-MSTL通过学习各源域与目标域之间的相似度,减小了由于源域的变化给分类结果带来的影响,因此算法的性能相对其他3种算法下降得较少,保持了较好的迁移学习性能.

表 2　二维曲线设定

组合	源域1	源域2	源域3	目标域
MD1	不作处理	不作处理	不作处理	不作处理
MD2	旋转 $-5°$	旋转 $5°$	加噪 10 db	不作处理
MD3	旋转 $-5°$	旋转 $10°$	加噪 5 db	不作处理
MD4	旋转 $10°$	旋转 $5°$	加噪 5 db	不作处理
MD5	旋转 $10°$	加噪 5 db	加噪 10 db	不作处理
MD6	旋转 $-5°$	旋转 $5°$	加噪 5 db	不作处理

表 3　人工数据集实验结果

组合	算法			
	SVM	CDSVM	TrAdaBoost	SL-MSTL
MD1	0.583 1±0.009 1	0.703 4±0.014 1	0.637 5±0.055 9	**0.820 9±0.014 5**
MD2	0.605 0±0.010 8	0.751 6±0.012 0	0.598 4±0.021 8	**0.835 6±0.013 5**
MD3	0.702 8±0.010 4	0.745 6±0.014 6	0.636 9±0.053 1	**0.832 2±0.011 3**
MD4	0.624 4±0.009 0	0.778 4±0.028 9	0.529 7±0.031 0	**0.833 7±0.010 8**
MD5	0.594 4±0.009 3	0.673 1±0.013 8	0.500 0±0.000 0	**0.827 2±0.013 9**
MD6	0.669 1±0.011 4	0.753 7±0.022 7	0.616 2±0.026 6	**0.835 0±0.011 5**

3.3　文本数据集

为了验证算法的实用性,实验数据集采用文本数据集NewsGroup20. 选取此数据集中的comp、rec、sci、talk四个父类作为总数据源,每个父类下面有4个子类,具体的源域和目标源组合如表4所示. 为了构造多源域和目标域,分别选取不同父类下的4个子类,通过子类的两两组合构成源域和目标域. 因为来自同一个父类下的子类之间具有一定的相似性和差异性,所以组合成的源域与源域、源域与目标域之间具有一定的相似度和差异性,即源域之间、源域与目标域之间不是同分布,这满足了多源迁移学习的前提条件. 每组实验共有3个源域,每个源域大约2 000个,目标源中随机均匀选取200个数据作为目标域,均匀选取目标域10%的数据作为训练数据,余下的作为测试数据. 为保证实验的公正性,每组实验运行20次,结果取均值.

表5为文本数据集实验结果. 实验结果表明,SL-MSTL算法的分类准确率不仅比传统的SVM算法的分类准确率高,而且比常用的迁移算法CDSVM、TrAdaBoost算法的准确率要高. 由于传统的SVM算法没有引入迁移学习机制,算法的分类结果低于SL-MSTL算法. 而迁移学习算法CDSVM、TrAdaBoost

表4 文本数据组合细节

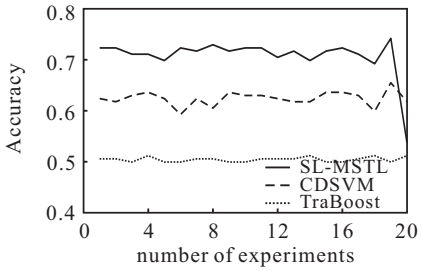
组合	数据集	源域/目标源	正例	负例
NG1	comp vs rec	源域	comp.windows.x	rec.sport.baseball
			comp.graphics	rec.sport.hockey
			comp.os.ms-windows.misc	rec.motorcycles
		目标源	comp.sys.mac.hardware	rec.autos
NG2	comp vs sci	源域	comp.sys.ibm.pc.hardware	sci.space
			comp.windows.x	sci.electronics
			comp.os.ms-windows.misc	sci.crypt
		目标源	comp.graphics	soc.religion.christian
NG3	comp vs talk	源域	comp.sys.ibm.pc.hardware	talk.politics.misc
			comp.graphics	talk.religion.misc
			comp.os.ms-windows.misc	talk.politics.guns
		目标源	comp.sys.mac.hardware	talk.politics.mideast
NG4	rec vs sci	源域	rec.motorcycles	sci.electronics
			rec.autos	sci.crypt
			rec.sport.hockey	sci.space
		目标源	rec.sport.baseball	sci.med
NG5	rec vs talk	源域	rec.autos	talk.politics.misc
			rec.sport.hockey	talk.politics.guns
			rec.sport.baseball	talk.religion.misc
		目标源	rec.motorcycles	talk.politics.mideast
NG6	sci vs talk	源域	sci.crypt	talk.politics.misc
			sci.space	talk.politics.guns
			sci.med	talk.religion.misc
		目标源	sci.electronics	talk.politics.mideast

表5 文本数据集实验结果

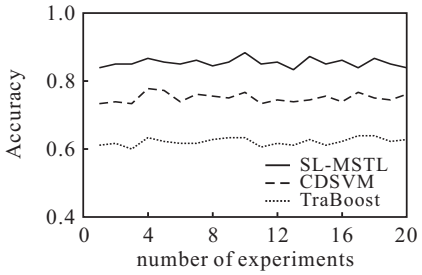
组合	算法			
	SVM	CDSVM	TrAdaBoost	SL-MSTL
NG1	0.615 6±0.017 3	0.625 3±0.014 3	0.505 0±0.004 8	**0.708 7±0.042 0**
NG2	0.846 7±0.009 4	0.750 3±0.013 7	0.621 7±0.010 9	**0.853 6±0.012 4**
NG3	0.840 3±0.007 2	0.727 2±0.006 5	0.680 3±0.008 4	**0.857 2±0.035 4**
NG4	0.581 3±0.018 8	0.588 4±0.016 5	0.512 5±0.017 5	**0.717 5±0.021 5**
NG5	0.667 2±0.010 6	0.654 4±0.010 0	0.601 9±0.009 6	**0.721 7±0.042 0**
NG6	**0.777 8±0.009 0**	0.653 9±0.009 7	0.647 2±0.012 0	0.717 2±0.027 8

不能有效地学习各源域与目标域的相似度,无法有效地学习出源域中的有用信息,因此算法的实用性能低于SL-MSTL算法,见图1.

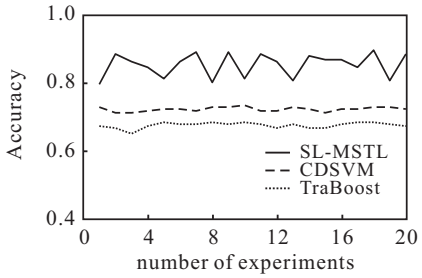
由图1可以看出,在相同的情况下,SL-MSTL算法的平均准确率高于其他两种迁移算法,从而表明SL-MSTL算法的实用性较强.



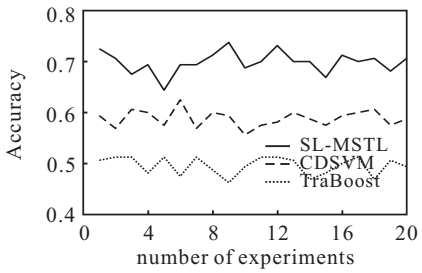
(a) comp vs rec



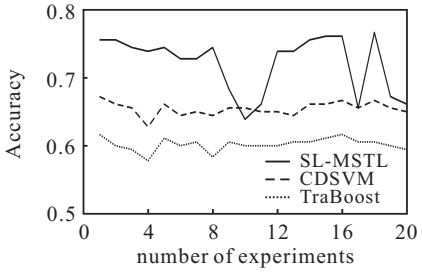
(b) comp vs sci



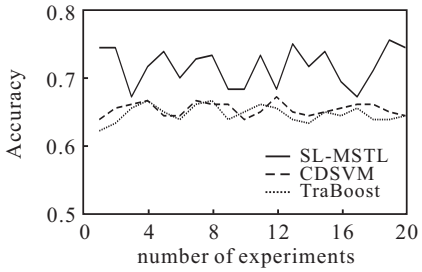
(c) comp vs talk



(d) rec vs sci



(e) rec vs talk



(f) sci vs talk

图 1　迁移算法在不同组合上的准确率对比

3.4　图像数据集

为了进一步验证算法的实用性,实验使用图像数据集,图像数据集选取 Caltech 256 图像数据集,下载已经处理好的图像数据集(http://files.is.tue.mpg.de/pgehler/projects/iccv09/),并选取其中一种图像描述方式: PHOG 形状描述[17]. 本文只选取其中第 1 层次的数据集,即选取图像数据的特征为20维.

图像数据集和文本数据集一样,含有多种大类,每个大类下面包含若干个子类. 本次实验选取4个大类,由于图像数据集的每个子类数量较少,本文将每个大类下的若干子类随机打乱合并成一个总的正或负源,从里面随机选取不同的数量组成各源域. 由于同属于一个父类下的子类之间都有一定的相似性,不同的父类之间有差异性,因此组合成的多源域和目标域之间有一定的相似性和差异性,这样的多源域和目标域可以应用于多源迁移学习.

图像多源域和目标域的具体构造方法如下: 首先选取两个父类作为总的正负类,再各从每个父类的下面选择数个子类,见表6. 将属于同一父类下的4个子类数据集随机打乱顺序构成一个总的数据域,得到两个总的数据域. 分别从两个总域中随机选取一定量但不等量的数据构成3个源域,每个源域的数据约200个,再从余下的数据中等量选取一定量的数据集作为目标域,数量约20个. 由于目标域图像数据较少,为了尽可能学习源域的相似度,实验均匀选取目标域中20% 数据作为训练数据集,余下的作为测试数据集. 由于源域和目标域的数据相对较少,每组实验运行100次,实验结果取平均值.

实验结果表明,在相同情况下,本文提出的SL-MSTL 算法的分类效果较好. 在大部分情况下,由于传统的SVM 算法没有迁移学习的性质,传统的SVM 算法分类性能比其他迁移算法的准确率低. 由于在构造源域时,每个源域中所包含的子类种类和数量各不相同,在对源域进行学习时,不能得到适合于目标域的最佳的分类器. 本文提出的 SL-MSTL 首先对多源域进行相似度学习,通过学习得到各源域与目标域的相似度,得到最适合于目标域的分类器,因此 CDSVM 算法的分类性能比 SL-MSTL 算法低. TrAdaBoost 算法通过选取不同数据生成若干分类器,通过不断迭代将其加权组合成最适合于目标域的分类器. 由于源域中的数据图像来自不同的子类且数量不同, TrAdaBoost 算法构建的分类器不能尽可能地体现数据的特征,算法的分类性能比 SL-MSTL 低. 实验结果验证了这一结论,具体数据如表7所示.

表6 图像数据集细节

父类名称	Animal	Traffic	Building	HA
子类名称	camel, dog, dolphin, elephant, elk, giraffe, goat, horse, kangaroo	blimp, canone, firetruck, helicopter, ketch, motorbikes	eiffel-tower, golden-gate-bridge, minaret, skyscraper, tower-pisa, smokestack	Breadmarker, computer-monitor, ipod washing-machine

表7 图像数据集实验结果

组合	算法			
	SVM	CDSVM	TrAdaBoost	SL-MSTL
animal vs traffic	0.653 0±0.042 8	0.501 7±0.012 3	0.620 2±0.057 3	**0.689 8±0.045 4**
animal vs building	0.726 3±0.007 7	0.731 6±0.007 7	0.742 1±0.007 7	**0.832 3±0.045 3**
traffic vs building	0.650 9±0.008 3	0.720 9±0.007 3	0.706 8±0.035 2	**0.759 8±0.040 6**
traffic vs HA	0.755 9±0.008 0	0.756 3±0.007 5	0.768 7±0.020 4	**0.787 3±0.009 8**
building vs HA	0.631 2±0.007 6	0.641 2±0.007 2	0.632 0±0.040 9	**0.678 8±0.063 7**

4 结 论

本文针对源域中包含大量带标签数据,目标域中只包含少量带标签数据的情况,提出了一种基于相似度学习的多源域迁移算法SL-MSTL. SL-MSTL算法在传统SVM算法分类框架的基础上加入了迁移学习框架,并通过源域和目标域之间的相似度学习将源域中的有用信息迁移到目标域中. SL-MSTL算法将相似度学习与迁移机制相结合,使得算法能够有效学习源域和目标域之间的相似度,提高了算法的学习效率. 通过对源域和目标域之间的相似度学习,尽可能地将源域中对于目标域重要的信息有效地学习出来,提高了迁移学习的正确率.

参考文献(References)

[1] Chen J, Ji S, Ceran B, et al. Learning subspace kernels for classification[C]. ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. Las Vegas: ACM, 2008: 106-114.

[2] Gereke M, Jung S, Buer J, et al. On kernel-target alignment[J]. Advances in Neural Information Processing Systems, 2010, 179(5): 367-373.

[3] Qamar A M, Gaussier E, Chevallet J P, et al. Similarity learning for nearest neighbor dlassification[C]. The 8th IEEE Int Conf on Data Mining. Pisa: IEEE, 2008: 983-988.

[4] 张景祥, 王士同. 基于共同决策方向矢量的多源迁移及其快速学习方法[J]. 电子学报, 2015, 43(7): 1349-1355.
(Zhang J X, Wang S T. Common-decision-vector based multiple source transfer learning classification and its fast learning method[J]. Acta Electronica Sinica, 2015, 43(7): 1349-1355.)

[5] Pan S J, Kwok J T, Yang Q. Transfer learning via dimensionality reduction[C]. National Conf on Artificial Intelligence. Washington: AAAI Press, 2008: 677-682.

[6] Pan S J, Ni X, Sun J T, et al. Cross-domain sentiment classification via spectral feature alignment[C]. Int Conf on World Wide Web. North Carolina: ACM, 2010: 751-760.

[7] Sun S. Multi-view laplacian support vector machines[J]. Applied Intelligence, 2013, 41(4): 209-222.

[8] Xu Z, Sun S. Multi-view transfer learning with adaboost[C]. The 23rd IEEE Int Conf on Tools with Artificial Intelligence. Boca Raton: IEEE Computer Society, 2011: 399-402.

[9] Zhuang F, Luo P, Xiong H, et al. Cross-domain learning from multiple sources: A consensus regularization perspective[J]. IEEE Trans on Knowledge & Data Engineering, 2010, 22(12): 1664-1678.

[10] 蒋亦樟, 邓赵红, 王士同. ML型迁移学习模糊系统[J]. 自动化学报, 2012, 38(9): 1393-1409.
(Jiang Y Z, Deng Z H, Wang S T. Mamdani-larsen type transfer learning fuzzy system[J]. Acta Automatica Sinica, 2012, 38(9): 1393-1409.)

[11] Ling X, Dai W, Xue G R, et al. Spectral domain-transfer learning[C]. ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. Las Vegas: ACM, 2008: 488-496.

[12] Gao J, Fan W, Sun Y, et al. Heterogeneous source consensus learning via decision propagation and negotiation[C]. ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. Paris: ACM, 2009: 339-347.

[13] Cortes C, Vapnik V. Support vector network[J]. Machine Learning, 1995, 20(3): 273-297.

[14] Yang J, Yan R, Hauptmann A G. Cross-domain video concept detection using adaptive svms[C]. Int Conf on Multimedia 2007. Augsburg: ACM, 2007: 188-197.

[15] Jiang W, Zavesky E, Chang S F, et al. Cross-domain learning methods for high-level visual concept classification[C]. The 15th IEEE Int Conf on Image Processing. San Diego: IEEE, 2008: 161-164.

[16] Dai W, Yang Q, Xue G R, et al. Boosting for transfer learning[C]. Proc of the 24th Int Conf on Machine Learning. Corvalis: ACM, 2007: 193-200.

[17] Bosch A, Zisserman A, Munoz X. Representing shape with a spatial pyramidkernel[C]. ACM Int Conf on Image and Video Retrieval. Amsterdam: ACM, 2007: 401-408.

（责任编辑：郑晓蕾）