

## 基于 SVC 和 SVR 约束组合的迁移学习分类算法

毕安琪, 王士同

(江南大学 数字媒体学院, 江苏 无锡 214122)

**摘要:** 根据迁移学习思想, 针对分类问题, 以支持向量机(SVM)模型为基础提出一种新的迁移学习分类算法 CC-TSVM. 该方法以邻域间的分类超平面为纽带实现源域对目标域的迁移学习. 具体地, 以支持向量分类的约束条件完成对目标域数据的学习, 获取分类超平面参数, 再以支持向量回归的约束条件有效利用源域数据矫正目标域超平面参数, 并在上述组合约束的共同作用下实现邻域间迁移, 提高分类器性能. 在人工和真实数据集上的实验表明, 所提出算法具有良好的迁移能力和优越的分类性能.

**关键词:** 支持向量机; 支持向量分类; 支持向量回归; 迁移学习

**中图分类号:** TP391.4

**文献标志码:** A

## Transfer classification learning based on combination of both SVC and SVR's constraints

BI An-qi, WANG Shi-tong

(School of Digital Media, Jiangnan University, Wuxi 214122, China. Correspondent: BI An-qi, E-mail: angela.sue.bi@gmail.com)

**Abstract:** Transfer learning focuses on solving learning tasks in the target domain by leveraging the useful information in the target domain. Therefore, a novel transfer classification learning algorithm called CC-TSVM is proposed, which is based on SVM learning framework. CC-TSVM firstly adopts SVC to get the hyperplane for the source domain and then corrects the obtained hyperplane as the hyperplane for the target domain by using SVR with the constraints for the labeled data in the target domain. The experimental results on artificial and real datasets show that the proposed algorithm has high classification accuracy and can well leverage the useful information in the source domain.

**Key words:** support vector machine; support vector classification; support vector regression; transfer learning

### 0 引言

近年来, 迁移学习理论得到了不断的发展, 根据迁移学习的特征可归纳为3个方面: 归纳式迁移学习<sup>[1-2]</sup>、直推式迁移学习<sup>[3-4]</sup>和无监督迁移学习<sup>[5-6]</sup>. 其中: 归纳式和直推式迁移学习为当前研究的重点; 无监督迁移学习由于难度较大, 研究得较少. 目前较典型的研究成果有: 文献[7]提出了一种基于辅助集的迁移学习算法; 文献[8]提出了基于传统的Logistic回归算法的Migratory-Logit迁移算法(即M-Logit算法); 文献[9]提出了一种基于Adaboost算法的迁移学习分类算法TrAdaboost<sup>[10]</sup>; 文献[11]提出了LMPROJ算法; 文献[12]提出了TL-SSVM算法; 文献[13-15]提出了基于马科夫逻辑的迁移学习算法.

本文以分类问题为具体研究对象, 结合经典的支持向量机(SVM)模型<sup>[16]</sup>, 探讨一种新的迁移学习机制以实现邻域间的迁移学习. 值得指出的是, 当前基于SVM的迁移分类器较为常见<sup>[11-12, 17]</sup>, 研究这类算法可发现当前的迁移策略一般均以邻域间的超平面参数为迁移对象, 以平方逼近原则为具体实现方式完成源域到目标域的迁移. 该方法确实是一种有效的迁移学习方式, 但上述逼近方式不易控制, 且性能常受限于逼近项前的迁移学习人工设置参数.

针对上述问题, 本文仍以邻域间的分类超平面为纽带, 但不再使用平方逼近原则, 具体地以支持向量分类(SVC)的约束条件完成对目标域数据的学习, 获取分类超平面参数, 再以支持向量回归(SVR)的约束

收稿日期: 2013-04-26; 修回日期: 2013-10-23.

基金项目: 国家自然科学基金项目(61272210).

作者简介: 毕安琪(1989—), 女, 博士生, 从事模式识别、人工智能的研究; 王士同(1964—), 男, 教授, 博士生导师, 从事模式识别、人工智能等研究.

条件有效利用源域数据校正目标域超平面参数. 通过上述组合约束的共同作用来实现源邻域到目标域邻域的迁移, 并给出具体的学习算法 CC-TSVM 算法. 可以看出, 所提出的方法也是一种直推式迁移学习算法, 要求源域类别信息已知, 而目标域类别信息未知, 但采用的迁移学习策略与以往的方法有较大的不同.

## 1 CC-TSVM 算法

### 1.1 问题定义

给定目标域  $T$  和源域  $S$ , 样本的特征空间相同,  $\mathbf{x}$  为特征向量,  $y$  为标记值, 则有

$$T = \{(\mathbf{x}_i, y_i) | i = 1, 2, \dots, N_t\},$$

$$S = \{(\mathbf{x}_i, y_i) | i = 1, 2, \dots, N_s\},$$

其中  $N_t$  和  $N_s$  分别为目标域和源域的已有类标的样本数. 假定需要迁移的场景是目标域的  $N_t$  较小, 故需要借鉴源域的知识得到目标域的分类函数.

本文考虑经典的二分类问题, 定义  $y_i$  为 +1 或 -1, 多分类问题可通过二分类两两组合解决, 不在本文的讨论范围内. 针对迁移场景, 假定源域样本服从分布  $P_S$ , 记为  $(\mathbf{X}_S, \mathbf{Y}_S) \sim P_S(\mathbf{X}, \mathbf{Y})$ , 目标域样本服从分布  $P_T$ , 记为  $(\mathbf{X}_T, \mathbf{Y}_T) \sim P_T(\mathbf{X}, \mathbf{Y})$ .  $P_S$  与  $P_T$  相似但不相同, 源域的类标  $\mathbf{Y}_S$  已知, 目标域的类标  $\mathbf{Y}_T$  未知. 本文研究的问题是利用源域的样本和样本分布相似的迁移特性, 训练适用于目标域的高精度分类模型.

### 1.2 CC-TSVM 算法构造

参照 SVR 理论中的回归逼近“间隔”思想, 定义一种具备目标域向源域借鉴迁移知识的“间隔”约束, 进一步利用 SVC 模型的大间隔分类思想构造专属于目标域的分类约束, 在上述两大约束的共同作用下完成邻域间的迁移. 对源域  $S$  应用经典 SVC 算法, 目标函数为

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i.$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N;$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, N. \quad (1)$$

由式(1)得到基于源域的 SVM 分类器  $f_S$ , 将  $f_S$  作用于源域数据集, 得到原始预测值, 记为  $y_{s,k}$ , 有

$$y_{s,k} = \mathbf{w}_s^T \mathbf{x}_k + b_s, \quad k = 1, 2, \dots, N_s. \quad (2)$$

在训练目标域数据集时,  $y_{s,k}$  可以作为超平面需要逼近的一个平面, 即寻找与分类函数  $f_S$  “间隔”最小的超平面, 作为目标域的分类函数. 因此, 求解源域的  $y_{s,k}$  为训练新的针对目标域的分类器提供了前提.

鉴于以上分析, 给出 CC-TSVM 算法的原理依据如图 1 所示. 图 1(c) 中  $H$  是分类面,  $H_1$  和  $H_2$  是平行于

$H$  且离  $H$  最近的两个样本的直线, 使得  $H_1$  与  $H$ 、 $H_2$  与  $H$  距离最大; 图 1(d) 中, 在源域分类器的基础上引入  $\varepsilon$  控制新的分类器和源域分类器的距离, 保证源域和目标域数据的紧密联系的约束, 找到对目标域分类精度最高的超平面  $H$ . 该算法与 SVC 相似, 最终也需要寻找一个最优超平面  $H$ , 使得间隔尽量最大, 如图 1(c) 所示, 要求  $H_1$  与  $H$ 、 $H_2$  与  $H$  距离最大; 另一方面, CC-TSVM 算法利用 SVR 理论, 保证  $H$  与源域  $S$  分类超平面的偏差最小, 如图 1(b) 和图 1(d) 所示, 通过误差函数  $\varepsilon$  将  $H$  控制在  $y_{s,k} + \varepsilon$  和  $y_{s,k} - \varepsilon$  之间.

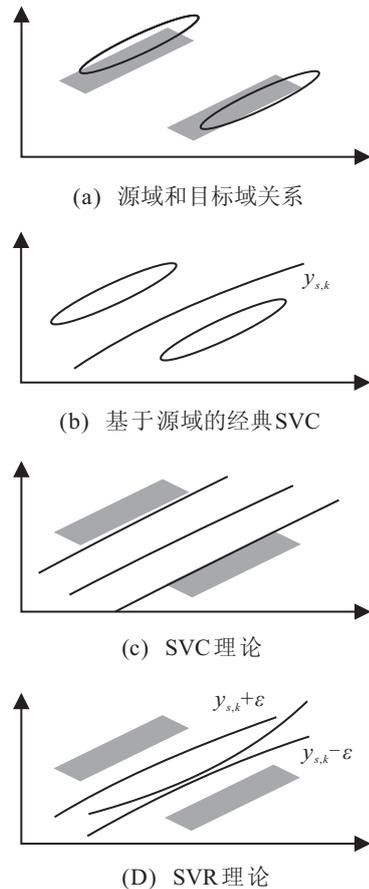


图 1 CC-TSVM 算法原理

本文引入松弛变量  $\xi_{c,i}$ ,  $\xi_{r^+,k}$  和  $\xi_{r^-,k}$  ( $k = 1, 2, \dots, N_s$ ), 定义  $C_1$  和  $C_2$  为容错惩罚系数, 结合迁移学习的思想, 构造如下优化问题, 从而得到 CC-TSVM 迁移学习算法:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{N_t} \xi_{c,i} + C_2 \sum_{k=1}^{N_s} (\xi_{r^+,k} + \xi_{r^-,k}).$$

$$\text{s.t. } y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_{c,i}, \quad i = 1, 2, \dots, N_t;$$

$$(\mathbf{w}^T \mathbf{x}_{s,k} + b) - y_{s,k} \geq \varepsilon + \xi_{r^+,k}, \quad k = 1, 2, \dots, N_s;$$

$$y_{s,k} - (\mathbf{w}^T \mathbf{x}_{s,k} + b) \geq \varepsilon + \xi_{r^-,k}, \quad k = 1, 2, \dots, N_s;$$

$$\xi_{c,i} \geq 0, \quad i = 1, 2, \dots, N_t;$$

$$\xi_{r^+,k} \geq 0, \quad \xi_{r^-,k} \geq 0, \quad k = 1, 2, \dots, N_s. \quad (3)$$

其中:  $\mathbf{x}_{s,k}$  为源域数据的特征向量,  $\mathbf{x}_i$  为目标域数据

的特征向量,  $y_i$  为目标域数据的类标值. 式 (3) 体现了数据分类和回归两者的特点, 将以上分类算法称为 CC-TSVM 迁移学习分类算法, 决策函数如下:

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b. \quad (4)$$

### 1.3 核化的 CC-TSVM 算法

在支持向量的理论中,  $\mathbf{x}_i^T \mathbf{x}$  的形式使得核函数的应用成为可能, 另一方面, 在将输入的样本映射到高维特征空间中时, 选择合适的核函数非常必要和关键. 应用核函数后, 源域的经典 SVM 分类器的决策函数为

$$f_S(\mathbf{x}) = \mathbf{w}_S^T \mathbf{x} + b_S = \sum_{i=1}^{N_s} a_i y_i K(\mathbf{x}_i, \mathbf{x}) + b_S. \quad (5)$$

由于 CC-TSVM 迁移分类算法基于 SVM 分类和 SVR 算法, 核函数也可以应用到该算法中. 常见的核函数有线性核函数、高斯核函数和多项式核函数, 本文实验中, 核函数均采用高斯核函数, 核参数  $\sigma$  按照 5 重交叉验证的方式获取.

### 1.4 CC-TSVM 迁移分类算法的优化问题

构造 CC-TSVM 迁移分类算法的拉格朗日方程

$$\begin{aligned} L(\mathbf{w}, b, \xi_{c,i}, \xi_{r^+,k}, \xi_{r^-,k}, \lambda_{c,i}, \lambda_{r^+,k}, \\ \lambda_{r^-,k}, \eta_{c,i}, \eta_{r^+,k}, \eta_{r^-,k}) = \\ \frac{1}{2} \|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{N_t} \xi_{c,i} + C_2 \sum_{k=1}^{N_s} (\xi_{r^+,k} + \xi_{r^-,k}) + \\ \sum_{i=1}^{N_t} \lambda_{c,i} [1 - \xi_{c,i} - y_i (\mathbf{w}^T \mathbf{x}_i + b)] + \\ \sum_{k=1}^{N_s} \lambda_{r^+,k} [(\mathbf{w}^T \mathbf{x}_{s,k} + b) - y_{s,k} - \varepsilon - \xi_{r^+,k}] + \\ \sum_{k=1}^{N_s} \lambda_{r^-,k} [y_{s,k} - (\mathbf{w}^T \mathbf{x}_{s,k} + b) - \varepsilon - \xi_{r^-,k}] - \\ \sum_{i=1}^{N_t} \eta_{c,i} \xi_{c,i} - \sum_{k=1}^{N_s} \eta_{r^+,k} \xi_{r^+,k} - \sum_{k=1}^{N_s} \eta_{r^-,k} \xi_{r^-,k}, \quad (6) \end{aligned}$$

其中  $\lambda_{c,i}, \lambda_{r^+,k}, \lambda_{r^-,k}, \eta_{c,i}, \eta_{r^+,k}, \eta_{r^-,k} \geq 0$  为拉格朗日乘子.  $L(\mathbf{w}, b, \xi_{c,i}, \xi_{r^+,k}, \xi_{r^-,k}, \lambda_{c,i}, \lambda_{r^+,k}, \lambda_{r^-,k}, \eta_{c,i}, \eta_{r^+,k}, \eta_{r^-,k})$  对原始问题变量  $\mathbf{w}, b, \xi_{c,i}, \xi_{r^+,k}, \xi_{r^-,k}$  求偏导数, 并考虑满足 K.K.T 条件, 可得

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N_t} \lambda_{c,i} y_i \mathbf{x}_i + \sum_{k=1}^{N_s} \lambda_{r^+,k} \mathbf{x}_{s,k} - \\ \sum_{k=1}^{N_s} \lambda_{r^-,k} \mathbf{x}_{s,k} = 0; \quad (7) \end{aligned}$$

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^{N_t} \lambda_{c,i} y_i + \sum_{k=1}^{N_s} \lambda_{r^+,k} - \sum_{k=1}^{N_s} \lambda_{r^-,k} = 0; \quad (8)$$

$$\frac{\partial L}{\partial \xi_{c,i}} = C_1 - \lambda_{c,i} - \eta_{c,i} = 0, \quad i = 1, 2, \dots, N_t; \quad (9)$$

$$\begin{aligned} \frac{\partial L}{\partial \xi_{r^+,k}} = C_2 - \lambda_{r^+,k} - \eta_{r^+,k} = 0, \\ k = 1, 2, \dots, N_s; \quad (10) \end{aligned}$$

$$\begin{aligned} \frac{\partial L}{\partial \xi_{r^-,k}} = C_2 - \lambda_{r^-,k} - \eta_{r^-,k} = 0, \\ k = 1, 2, \dots, N_s. \quad (11) \end{aligned}$$

将式 (7)~(11) 代入 (6), 得到原始问题的对偶形式<sup>[16]</sup>

$$\begin{aligned} \min(-L) = \frac{1}{2} \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \lambda_{c,i} \lambda_{c,j} y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \\ \frac{1}{2} \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \lambda_{r^+,i} \lambda_{r^+,j} \mathbf{x}_{s,i}^T \mathbf{x}_{s,j} + \\ \frac{1}{2} \sum_{k=1}^{N_s} \sum_{j=1}^{N_s} \lambda_{r^-,k} \lambda_{r^-,j} \mathbf{x}_{s,k}^T \mathbf{x}_{s,j} - \\ \sum_{i=1}^{N_t} \sum_{k=1}^{N_s} \lambda_{c,i} \lambda_{r^+,k} y_i \mathbf{x}_i^T \mathbf{x}_{s,k} - \\ \sum_{k=1}^{N_s} \sum_{j=1}^{N_s} \lambda_{r^+,k} \lambda_{r^-,j} \mathbf{x}_{s,k}^T \mathbf{x}_{s,j} + \\ \sum_{i=1}^{N_t} \sum_{k=1}^{N_s} \lambda_{c,i} \lambda_{r^-,k} y_i \mathbf{x}_i^T \mathbf{x}_{s,k} - \\ \sum_{i=1}^{N_t} \lambda_{c,i} + \sum_{k=1}^{N_s} \lambda_{r^+,k} (y_{s,k} + \varepsilon) - \\ \sum_{k=1}^{N_s} \lambda_{r^-,k} (y_{s,k} - \varepsilon). \quad (12) \end{aligned}$$

显然, 对偶问题是一个 QP 问题, 求得最优解后, 可由式 (10) 求得

$$\mathbf{w} = \sum_{i=1}^{N_t} \lambda_{c,i} y_i \mathbf{x}_i - \sum_{k=1}^{N_s} \lambda_{r^+,k} \mathbf{x}_{s,k} + \sum_{k=1}^{N_s} \lambda_{r^-,k} \mathbf{x}_{s,k}. \quad (13)$$

### 1.5 问题求解

为了简化式 (15), 定义  $\mathbf{K}_{cc}$  为  $N_t \times N_t$  矩阵,  $\mathbf{K}_{rc}$  为  $N_t \times N_s$  矩阵,  $\mathbf{K}_{cc}$  为  $N_s \times N_s$  矩阵. 其中

$$\begin{aligned} [\mathbf{K}_{cc}]_{i,j} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j, \quad [\mathbf{K}_{rc}]_{i,j} = y_i \mathbf{x}_i^T \mathbf{x}_{s,j}, \\ [\mathbf{K}_{rr}]_{i,j} = \mathbf{x}_{s,i}^T \mathbf{x}_{s,j}. \end{aligned}$$

同时, 令  $\mathbf{1}$  表示所有元素均为 1 的  $N$  维列向量, 令  $\mathbf{Y}_s = [y_{s,1} \ y_{s,2} \ \dots \ y_{s,N_s}]^T$ . 假设

$$\begin{aligned} \mathbf{a} = [\lambda_{c,1}, \dots, \lambda_{c,N_t}, \lambda_{r^+,1}, \dots, \lambda_{r^+,N_s}, \\ \lambda_{r^-,1}, \dots, \lambda_{r^-,N_s}]_{N_t+2N_s}, \quad (14) \end{aligned}$$

$$\mathbf{H} = \begin{bmatrix} \mathbf{K}_{cc} & -\mathbf{K}_{rc} & \mathbf{K}_{rc} \\ -\mathbf{K}'_{rc} & \mathbf{K}_{rr} & -\mathbf{K}_{rr} \\ \mathbf{K}'_{rc} & -\mathbf{K}_{rr} & \mathbf{K}_{rr} \end{bmatrix}_{(N_t+2N_s) \times (N_t+2N_s)}, \quad (15)$$

$$\mathbf{f} = \begin{bmatrix} -\mathbf{I} \\ \mathbf{Y}_S + \varepsilon \\ -\mathbf{Y}_S + \varepsilon \end{bmatrix}_{N_t+2N_s}. \quad (16)$$

简化后, 需要求解的 QP 问题为

$$\begin{aligned} & \min \frac{1}{2} \mathbf{a}^T \mathbf{H} \mathbf{a} + \mathbf{a}^T \mathbf{f}. \\ & \text{s.t.} \quad \sum_{i=1}^{N_t} a_i y_i - \sum_{i=N_t+1}^{N_t+N_s} a_i + \sum_{i=N_t+N_s+1}^{N_t+2N_s} a_i = 0; \\ & \quad 0 \leq a_i \leq C_1, \quad i = 1, 2, \dots, N_t; \\ & \quad 0 \leq a_i \leq C_2, \quad i = N_t + 1, N_t + 2, \dots, N_t + 2N_s. \end{aligned} \quad (17)$$

QP 问题的求解办法已相当成熟, 如 SMO<sup>[18]</sup>等, 本文不再赘述, 利用 QP 求解可最终获取式(3)的最优解. 根据 K.K.T. 定理, 对于偏置项  $b$ , 采用如下的计算方式:

$$b_3 = \varepsilon - y_{s,k} + \mathbf{w}^T \mathbf{x}_{s,k}. \quad (18)$$

## 2 相关工作比较分析

### 2.1 CC-TSVM 与经典 SVM 模型

经典 SVC 分类器的原始优化问题可以定义为

$$\begin{aligned} & \min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \xi_i. \\ & \text{s.t.} \quad y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N; \\ & \quad \xi_i \geq 0, \quad i = 1, 2, \dots, N. \end{aligned} \quad (19)$$

经典的 SVR 目标函数为

$$\begin{aligned} & \min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*). \\ & \text{s.t.} \quad y_i - (\mathbf{w}^T \mathbf{x}_i + b) \leq \varepsilon + \xi_i, \quad i = 1, 2, \dots, N; \\ & \quad (\mathbf{w}^T \mathbf{x}_i + b) - y_i \leq \varepsilon + \xi_i^*, \quad i = 1, 2, \dots, N; \\ & \quad \xi_i \geq 0, \quad \xi_i^* \geq 0, \quad i = 1, 2, \dots, N. \end{aligned} \quad (20)$$

其中:  $\mathbf{x}$  为特征向量,  $y$  为标记值,  $\xi$  为松弛变量,  $C$  为容错惩罚系数,  $\varepsilon$  为常用的不灵敏误差函数. SVC 和 SVR 寻找的超平面可以表示为

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^N a_i y_i \mathbf{x}_i^T \mathbf{x} + b. \quad (21)$$

与 CC-TSVM 分类的目标函数比较, 经典的 SVC 和 SVR 仅考虑了单个数据集的情况, 当有两个数据集时, 若将源域分类器直接作用于目标域数据, 则分类精度会受到很大影响. CC-TSVM 算法的目标函数(如式(3)所示)结合了经典 SVC 和 SVR 的特点, 在 SVC 的基础上, 利用 SVR 原理将源域分类器迁移到目标域数据上, 使得两个分类器的“间隔”最小, 从而完成对目标域数据的分类.

### 2.2 CC-TSVM 与基于 Adaboost 的迁移方法

TrAdaBoost 算法是基于 AdaBoost 算法的迁移学习算法, 在每一次迭代中, TrAdaBoost 算法根据训练

误差, 降低分类错误的旧训练样本在全部样本中的权重. 经过多次迭代后, 与目标域分布一致的源域样本得到较高的权重, 与目标域样本冲突的源域数据具有较低的权重, 从而提高新的分类器对目标域数据的分类精度. 文献[10]证明, TrAdaBoost 算法能够有效地选择出对分类器最有帮助的源域数据, 同时降低与目标域无关的数据对分类器的负面影响. 然而, TrAdaBoost 算法依赖目标域中的已标记数据不断修正分类器, 使得该算法对这部分样本的质量较为敏感. CC-TSVM 算法利用 SVR 理论, 考虑源域和目标域分类器之间的“间隔”, 使得分类器的精度不再依赖某些特定样本.

### 2.3 CC-TSVM 与基于辅助集的迁移算法

基于辅助集的迁移算法需要借助一个与源域和目标域不同的数据集完成迁移, 该算法要求在原来目标函数的基础上, 加入辅助集的损失函数, 并且为了防止过度学习设置一个惩罚函数, 利用相应的参数平衡各部分损失, 所以分类器的精度与辅助集与源域的相似度有很大的关系. 值得一提的是, 这种思想可以推广到其他的学习算法中, 如对 SVM 的目标函数和约束条件中, 增加辅助集的惩罚函数和约束条件,  $K$  均值算法增加辅助集的投票权, 并与源域的投票结果加权. 文献[7]证明, 基于辅助集的迁移算法的分类精度与辅助集的选择有较大关系, 这限制了算法的应用. 然而, CC-TSVM 算法建立在源域数据的基础上, 不需要借助辅助集即可达到迁移学习的目的.

## 3 实验结果和分析

### 3.1 实验参数和实验环境

为了验证 CC-TSVM 算法的有效性, 在不同数据集上对其性能进行评估, 采用的数据集有人造数据集和真实数据集(包括 UCI 数据集)两类. 为了方便实验描述, 对实验中涉及的有关符号给出相应定义, 如表 1 所示.

表 1 实验中涉及的符号定义

符号	定义
$S$	源域训练集
$T$	目标域测试集
$T_d$	目标域训练集(数据量较少)
SVM-S	仅利用源域数据集训练得到目标域的分类器
TSVM	直接利用源域和目标域中已标记数据训练分类器
TrAdaBoost	TrAdaBoost 算法训练得到分类器
TL-SSVM	文献[20]提出的迁移学习分类算法
LMPROJ	文献[19]提出的迁移学习分类算法
CC-TSVM	本文训练的分类器

TL-SSVM 和 LMPROJ 均为迁移学习分类算法, 与之比较, 表明 CC-TSVM 算法在迁移学习问题上具有优势. 实验中, 为了将 CC-TSVM 算法与相关算法

进行性能比较, 采用目标训练集的分类精度作为评价指标, 具体表示为

$$\text{Accuracy} = \frac{|\{\mathbf{x}|\mathbf{x}_t \in T \cap f(\mathbf{x}_t) = Y_T\}|}{|\{\mathbf{x}|\mathbf{x}_t \in T\}|} \quad (22)$$

其中:  $Y_T$  为  $\mathbf{x}_t$  的真实类标,  $f(\mathbf{x}_t)$  为 CC-TSVM 算法所得分类器对  $\mathbf{x}_t$  的分类结果, Accuracy 越高表示分类精度越高.

表 2 算法参数寻优区间

算法	参数
SVM-S	$C$ 的寻优区间为 $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$ ,
TrAdaBoost	$10^1, 10^2, 10^3$ , 高斯核参数 $\delta$ 的寻优
LMPROJ	区间为 $2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3$
TL-SSVM	高斯核参数的寻优区间为 $2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3$ ; $\delta$ 寻优区间为 $\{1, 2, 3, 5, 9, 16, 28, 48\}$ ; $C$ 的寻优区间为 $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$
CC-TSVM	高斯核函数 $\delta$ 的寻优区间为 $\{2^{-3}, 2^{-2}, 2^{-1}, 2^0, 2^1, 2^2, 2^3\}$ ; 正则化参数 $C_1$ 和 $C_2$ 的寻优区间为 $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$

以上所有参数均在训练集上采用五重交叉验证的方式确定最优值, 进而利用该值对测试集作进一步的算法性能分析. 实验环境如下: 实验硬件平台为 Intel Core 2 Duo T8100, 主频为 2.1 GHz, 内存为 1 G, 编程环境为 Matlab 2010 a 等.

### 3.2 模拟数据实验

本次实验采用香蕉数据集, 共 2 类, 每类包括 150 个样本, 原本的数据集作为源域, 目标域中的数据在源域数据的基础上旋转某些角度  $\theta$  得到. 分类情况如图 2 所示.

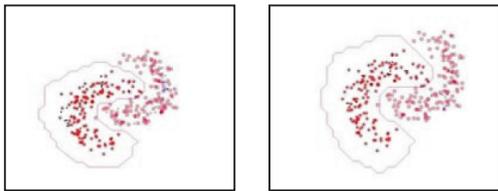


图 2 分类效果

与其他迁移学习算法分类精度比较如表 2 所示, 其中 TL-SSVM 和 TrAdaBoost 算法的分类精度是计算分别占  $T$  总量的 6.67%、10% 和 13.33% 时的 3 次实验的平均分类精度. 由表 3 的模拟数据实验结果可见, CC-TSVM 迁移算法对目标域的分类精度较高, 在目标域不能提供类标信息时, 分类精度依然高于 TSVM 和 TrAdaBoost 迁移算法.

### 3.3 UCI 真实数据实验

实验采用 MAT\_UCI 中 waveform\_21(1 658 个正类样本, 1 648 个负类样本)、ionosphere(126 个正类样本, 225 个负类样本)、spambase(1 813 个正类样本, 2 788 个负类样本)数据集, 对每个数据集的具体处理和分类精度如表 4 所示.

由表 4 可见, CC-TSVM 在 UCI 真实数据集上的实验结果与模拟数据集的性能一致, 依旧高于其他各相关算法, 再次印证了 CC-TSVM 较其他迁移算法更优越的迁移性能, 对于源域和目标域的把控更灵活有效.

### 3.4 实验结果分析

由上述实验可见, CC-TSVM 算法能够针对不同类型的数据分类, 得到较准确的分类精度. 通过实验可以总结关于 CC-TSVM 算法的几个结论:

- 1) CC-TSVM 算法香蕉状的数据能够得到较高的分类精度;
- 2) 只要源域数据与目标域数据之间存在相似, CC-TSVM 算法即可通过这种联系学习到关于目标域数据的信息;
- 3) 源域和目标域间隔参数的取值对分类器的精度影响较大;
- 4) CC-TSVM 算法能够仅利用源域数据信息, 通过补偿源域和目标域之间的间隔, 准确地学习到目标域的信息;

表 3 香蕉数据集实验的分类精度

旋转角度 $\theta$	SVM-S	TrAdaBoost	TL-SSVM	LMPROJ	CC-TSVM
5°	92.00	96.48	55.67	85.67	<b>99.61</b>
10°	95.00	94.78	55.00	86.00	<b>99.33</b>
15°	85.67	90.70	55.00	86.33	<b>98.67</b>

表 4 CC-TSVM 算法在 UCI 数据集上的分类精度

数据集	源域样本数	目标域样本数	SVM-S/%	TrAdaBoost /%	TL-SSVM/%	LMPROJ/%	CC-TSVM/%
ionosphere	160	100	71.25	77.50	60.00	82.10	<b>83.75</b>
	200	200	51.00	59.00	65.50	<b>70.00</b>	69.00
spambase	1 000	400	55.75	61.75	62.25	68.60	<b>80.25</b>
	1 600	600	55.67	61.00	61.50	75.50	<b>82.67</b>
waveform	800	200	51.00	56.50	64.50	51.50	<b>65.00</b>
	1 000	300	50.25	<b>67.50</b>	67.00	51.00	<b>67.50</b>

5) CC-TSVM 算法融合了 SVC 和 SVR 两者的特点, 并且通过与 Tradaboost、LMPROJ、TL-SSVM 等迁移学习分类算法的比较表明了其可靠性.

#### 4 结 论

在迁移学习中, 主要研究如何利用相关邻域的已知信息完成目标数据的学习任务, 这就要求应充分了解旧的邻域与新的邻域之间的相似性关系. 本文考察两个邻域数据的关系, 与 SVR 理论相结合, 提出了 CC-TSVM 迁移学习分类算法. 通过定义容错惩罚参数  $C_1$ 、 $C_2$  和源域与目标域分类器的间隔变量  $\varepsilon$ , 利用 SVR 理论中“间隔”的概念, 找到对目标域分类的最优超平面, 达到迁移学习的目的. 实验表明, CC-TSVM 算法能够保证分类精度.

本文的不足和未来研究的工作是: QP 问题的求解限制了 CC-TSVM 算法的总体速度, 使得在处理较多数据时花费更多的时间; 在今后的工作中, 可以考虑通过提取源域中与目标域更相似的数据进行训练, 抛弃部分对分类器有负作用的源域数据, 从而减少 QP 问题的计算. 另外, CC-TSVM 算法对大样本数据的应用仍然有待证明.

#### 参考文献(References)

- [1] Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Trans on Knowledge and Data Engineering, 2010, 22(10): 1345-1359.
- [2] Caruana R. Multitask learning[J]. Machine Learning, 1997, 28(1): 41-75.
- [3] Raina R, Battle A, Lee H, et al. Self-taught learning: Transfer learning from unlabeled data[C]. Proc of the 24th Int Conf on Machine Learning. Corvalis, 2007: 759-766.
- [4] Daum'Eiii H, Marcu D. Domain adaptation for statistical classifiers[J]. J of Artificial Intelligence Research, 2006, 26: 101-126.
- [5] Zadrozny B. Learning and evaluating classifiers under sample selection bias[C]. Proc of the 21st Int Conf on Machine Learning. Banff, 2004: 114-122.
- [6] Ben-David S, Schuller R. Exploiting task relatedness for multiple task learning[C]. Proc of the 16th Annual Conf on Learning Theory. San Francisco: Morgan Kaufmann, 2003: 825-830.
- [7] Wu P, Dietterich T G. Improving svm accuracy by training on auxiliary data sources[C]. Proc of the 21st Int Conf on Machine Learning. Banff, 2004: 69.
- [8] Liao X, Xue Y, Carin L. Logistic regression with an auxiliary data source[C]. Proc of the 22nd Int Conf on Machine Learning Bonn. Germany, 2005: 505-512.
- [9] Freund Y, Schapire R E. A decision theoretic generalization of on-line learning and an application to boosting[J]. J of Computer and System Sciences, 1997, 55(1): 119-139.
- [10] Dai W, Yang Q, Xue G, et al. Boosting for transfer learning[C]. Proc of the 24th Int Conf on Machine Learning. New York: ACM, 2007: 193-200.
- [11] Quanz B, Huan J. Large margin transductive transfer learning[C]. Proc of the 18th ACM Conf Information and knowledge management. New York: ACM, 2009: 1327-1336.
- [12] Ablavsky V H, Becker C J, Fua P. Transfer learning by sharing support vectors[R]. No.EPFL\_REPORT\_181360, 2012.
- [13] Ling X, Dai W Y, Xue G R, et al. Spectral domain-transfer learning[C]. Proc of the 14th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. Las Vegas, 2008: 488-496.
- [14] Mihakova L, Mooney R J. Transfer learning by mapping with minimal target data[C]. Proc of the AAAI-2008 Workshop on Transfer Learning for Complex Tasks. Chicago, 2008.
- [15] Davis J, Domingos P. Deep transfer via second-order Markov logic[C]. Proc of the 26th Annual Inte Conf on Machine Learning. ACM, 2009: 217-224.
- [16] Vapnik V, Golowich S, Smola A. Support vector method for function approximation, regression estimation and signal processing[C]. Neural Information Processing Systems. Cambridge: MIT Press, 1997: 281-287.
- [17] Wang Z, Song Y, Zhang C. Transferred dimensionality rednction[M]. Machine Learning and Knowledge Discovery in Databases. Berlin: Springer, 2008: 550-565.
- [18] Body S, Vandenberghe L. Convex optimization[M]. Cambridge: Cambridge University Press, 2004.

(责任编辑: 郑晓蕾)