基于张量分解的域适应算法

徐书艳 韩立新 徐国夏
(河海大学计算机与信息学院 南京 211100)

摘 要 由于训练数据易过期,在多数情况下训练数据和测试数据具有不同的特征分布,因此在利用源域信息时,须先尽量减小不同领域的特征分布的差异。使用张量表示特征可以维持高维空间数据的本征结构信息。朴素张量子空间学习法虽然是面向张量特征的域适应方法,但其复杂度较高,且没有达到较好的知识迁移效果。为此,文中提出了基于张量分解的域适应算法,即张量列子空间学习法和张量环子空间学习法,二者的主要思想相似。首先,使用张量表示源域和目标域的特征;其次利用张量分解方法,将特征分解为一系列三阶张量来表示子空间;然后,依次将源域特征和目标域特征映射到子空间中;最后,将特征张量重塑为矩阵形式,基于映射后的源域特征训练模型,基于映射后的目标域特征完成新领域的任务。实验结果表明,在无监督图像分类中,张量列子空间学习法和张量环子空间学习法在准确率和运行时间方面都有所提升。相比于朴素张量子空间学习法,张量列子空间学习法和张量环子空间学习法的准确率分别提高了 1.68%和 2.08%,且运行时间也有明显减少,算法复杂度较小。实验数据充分说明,基于张量分解的域适应算法充分减小了源域特征和目标域特征之间的差异,实现了不同领域间的知识复用。

关键词 迁移学习,域适应,张量分解,子空间学习,图像分类
中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/jsjkx.190300095

Domain Adaptation Algorithm Based on Tensor Decomposition

XU Shu-yan HAN Li-xin XU Guo-xia
(College of Computer and Information, Hohai University, Nanjing 211100, China)

Abstract Because training data tend to be outdated, training data and test data have different feature distributions in most cases. Therefore, when using the source domain information, it is necessary to minimize the difference of feature distributions in different fields. Features represented by tensor can maintain the intrinsic structure information of high-dimensional spatial data. Naive tensor subspace learning is a domain adaptation method for tensor features, but it has high complexity and can not achieve good knowledge transfer effect. For this reason, this paper proposed a domain adaptation algorithm based on tensor decomposition, namely tensor train subspace learning and tensor ring subspace learning, and the main ideas of the two methods are similar. Firstly, the features of source domain and target domain are coded into tensor. By using the tensor decomposition, the tensor of features is decomposed into a series of third-order tensors to represent the subspace. Then, the features of source domain and target domain are mapped into subspace successively. Finally, the feature tensor is reshaped into matrix form. Based on the mapped features of source domain training model and the mapped feature of target domain, the task in new domain is completed. Experiments show that the tensor train subspace learning and the tensor ring subspace learning are improved in terms of accuracy and running time for unsupervised image classification. Compared with the naive tensor subspace learning, the accuracy of the tensor train subspace learning and the tensor ring subspace learning is improved by 1.68% and 2.08% respectively, the running time is also reduced significantly, and the complexity of the algorithm is smaller. Experimental results show that the domain adaptation algorithm based on tensor decomposition can reduce the difference between source domain and target domain, and realize the reuse of knowledge between different domains.

Keywords Transfer learning, Domain adaptation, Tensor decomposition, Subspace learning, Image classification

到稿日期:2019-03-20 返修日期:2019-07-14
徐书艳(1995—),女,硕士,主要研究方向为迁移学习、计算机视觉;**韩立新**(1967—),男,博士,教授,主要研究方向为信息检索、数据挖掘、模式识别,E-mail:lixinhan2002@aliyun.com(通信作者);**徐国夏**(1994—),男,硕士,主要研究方向为迁移学习、计算机视觉。

1 引言

已有的很多数据挖掘和机器学习方法都有一个基本假设:训练数据和测试数据必须服从相同的分布[1]。由于训练数据易过期,因此在多数情况下这种同分布假设并不能得到满足。迁移学习放宽了传统机器学习中训练数据和测试数据必须服从独立同分布的约束,因而能够在彼此不同但又相互关联的两个领域间挖掘领域不变的本质特征和结构,使得标注数据等有监督信息可以在领域间实现迁移和复用[2]。因此,伴随着最近几年的机器学习热潮,迁移学习成为了目前最炙手可热的研究方向之一。

域适应是迁移学习领域中的研究热点[3-5],它研究的主要问题是:假设源域和目标域的类别空间与特征空间相同,但是数据分布不同,如何利用有标注的源域数据来标注目标域数据。针对单源域-单目标域的域适应学习,目前已有的方法可分为 3 类[6]:基于样本的域适应方法、基于特征表示的域适应方法和基于模型的域适应方法。其中,基于特征表示的域适应方法最为热门。从特征表示形式的角度可以发现,现有的大多数域适应方法只适用于向量,用这些方法表示高维数据时,须先将数据向量化,这严重破坏了高维数据的本征结构;而且在表示高维数据时,向量表示会导致由大量参数估计带来的误差和计算复杂度高等问题。

2017 年,Lu 等[7]首次将张量应用于域适应,并结合子空间学习法提出朴素张量子空间学习法(Naive Tensor Subspace Learning,NTSL),使用张量表示源域和目标域特征,使用 Tucker 分解[8]将特征张量分解为核张量和一系列因子矩阵,这一系列矩阵就可以表示源域和目标域的共享子空间。但是,基于 Tucker 分解的域适应方法不仅复杂度过高,而且削弱了子空间的全局表示,并没有达到较好的知识迁移效果。

基于以上分析,本文提出了区别于 NTSL 的基于张量分解的域适应算法。首先提出了张量列子空间学习法(Tensor Train Subspace Learning,TTSL),其使用张量表示源域和目标域的特征,然后将特征张量进行张量列分解,并利用分解结果构造子空间,依次将源域和目标域特征映射到子空间中,最后基于映射后的源域特征训练模型,基于映射后的目标域特征完成新领域的任务。随后又提出了张量环子空间学习法(Tensor Ring Subspace Learning,TRSL),该算法与 TTSL 算法的思想和流程相似,但由于张量环分解自身的优势和映射过程中张量运算的灵活运用,基于张量环分解的域适应算法更高效。

2 相关工作

2.1 张量列分解

张量分解[9]的目的是通过对潜在因子的多线性运算来表示一个高阶张量数据。张量网络[10-11]是张量分解的推广,即将一个高阶张量表示为相互连通的低阶张量,其已成为分析大规模张量数据的有力工具。

张量列分解[12]是张量网络的基石,在其他领域也被称为矩阵乘积态(MPS)或线性张量网络(LTN)[13]。它主要是基于一个连续的奇异值分解序列,实现起来非常简单。由于可以绕过局部交替优化得到全局最优矩阵,张量列分解具有较好的分类性能和较低的计算成本[14]。

具体来说,张量 $\boldsymbol{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_d}$ 经过张量列分解后,$\boldsymbol{X}$ 中的每一个元素都可以用一系列的三阶张量的乘积来近似表示,即:

$$\boldsymbol{X}(i_1,i_2,\cdots,i_d)=\boldsymbol{U}_1(i_1)\boldsymbol{U}_2(i_2)\cdots\boldsymbol{U}_d(i_d) \tag{1}$$

其中,$\boldsymbol{U}_k$ 是三阶张量,在张量列分解中也叫做核张量,维度为 $r_{k-1}\times I_k\times r_k$;$\boldsymbol{U}_k(i_k)$ 是指定 U_k 的第 i_k 个侧面切片矩阵,则 $\boldsymbol{U}_k(i_k)$ 的维度为 $r_{k-1}\times r_k$。这些维度相关的矩阵相乘的结果是维度为 $r_0\times r_d$ 的矩阵,而式(1)中等号左边 $\boldsymbol{X}(i_1,i_2,\cdots,i_d)$ 表示 $\boldsymbol{X}$ 中的一个元素值,因此在张量列分解中存在边界条件:$r_0=r_d=1$。

在索引形式中,张量列分解可写为:

$$\boldsymbol{X}(i_1,i_2,\cdots,i_d)=\sum_{\alpha_0,\alpha_1,\cdots,\alpha_d}^{r_0,r_1,\cdots,r_d}\prod_{k=1}^{d}\boldsymbol{U}_k(\alpha_{k-1},i_k,\alpha_k) \tag{2}$$

张量列分解是奇异值分解(SVD)的一种延伸,其中的核张量可通过分解张量的展开矩阵得到。Bengua 等[14]提出了基于 MPS 的张量特征提取算法。对于 n 个 d 阶的张量,该算法先将所有张量拼接为一个 $d+1$ 阶张量,记为 X;然后为了降低分解损失,重新排列 $\boldsymbol{X}$ 的维度,使其满足 $\boldsymbol{X}\in\mathbb{R}^{I_1\times\cdots\times I_{m-1}\times n\times I_m\times\cdots\times I_d}$,其中 $I_1\geqslant\cdots\geqslant I_{m-1}$,$I_m\leqslant\cdots\leqslant I_d$;接着从 $\boldsymbol{X}$ 的第一阶模展开矩阵开始进行奇异值分解。详细过程如下:

1)根据每次分解后的 $\boldsymbol{U}$,得到核张量$\{\boldsymbol{U}_1,\cdots,\boldsymbol{U}_{m-1}\}$,$\boldsymbol{U}_k\in\mathbb{R}^{r_{k-1}\times I_k\times r_k}$,将 $\boldsymbol{S}$ 与 $\boldsymbol{V}^{\mathrm{T}}$ 相乘,得到要进行下次奇异值分解的矩阵。

2)得到 $\boldsymbol{U}_{m-1}$ 后,根据每次分解后的 $\boldsymbol{V}$,依次得到核张量$\{\boldsymbol{U}_d,\cdots,\boldsymbol{U}_m\}$,$\boldsymbol{U}_k\in\mathbb{R}^{r_k\times I_k\times r_{k+1}}$,将 $\boldsymbol{U}$ 与 $\boldsymbol{S}$ 相乘,得到要进行下次奇异值分解的矩阵。

3)根据 r_{m-1},n,r_m 重塑最后得到的 $\boldsymbol{U}$ 与 $\boldsymbol{S}$ 的乘积,从而得到第 $d+1$ 个核张量。最终,$\boldsymbol{X}$ 中的每一个元素都可以用这 $d+1$ 个核张量的乘积来近似表示。

张量列分解要求核张量序列的首尾维度为 1,即 $r_0=r_d=1$,而且核张量的多线性乘积必须遵循严格的顺序关系,因此张量列分解方法存在表示能力和灵活性受限的问题[15]。

2.2 张量环分解

张量环分解[15-16]可以看作是张量列分解的线性组合,具有强大的广义表示能力。张量环分解的目的是使用循环相乘的三阶张量序列来表示一个高阶张量。

具体来说,张量 $\boldsymbol{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_d}$ 经过张量环分解后,其中的每一个元素都可以通过对核张量的多线性乘积进行迹运算来近似表示,即:

$$\begin{aligned}\boldsymbol{X}(i_1,i_2,\cdots,i_d)&=\mathrm{Tr}\{\boldsymbol{U}_1(i_1)\boldsymbol{U}_2(i_2)\cdots\boldsymbol{U}_d(i_d)\}\\&=\mathrm{Tr}\{\prod_{k=1}^{d}\boldsymbol{U}_k(i_k)\}\end{aligned} \tag{3}$$

与式(1)相同,核张量 $\boldsymbol{U}_k$ 是三阶张量,维度为 $r_{k-1}\times I_k\times r_k$,$\boldsymbol{U}_k(i_k)$ 是指定 $\boldsymbol{U}_k$ 的第 i_k 个侧面切片矩阵,则 $\boldsymbol{U}_k(i_k)$ 的维度为 $r_{k-1}\times r_k$。相邻的两个核张量 $\boldsymbol{U}_{k-1}(i_{k-1})$ 和 $\boldsymbol{U}_k(i_k)$ 有一个相同的维数 r_{k-1},最后一个核张量与第一个核张量也有相同的维度 r_d,即 $r_0=r_d$。因此,这些核张量的乘积是一个大小为 $r_d\times r_d$ 的方阵。可见,张量 $\boldsymbol{X}$ 中的每一个元素 $\boldsymbol{X}(i_1,i_2,\cdots,i_d)$ 等于矩阵$\{\boldsymbol{U}_k(i_k)\}$序列乘积的迹。

在索引形式中,张量环分解表示张量 $\boldsymbol{X}$ 中的每一个元素

$\boldsymbol{X}(i_1,i_2,\cdots,i_d)$与式(2)相同。

Wang 等[17]提出张量环近似算法(TRA)，其使用随机数填充的方法将张量列分解结果扩展到我们所期望的维度，以此得到相似的张量环分解结果。详细过程如下：

1)首先设定张量环分解的秩 $r_0,r_1,\cdots,r_d$，输入 $\boldsymbol{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_d}$。

2)从 $\boldsymbol{X}$ 的第一阶模展开矩阵开始进行 $d-1$ 次奇异值分解，奇异值个数为 $r_k,t_{k-1}I_k,I_{k+1},\cdots,I_d$ 中的最小值，记为 t_k，其中 $t_0=1$，每次分解得到 $\boldsymbol{U},\boldsymbol{S},\boldsymbol{V}$。将 $\boldsymbol{S}$ 与 $\boldsymbol{V}^{\mathrm{T}}$ 相乘，得到下次要进行奇异值分解的矩阵。重塑 $\boldsymbol{U}_k$，使其维度变为 $t_{k-1}\times I_k\times t_k$，用随机数填充 $\boldsymbol{U}_k$，使得 $\boldsymbol{U}_k\in\mathbb{R}^{r_{k-1}\times I_k\times r_k}$。

3)得到 $d-1$ 个三阶张量后，将上一轮得到的 $\boldsymbol{S}$ 与 $\boldsymbol{V}^{\mathrm{T}}$ 相乘，得到 $\boldsymbol{U}_d$，此时 $\boldsymbol{U}_d\in\mathbb{R}^{t_{d-1}\times I_k}$。用随机数填充 $\boldsymbol{U}_d$，使得 $\boldsymbol{U}_d\in\mathbb{R}^{r_{d-1}\times I_d\times r_d}$。

4)最终得到 d 个三阶张量近似 $\boldsymbol{X}$ 的张量环分解结果。

相较于张量列分解，张量环分解通过使用张量的迹运算，放宽了对秩的约束，无须要求 $r_0=r_d=1$，只需 $r_0=r_d$ 即可，并且表示能力得到增强；核张量可以进行循环移位和等效处理[15]。

2.3　子空间学习法

子空间学习法[18]是域适应方法中的一类，其主要思想是：假设源域数据和目标域数据经过特征变换到子空间后，两者的分布相似。

按照表示特征的方式，可将子空间学习法分为两类：面向向量特征的子空间学习法和面向张量特征的子空间学习法。

现有的大多数子空间学习法都是使用向量表示一个样本的特征。文献[18]提出基于无监督子空间对齐的域适应算法，将每个域的数据映射到各自的子空间中，然后通过一个线性变换将数据的统计特征进行变换对齐。当源域和目标域不太相似时，可采用基于流形变换的流形学习方法[19]，因为此时若强行提取一些共同的潜在因素，则可能导致目标域聚类结构与源域判别结构之间的不一致。因此，建议保留每个领域的几何结构，将特征变换到流行空间中[20]。在流形空间中的特征通常有很好的几何性质，可以避免特征扭曲。

2017 年，面向张量特征的子空间方法[7]首次被提出。实验证明，张量表示与子空间学习法的融合，可以取得良好的域适应效果。Lu 等提出使用张量表示源域和目标域的特征，并使用 Tucker 分解[8]将特征张量分解为核张量和一系列因子矩阵，如式(4)所示：

$$\boldsymbol{X}=\boldsymbol{G}\times_1\boldsymbol{U}_1\times_2\boldsymbol{U}_2\times_3\cdots\times_d\boldsymbol{U}_d \tag{4}$$

其中，$\boldsymbol{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_d}$ 为源域和目标域特征拼接后的特征张量，$\boldsymbol{G}\in\mathbb{R}^{r_1\times r_2\times\cdots\times r_d}$ 和 $\boldsymbol{U}_k\in\mathbb{R}^{I_k\times r_k}$ 分别为特征张量 $\boldsymbol{X}$ 经过 Tucker 分解后得到的核心张量和因子矩阵。这一系列因子矩阵就可以表示源域和目标域的共享子空间。由于因子矩阵是正交的，即 $\boldsymbol{U}_k^{\mathrm{T}}\boldsymbol{U}_k=\boldsymbol{I}$，$\boldsymbol{I}$ 为单位矩阵，因此特征映射结果可表示为原特征张量与因子矩阵转置的模积。源域特征的映射结果如式(5)所示：

$$\boldsymbol{G}_S=\boldsymbol{X}_S\times_1\boldsymbol{U}_1^{\mathrm{T}}\times_2\boldsymbol{U}_2^{\mathrm{T}}\times_3\cdots\times_d\boldsymbol{U}_d^{\mathrm{T}} \tag{5}$$

其中，$\boldsymbol{G}_S\in\mathbb{R}^{r_1\times r_2\times\cdots\times r_d}$ 为子空间中的源域特征表示，$\boldsymbol{X}_S\in\mathbb{R}^{I_1\times I_2\times\cdots\times n_s}$ 为源域特征，映射过程较为直观和简洁。

在构造子空间时，基于 Tucker 分解的 NTSL 算法的复杂度较高，而且使用因子矩阵表示子空间，削弱了子空间的全局表示，不能很好地缓解源域数据和目标域数据间分布差异带来的问题，且没有达到较好的知识迁移效果。

3　基于张量分解的域适应

受到 NTSL 算法的启发，我们提出了区别于 NTSL 的基于张量分解的域适应算法，分别是张量列子空间学习法 TTSL 和张量环子空间学习法 TRSL。二者在子空间表示和特征映射部分有所不同，主要思想均是：使用张量表示源域和目标域的特征；将特征张量进行张量分解，并利用分解出的一系列三阶张量构造子空间；依次将源域和目标域特征映射到子空间中；基于映射后的源域特征训练模型，基于映射后的目标域特征完成新领域的任务。

3.1　张量列子空间学习法

首先使用张量表示源域和目标域的特征，分别记为 $\boldsymbol{X}_S$ 和 $\boldsymbol{X}_T$。拼接 $\boldsymbol{X}_S$ 和 $\boldsymbol{X}_T$，得到 $\boldsymbol{X}$。

根据文献[14]提出的基于 MPS 的张量特征提取算法，对 $\boldsymbol{X}$ 进行张量列分解。首先，为了降低分解损失，对 $\boldsymbol{X}$ 的维度进行重新排列，假设源域样本数量和目标域样本数量分别为 n_S 和 n_T，$n=n_S+n_T$，重新排列 $\boldsymbol{X}$ 的维度，使其满足 $\boldsymbol{X}\in\mathbb{R}^{I_1\times\cdots\times I_{m-1}\times n\times I_m\times\cdots\times I_d}$，其中 $I_1\geqslant\cdots\geqslant I_{m-1}$，$I_m\leqslant\cdots\leqslant I_d$；然后进行张量列分解，从 $\boldsymbol{X}$ 的第一阶模展开矩阵开始进行奇异值分解；最后得到 $d+1$ 个核张量来近似 $\boldsymbol{X}$。这些核张量满足以下公式：

$$\min_{\boldsymbol{U}_1,\boldsymbol{U}_2,\cdots,\boldsymbol{U}_{d+1}}\|f(\boldsymbol{U}_1\cdots\boldsymbol{U}_{m-1}\boldsymbol{U}_{d+1}\boldsymbol{U}_m\cdots\boldsymbol{U}_d)-\boldsymbol{X}\|_F^2 \tag{6}$$

其中，函数 f[17]表示将 $\boldsymbol{U}_1,\boldsymbol{U}_2,\cdots,\boldsymbol{U}_{d+1}$ 相乘的结果重塑为维度为 $I_1\times\cdots\times I_{m-1}\times n\times I_m\times\cdots\times I_d$ 的 $d+1$ 阶张量，前 d 个与 $I_1,I_2\cdots,I_d$ 相关的核张量$\{\boldsymbol{U}_1,\boldsymbol{U}_2,\cdots,\boldsymbol{U}_d\}$表示子空间。

依次将源域特征和目标域特征映射到子空间中。对 $\boldsymbol{X}_S$ 和 $\boldsymbol{X}_T$ 进行与 $\boldsymbol{X}$ 相同的维度转换，得到 $\boldsymbol{X}_S\in\mathbb{R}^{I_1\times\cdots\times I_{m-1}\times n_S\times I_m\times\cdots\times I_d}$，$\boldsymbol{X}_T\in\mathbb{R}^{I_1\times\cdots\times I_{m-1}\times n_T\times I_m\times\cdots\times I_d}$；然后分别将 $\boldsymbol{X}_S$ 和 $\boldsymbol{X}_T$ 与$\{\boldsymbol{U}_1,\cdots,\boldsymbol{U}_d\}$进行结合，此过程须利用对张量维度的调整和重塑来完成特征的变换，同时 $\boldsymbol{X}_S$ 和 $\boldsymbol{X}_T$ 降阶为三阶张量，此时 $\boldsymbol{X}_S\in\mathbb{R}^{r_{m-1}\times n_S\times r_m}$，$\boldsymbol{X}_T\in\mathbb{R}^{r_{m-1}\times n_T\times r_m}$。

将原 $\boldsymbol{X}_S$ 的第 k 阶元素映射到子空间，$k=1,2,\cdots,m-1$ 时，$\boldsymbol{X}_S\in\mathbb{R}^{(r_{k-1}\times I_k)\times I_{k+1}\times\cdots\times I_{m-1}\times n_S\times I_m\times\cdots\times I_d}$，为 $d-k+2$ 阶张量，并重塑三阶张量 $\boldsymbol{U}_k$ 为矩阵，$\boldsymbol{U}_k\in\mathbb{R}^{(r_{k-1}\times I_k)\times r_k}$，更新 $\boldsymbol{X}_S$ 为 $\boldsymbol{U}_k$ 与 $\boldsymbol{X}_S$ 的第一阶模展开矩阵的乘积，其元素如式(7)所示；$k=m,m+1,\cdots,d$ 时，$\boldsymbol{X}_S\in\mathbb{R}^{r_{m-1}\times n_S\times I_m\times\cdots\times I_{k-1}\times(I_k\times r_{k+1})}$，为 $k-m+3$ 阶张量，并重塑三阶张量 $\boldsymbol{U}_k$ 为矩阵，$\boldsymbol{U}_k\in\mathbb{R}^{r_k\times(I_k\times r_{k+1})}$，更新 $\boldsymbol{X}_S$ 为 $\boldsymbol{U}_k$ 与 $\boldsymbol{X}_S$ 的最后一阶模展开矩阵的乘积，其元素如式(8)所示。

$$\begin{aligned}\boldsymbol{X}_{S(j,i_2,\cdots,i_{d-k+2})}&=(\boldsymbol{X}_S\times_1\boldsymbol{U}_k)_{(j,i_2,\cdots,i_{d-k+2})}\\&=\sum_{i_1=1}^{r_{k-1}I_k}\boldsymbol{X}_{S(i_1,i_2,\cdots,i_{d-k+2})}\boldsymbol{U}_{(i_1,j)}\end{aligned} \tag{7}$$

$$\begin{aligned}\boldsymbol{X}_{S(i_1,\cdots,i_{k-m+2},j)}&=(\boldsymbol{X}_S\times_{k-m+3}\boldsymbol{U}_k)_{(i_1,\cdots,i_{k-m+2},j)}\\&=\sum_{i_1=1}^{I_kr_{k+1}}\boldsymbol{X}_{S(i_1,i_2\cdots,i_{k-m+3})}\boldsymbol{U}_{(j,i_{k-m+3})}\end{aligned} \tag{8}$$

其中，$j=1,2,\cdots,r_k$。

$\boldsymbol{X}_S$ 与$\{\boldsymbol{U}_1,\cdots,\boldsymbol{U}_d\}$相乘时，$\boldsymbol{X}_S$ 的降阶过程，以及 $\boldsymbol{X}_S$ 与 $\boldsymbol{U}_k$

的具体维度如表 1 所列；将 $\boldsymbol{X}_T$ 与 $\{\boldsymbol{U}_1,\cdots,\boldsymbol{U}_d\}$ 进行结合的过程与此相同。

表 1 $\boldsymbol{X}_S$ 与 $\boldsymbol{U}_k$ 的维度变换

Table 1 Dimensional transformation of $\boldsymbol{X}_S$ and $\boldsymbol{U}_k$

$\boldsymbol{X}_S$ 的维度	$\boldsymbol{U}_k$ 的维度
$I_1\times\cdots\times I_{m-1}\times n_S\times I_m\times\cdots\times I_d$	$r_0\times I_1\times r_1(r_0=1,k=1)$
↓	
$(I_2\times\cdots I_{m-1}\times n_S\times I_m\times\cdots I_d)\times I_1$	$I_1\times r_1(k=1)$
↓	
$(I_3\times\cdots I_{m-1}\times n_S\times I_m\times\cdots I_d)\times r_1 I_2$	$r_1 I_2\times r_2(k=2)$
⋮	⋮
$(n_S\times I_m\times\cdots\times I_d)\times r_{m-2}I_{m-1}$	$r_{m-2}I_{m-1}\times r_{m-1}(k=m-1)$
↓	
$(r_{m-1}\times n_S\times I_m\times\cdots\times I_{d-1})\times I_d$	$I_d r_{d+1}\times r_d(r_{d+1}=1,k=d)$
↓	
$(r_{m-1}\times n_S\times I_m\times\cdots\times I_{d-2})\times I_{d-1}r_d$	$I_{d-1}r_d\times r_{d-1}(k=d-1)$
⋮	⋮
$r_{m-1}n_S\times I_m r_{m+1}$	$I_m r_{m+1}\times r_m(k=m)$
↓	
$r_{m-1}\times n_S\times r_m$	

重塑 $\boldsymbol{X}_S$ 和 $\boldsymbol{X}_T$ 的维度，使之用一个向量表示一个样本的所有特征，即 $\boldsymbol{X}_S\in\mathbb{R}^{(r_{m-1}\times r_m)\times n_S}$，$\boldsymbol{X}_T\in\mathbb{R}^{(r_{m-1}\times r_m)\times n_T}$。根据映射后的源域特征 $\boldsymbol{X}_S$ 训练模型，根据映射后的目标域特征 $\boldsymbol{X}_T$ 完成新领域的任务。

3.2 张量环子空间学习法

本文提出的张量环子空间学习法，利用样本特征的张量环分解结果构造子空间，在特征映射过程中，可根据特征张量的具体维度，选择从哪一阶模开始与核张量进行结合。

根据文献[17]提出的 TRA 算法，得到样本特征 $\boldsymbol{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_d}$ 的张量环分解结果，其中 $I_d=n_S+n_T$。首先，设定张量环分解的秩 $r_0,r_1,\cdots,r_d$，其中 $r_0=r_d$。然后，对 $\boldsymbol{X}$ 进行张量列分解，得到 $\{\boldsymbol{U}_1,\boldsymbol{U}_2,\cdots,\boldsymbol{U}_d\}$，此时并不满足 $\boldsymbol{U}_k\in\mathbb{R}^{r_{k-1}\times I_k\times r_k}$，因此用随机数填充其余项，使 $\boldsymbol{U}_k$ 扩展为我们所期望的维度。

由于张量环的分解结果可以进行循环移位，在进行特征映射时，可从 $\boldsymbol{X}_S$ 和 $\boldsymbol{X}_T$ 的第一阶模开始，从左往右依次与 $\boldsymbol{U}_1,\boldsymbol{U}_2,\cdots,\boldsymbol{U}_{d-1}$ 结合；也可以从第 $d-1$ 阶模开始，从右往左依次与 $\boldsymbol{U}_{d-1},\boldsymbol{U}_{d-2},\cdots,\boldsymbol{U}_1$ 结合。根据具体的样本特征的维度选择特征映射的方向，可使复杂度大大减小。逆序映射时，张量环子空间学习法的具体步骤如算法 1 所示。

算法 1 张量环子空间学习法

输入：源域特征 $\mathbf{X}_S\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_{d-1}\times n_S}$；目标域特征 $\mathbf{X}_T\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_{d-1}\times n_T}$；张量环分解的秩 $r_0,r_1,\cdots,r_d$

输出：子空间中的源域特征表示 $\mathbf{X}_S\in\mathbb{R}^{n_S\times r_{d-1}\times r_d}$，目标域特征表示 $\mathbf{X}_T\in\mathbb{R}^{n_T\times r_{d-1}\times r_d}$

Step 1 拼接 $\mathbf{X}_S$ 与 $\mathbf{X}_T$，得到所有域特征 $\mathbf{X}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_d}$，其中 $I_d=n_S+n_T$。

Step 2 计算 $\mathbf{X}$ 的第一阶模展开矩阵，并将其作为进行第一次奇异值分解的矩阵，记为 $\mathbf{T}_{\langle1\rangle}$；奇异值个数 $t_1=\min(r_1,I_1,I_2,\cdots,I_d)$；对 $\mathbf{T}_{\langle1\rangle}$ 进行奇异值分解：$\mathbf{T}_{\langle1\rangle}=\mathbf{U}_1\mathbf{S}_1\mathbf{V}_1^{\mathrm{T}}$，其中 $\mathbf{U}_1\in\mathbb{R}^{I_1\times t_1}$，$\mathbf{S}_1\in\mathbb{R}^{t_1\times t_1}$，$\mathbf{V}_1\in\mathbb{R}^{(I_2\times I_3\times\cdots\times I_d)\times t_1}$。

Step 3 for k=2,3,⋯,d−1 do

$\mathbf{T}_{\langle k\rangle}=\mathrm{reshape}(\mathbf{S}_{k-1}\mathbf{V}_{k-1}^{\mathrm{T}},[t_{k-1}I_k,I_{k+1},\cdots,I_d])$

$t_k=\min(r_k,t_{k-1}I_k,I_{k+1},\cdots,I_d)$

SVD：$\mathbf{T}_{\langle k\rangle}=\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^{\mathrm{T}}$，其中 $\mathbf{U}_k\in\mathbb{R}^{I_k\times t_k}$，$\mathbf{S}_k\in\mathbb{R}^{t_k\times t_k}$，$\mathbf{V}_k\in\mathbb{R}^{(I_{k+1}\times I_{k+2}\times\cdots\times I_d)\times t_k}$

end

Step 4 $\mathbf{U}_d=\mathbf{S}_{d-1}\mathbf{V}_{d-1}^{\mathrm{T}}$。

Step 5 $\mathbf{U}_k=\mathrm{reshape}(\mathbf{U}_k,[t_{k-1},I_k,t_k])$；用随机数填充，使得 $\mathbf{U}_k\in\mathbb{R}^{r_{k-1}\times I_k\times r_k}$。

Step 6 $\mathbf{U}_{d-1}=\mathrm{reshape}(\mathrm{permute}(\mathbf{U}_{d-1},[2,1,3]),[I_{d-1},r_{d-2}r_{d-1}])$；将 $\mathbf{X}_S$ 的第 d−1 阶置换到第 d 阶，重塑 $\mathbf{X}_S$ 的维度为 $(I_1\times I_2\times\cdots\times I_{d-2}\times n_S)\times I_{d-1}$；$\mathbf{X}_S=\mathbf{X}_S\mathbf{U}_{d-1}$；重塑 $\mathbf{X}_S$ 的维度为 $I_1\times I_2\times\cdots\times I_{d-2}\times n_S\times r_{d-1}\times r_{d-2}$。

Step 7 for k=d−2,d−3,⋯,1 do

$\mathbf{U}_k=\mathrm{reshape}(\mathrm{permute}(\mathbf{U}_k,[2,3,1]),[I_k r_k,r_{k-1}])$；将 $\mathbf{X}_S$ 的第 k 阶置换到最后，并重塑 $\mathbf{X}_S$ 为 $I_k r_k$ 列的矩阵；$\mathbf{X}_S=\mathbf{X}_S\mathbf{U}_{d-1}$；重塑 $\mathbf{X}_S$ 的维度为 $I_1\times I_2\times\cdots\times I_{k-1}\times n_S\times r_{d-1}\times r_{k-1}$。

end

Step 8 返回 $\mathbf{X}_S$ 和 $\mathbf{X}_T$（二者的特征变换过程相同）。

4 实验与分析

为了直观地展示算法改进的效果，本文进行与文献[7]相同的无监督图像分类实验，基于 Office-Caltech10(OC10)[19] 数据集，采用了相同的数据处理方式和评估方式。将所提算法与传统的领域自适应算法和 NTSL 算法进行对比实验，并对准确率和算法复杂度进行分析。

4.1 实验数据

Office-Caltech10 数据集取自 office 数据集[21]和 Caltech-256[22]数据集中相同的 10 种类别的物体图像，共 2533 张，包含 4 个领域的数据：Amazon(从网上商店下载的图片，简写为 A)、Caltech(加利福尼亚理工学院收集整理的数据集，简写为 C)、DSLR(由数码单反相机拍摄的高分辨率图像，简写为 D)、Webcam(由网络摄像头拍摄的低分辨率图像，简写为 W)。各领域中的背包类别图像如图 1 所示，背景、拍摄角度和图像分辨率等的不同，造成了不同领域中相同类别的物体图像有不同的特征分布，因此这 4 个领域可以组成 12 组源域和目标域。

图 1 各领域中的背包类别图像

Fig. 1 Images of backpack category in each field

4.2　评估方法

依照文献[7]中的特征张量表示方式，将图像输入 VGG-16 网络[23]，对 CONV5_3 层的卷积特征进行空间金字塔池化[24]处理，可得到大小为 6×6×512 的三阶特征张量。在第四阶进行特征张量的拼接，因此源域和目标域特征在映射前是四阶张量，第四阶维数表示样本数量。Amazon 或 Caltech 领域作为源域时，我们将从该领域的每个类别中随机选取 20 个图像特征作为训练样本。DSLR 或 Webcam 领域作为源域时，我们将从该领域的每个类别中随机选取 8 个图像特征作为训练样本。每个领域作为目标域时，取该领域的所有图像特征作为测试样本。

本文通过张量列分解和张量环分解构造子空间，将源域和目标域特征映射到子空间后得到的特征为三阶张量，将特征重塑为矩阵，即用向量表示一张图像的特征。接着使用支持向量机的一对多法实现多分类，训练时依次将源域样本中的某一类作为正集，其余样本归为负集，共构造了 10 个分类器，分类时将目标域中的无标签样本归为具有最大分类函数值的那一类。

针对每一组源域和目标域，进行 20 次实验，每一次的实验数据都是随机选取的，最后取 20 次实验结果的均值作为该算法的准确率，并记录每次实验的时间。从准确率和运行时间两个角度，来评判域适应算法的迁移效果。

4.3　实验结果与分析

本文实验平台为 Matlab 2017a，运行环境为 Intel Core i5-8300 CPU，8 GB 内存。

针对无域适应（No Adaptation，DA）、NTSL 以及本文提出的 TTSL 算法和 TRSL 算法，分别进行实验。对于 NTSL 算法，根据文献[7]中的参数将子空间维度设为 6×6×128；在 TTSL 算法中，重新排列 $\boldsymbol{X}$ 的维度时，将样本数量 n 排在第 3 位，奇异值分解阈值为 0.1；在 TRSL 算法中，根据每轮实验的样本维度设置合适的秩向量。各算法的图像分类准确率如表 2 所列。

表 2　各算法在 OC10 数据集上的图像分类准确率

Table 2　Image classification accuracy of each algorithm on OC10 dataset

（单位：%）

源域→目标域	DA	NTSL	TTSL	TRSL
A→C	77.6	78.1	82.2	81.7
A→D	82.6	84.0	85.5	86.5
A→W	73.6	77.9	78.4	80.6
C→A	88.8	89.5	91.2	91.5
C→D	86.4	86.5	89.2	89.9
C→W	79.3	79.6	85.8	85.5
D→A	80.9	87.8	89.3	88.0
D→C	70.4	80.1	78.9	80.2
D→W	91.1	95.8	94.1	95.7
W→A	73.8	85.4	88.2	87.8
W→C	63.1	79.5	82.3	81.3
W→D	94.8	97.8	97.1	98.3
均值	80.20	85.17	86.85	87.25

比较各算法在 12 组域适应实验中的准确率，NTSL 算法占据 1 次第一，本文提出的 TTSL 算法占据 4 次第一，TRSL 算法占据 5 次第一，无域适应过程的 DA 方法的效果明显差于其他 3 种方法。可见，将已有知识迁移到特征分布不同的其他领域时，尽量减小源域和目标域分布的差异是十分必要的。另外，在平均准确率方面，TTSL 算法比 NTSL 算法高 1.68%，TRSL 算法比 NTSL 算法高 2.08%。可见，使用张量列分解和张量环分解的域适应方法能更大程度地减小源域和目标域的差异，从而达到更好的知识迁移效果。

在实验数据与运行环境相同的情况下，使用 Amazon 和 Caltech 领域的数据进行 20 次实验，各算法在每组实验中的运行时间对比如图 2 所示，各算法的平均运行时间如表 3 所列。可以看出，NTSL 算法的平均运行时间明显多于本文提出的 TTSL 算法和 TRSL 算法。

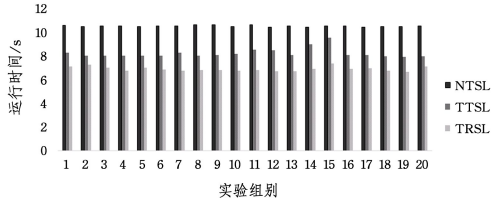

图 2　20 组实验中各算法在 A→C 上的运行时间

Fig. 2　Running time of each algorithm from Amazon to Caltech in 20 groups of experiments

表 3　各算法在 A→C 上的平均运行时间

Table 3　Average running time of each algorithm from Amazon to Caltech

算法	时间/s
NTSL	10.53
TTSL	8.24
TRSL	6.91

分别对各算法的子空间构造和特征映射过程计时，结果如表 4 所列。在 NTSL 算法中，子空间构造过于耗时，因为其用矩阵组合构造子空间，但映射过程较为简便；TRSL 算法的映射过程虽比 TTSL 算法灵活，但根据实验数据可发现 TRSL 算法的映射过程的耗时明显多于 TTSL 算法的耗时。结合张量列分解和张量环分解的结果可知，由于 TTSL 算法中 $r_0=r_d=1$，因此 TTSL 算法的映射过程的耗时较短。

表 4　各算法的子空间构造时间与特征映射时间

Table 4　Subspace construction time and feature mapping time of each algorithm

算法	子空间构造		特征映射	
	时间/s	占比/%	时间/s	占比/%
NTSL	8.56	81.26	0.96	9.16
TTSL	6.00	72.76	0.67	8.08
TRSL	3.86	55.79	1.39	20.14

Tucker 分解通过 1 个核张量和 d 个因子矩阵来近似一个 d 阶张量，需要 $O(dnr+r^d)$ 个参数；张量列分解和张量环分解均是通过 d 个三阶张量来近似一个 d 阶张量，需要 $O(dnr^2)$ 个参数。综合表 3 和表 4 中的实验结果可知，TTSL 算法和 TRSL 算法的复杂度小于 NTSL 算法的复杂度。

文献[7]将 NTSL 算法与许多面向向量特征的经典域适应方法进行对比实验，结果表明 NTSL 算法具有很大的优势。而在无监督图像分类实验中，本文提出的面向张量的 TTSL 算法和 TRSL 算法的准确率高于 NTSL 算法，且算法

复杂度较小。可见,TTSL 算法和 TRSL 算法充分减小了源域和目标域的差异,达到了更好的知识迁移效果。

结束语 针对域适应中特征分布差异的问题,本文提出了基于张量分解的域适应算法:TTSL 算法和 TRSL 算法。使用所有域特征的张量列分解或张量环分解结果构造子空间,将源域特征和目标域特征映射到同一子空间中,以减小二者特征分布的差异。实验结果表明,本文提出的算法优于基于 Tucker 分解的 NTSL 算法和面向向量特征的域适应算法。但本文所提出的算法仍存在两点不足:1)仅将特征映射到子空间中,并没有对特征进行对齐操作以进一步减小源域特征和目标域特征的分布差异;2)在数据量较大时,算法运行中涉及到的张量运算会占用较多资源。未来将考虑在本文算法的基础上加入合理的特征变换方式,将源域特征与目标域特征对齐;同时将更高效的张量环分解算法融入 TRSL 算法中。

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] LONG M S. Transfer Learning:Problems and Methods[D]. Beijing:Tsinghua University,2014. (in Chinese)
龙明盛. 迁移学习问题与方法研究[D]. 北京:清华大学,2014.

[3] ZHANG Z,WANG M,HUANG Y,et al. Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE,2018:3437-3445.

[4] YANG B,MA A J,YUEN P C. Domain-Shared Group-Sparse Dictionary Learning for Unsupervised Domain Adaptation[C]// Thirty-Second AAAI Conference on Artificial Intelligence. AAAI,2018.

[5] MENG J,HU G Y,PAN Z S,et al. Research and Perspective on Domain Adaptation Learning Algorithms[J]. Computer Science,2015,42(10):7-12,34. (in Chinese)
孟娟,胡谷雨,潘志松,等. 领域适应学习算法研究与展望[J]. 计算机科学,2015,42(10):7-12,34.

[6] LIU J W,SUN Z K,LUO X L. Review and Research Development on Domain Adaptation Learning[J]. Acta Automatica Sinica,2014,40(8):1576-1600. (in Chinese)
刘建伟,孙正康,罗雄麟. 域自适应学习研究进展[J]. 自动化学报,2014,40(8):1576-1600.

[7] LU H,ZHANG L,CAO Z,et al. When unsupervised domain adaptation meets tensor representations[C]// Proceedings of the IEEE International Conference on Computer Vision. IEEE,2017:599-608.

[8] TUCKER L R. Some mathematical notes on three-mode factor analysis[J]. Psychometrika,1966,31(3):279-311.

[9] KOLDA T G,BADER B W. Tensor decompositions and applications[J]. SIAM review,2009,51(3):455-500.

[10] ORÚ S R. A practical introduction to tensor networks:Matrix product states and projected entangled pair states[J]. Annals of Physics,2014,349:117-158.

[11] MURG V,VERSTRAETE F,SCHNEIDER R,et al. Tree tensor network state with variable tensor order:An efficient multireference method for strongly correlated systems[J]. Journal of chemical theory and computation,2015,11(3):1027-1036.

[12] OSELEDETS I V. Tensor-train decompositi-on[J]. SIAM Journal on Scientific Computing,2011,33(5):2295-2317.

[13] HÜBENER R,NEBENDAHL V,DÚ R W. Concatenated tensor network states[J]. New Journal of Physics,2010,12(2):025004.

[14] BENGUA J A,HO P N,TUAN H D,et al. Matrix product state for higher-order tensor compression and classification[J]. IEEE Transactions on Signal Processing,2017,65(15):4019-4030.

[15] ZHAO Q,ZHOU G,XIE S,et al. Tensor ring decomposition[J]. arXiv:1606.05535,2016.

[16] MICKELIN O,KARAMAN S. Tensor ring decomposition[J]. arXiv:1807.02513,2018.

[17] WANG W,AGGARWAL V,AERON S. Efficient low rank tensor ring completion[C]// Proceedings of the IEEE International Conference on Computer Vision. IEEE,2017:5697-5705.

[18] FERNANDO B,HABRARD A,SEBBAN M,et al. Unsupervised visual domain adaptation using subspace alignment[C]// Proceedings of the IEEE International Conference on Computer Vision. IEEE,2013:2960-2967.

[19] GONG B,SHI Y,SHA F,et al. Geodesic flow kernel for unsupervised domain adaptationn[C]// 2012 IEEE Conference on Computer Vision and Pattern Recognition. IEEE,2012:2066-2073.

[20] LONG M,WANG J,DING G,et al. Transfer learning with graph co-regularization[J]. IEEE Transactions on Knowledge and Data Engineering,2014,26(7):1805-1818.

[21] SAENKO K,KULIS B,FRITZ M,et al. Adapting visual category models to new domains[C]// European Conference on Computer Vision. Berlin:Springer,2010:213-226.

[22] GRIFFIN G,HOLUB A,PERONA P. Caltech-256 object category dataset:Technical Report 7694 [R]. California Institute of Technology,2007.

[23] SIMONYAN K,ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv:1409.1556,2014.

[24] HE K,ZHANG X,REN S,et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2015,37(9):1904-1916.