

基于 Tucker 分解的半监督支持张量机

吴振宇 李云雷 吴凡

(大连理工大学创新创业学院 辽宁 大连 116024)

摘要 传统的机器学习方法所使用的数据大多是基于向量空间的。支持向量机(Support Vector Machine, SVM)作为一种重要的机器学习方法,在解决小样本、非线性、高维数据等问题时具有较好的性能。但在实际应用中,图像和视频等数据都是用张量形式表示的,如果将这些张量数据直接转换成向量数据,往往会丢失一些原有的结构和相关性信息,有可能造成维度灾难和小样本问题。为了设法保持尽量多的张量结构信息,提出了一种采用 Tucker 分解的支持张量机(Support Tensor Machine, STM)算法。实验表明,该方法可以明显提高分类器性能;同时,支持张量机作为监督学习方式,存在无法利用未标记数据的缺点,往往受限于训练数据不足的情况。因此,将半监督学习方法与支持张量机相结合,进而提出了基于 Tucker 分解的半监督支持张量机算法(Semi-Supervised STM, S3TM)。该算法既可以保持较多的张量结构信息,又能充分利用未标记数据。实验表明,采用该算法的预测准确率达到 90.26%,从而验证了所提算法的有效性。

关键词 Tucker 分解,支持张量机,半监督学习

中图分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.09.028

Semi-supervised Support Tensor Based on Tucker Decomposition

WU Zhen-yu LI Yun-lei WU Fan

(School of Innovation and Entrepreneurship, Dalian University of Technology, Dalian, Liaoning 116024, China)

Abstract Most data used by traditional machine learning methods belong to vector space. As an important machine learning method, support vector machine (SVM) has better performance in solving small samples, nonlinearity and high-dimensionality problems. However, in practical applications, some data like images and videos are both stored in tensor form. If tensor data are convert into vector data, some original structure and related information may be lost, which will cause dimensional disasters and small samples problems. Therefore, to maintain as much tensor structure information as possible, support tensor machine (STM) based on Tucker decomposition was proposed. Experiments show that this method can significantly improve the classification performance. Meanwhile, as a supervised learning method, support vector machine cannot use unlabeled data, which often encounter problems with insufficient training data. Therefore, semi-supervised support tensor machine based on Tucker decomposition was proposed. This algorithm can not only maintain more tensor structure information, but also make full use of the unlabeled data. Experiments show that the prediction accuracy rate is 90.26%, which validates the effectiveness of the proposed method.

Keywords Tucker decomposition, Support tensor machine, Semi-supervised learning

1 引言

在数学的定义上,张量本质上是多维数组或多维阵列;张量可以看成是向量和矩阵由低维空间到高维空间的推广。例如,向量是一阶张量,矩阵是二阶张量^[1-2]。在现实生活中,很多数据也都是使用张量的形式表示,例如,人脸灰度图像和手写体图像可以看成是二阶张量;彩色图像可以表示为三阶张量,而彩色视频则是四阶张量^[3-4]。

传统的分类算法通常使用向量作为输入数据的描述方式。在处理上述问题时,一般将张量形式的的数据展开为向量

形式,但这种做法在扩大数据维数的同时,破坏了数据本身所包含的空间结构信息。为了解决这些问题,研究者们提出了各种基于张量数据分类的算法。Cai 等通过对矩阵空间的分析,提出了包含秩一支持矩阵机在内的多种基于矩阵数据的机器学习方法^[5-6];之后, Tao 等提出了监督张量学习的基本框架^[7],并在此基础上实现了秩一支持张量机。秩一支持张量机可以看作是支持矩阵机由二阶张量向高阶张量的推广。通过对秩一支持张量机进行理论分析与论证, Kotsia 等提出了高秩支持张量机^[8-10]。STM 在处理小样本问题时表现出的优势使其逐渐应用于各个方面^[11-13]。

到稿日期:2018-07-09 返修日期:2018-11-17

吴振宇(1971—),男,博士,副教授,主要研究方向为人工智能、机器学习, E-mail: zhenyuwu@dlut.edu.cn(通信作者);李云雷(1992—),男,硕士,主要研究方向为机器学习;吴凡(1996—),男,硕士,主要研究方向为机器学习。

SVM 和 STM 都是一种监督学习算法,必须依赖有标记的训练样本。而实际上,有标记样本的数量一般比较少,大部分数据都是未标记的。监督学习由于无法利用未标记数据,往往会因为训练数据较少而产生过拟合,对分类器的泛化能力会产生很大的影响^[14-16];而人为地对大量数据进行标记又会耗费大量的人力与物力。半监督学习算法则可以在少量标记样本信息下利用大量未标记样本,克服了监督学习的缺点,使机器学习具有更好的推广能力^[17-18]。

通过对以上内容的研究与分析,本文主要从以下两方面开展相应的研究工作。1)为了保持更多的张量结构信息,获得更好的分类性能,本文采用 Tucker 分解对张量数据进行处理;实验表明,与 SVM 对比,本文采用 Tucker 分解的 STM 在处理张量数据时具有明显的优势,显著提升了预测准确率。2)为了提升 STM 分类器的泛化能力,解决监督学习训练样本不足的问题,本文将半监督学习与 STM 结合,提出了基于 Tucker 分解的半监督支持张量机(S3TM)。实验表明,在具有大量未标记数据的情况下,采用本文所提算法时分类性能大幅提升,预测准确率达到 90.26%。

2 张量的基本理论

本文沿用了线性代数、模式识别和机器学习领域通用的符号表示方法,具体符号描述如表 1 所列。

表 1 符号列表
Table 1 List of symbols

符号	描述
\mathcal{X}_i	第 i 个张量输入样本
y_i	张量样本 \mathcal{X}_i 的类标
\mathcal{W}	张量权值
\mathbf{w}	向量权值
C	惩罚系数
ξ	松弛变量
m	样本数目
α, β	拉格朗日乘子向量
N	张量的阶数

定义 1(张量) 一个 N 阶张量就是一个 N 维数组,将其记为 $\mathcal{A} \in R^{(I_1 \times I_2 \times \dots \times I_N)}$,其阶数为 N 。张量的元素记为 $a_{i_1, i_2, \dots, i_N}, 1 \leq i_n \leq I_n, 1 \leq n \leq N$ 。实际上,向量为二阶张量,矩阵为二阶张量,三阶及以上张量称为高阶张量。

定义 2(张量的外积) 张量 $\mathcal{X} \in R^{(I_1, I_2, \dots, I_N)}$ 和 $\mathcal{Y} \in R^{(J_1, J_2, \dots, J_M)}$ 的外积定义为:

$$(\mathcal{X} \otimes \mathcal{Y})_{i_1, i_2, \dots, i_N, j_1, j_2, \dots, j_M} = x_{i_1, i_2, \dots, i_N} y_{j_1, j_2, \dots, j_M} \quad (1)$$

定义 3(张量的内积) 两个阶数为 N 的张量 $\mathbf{X}, \mathbf{Y} \in R^{(I_1, I_2, \dots, I_N)}$ 之间的内积被定义为:

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1, i_2, \dots, i_N} y_{i_1, i_2, \dots, i_N} \quad (2)$$

定义 4(张量 n -模积) 张量 $\mathcal{A} \in R^{(I_1, I_2, \dots, I_N)}$ 和矩阵 $U \in R^{(J_N \times I_N)}$ 的 n -模积被记为 $\mathcal{A} \times_n U$,它是一个张量,属于 $R^{(I_1, I_2, \dots, I_{n-1} \times J_1, I_{n+1}, \dots, I_N)}$ 空间,具体定义如下:

$$(\mathcal{A} \times_n U)_{(i_1, i_2, \dots, i_{n-1}, j_1, i_{n+1}, \dots, i_N)} = \sum_{i_n=1}^{I_n} a_{i_1, i_2, \dots, i_N} u_{j_n, i_n} \quad (3)$$

给定一个张量 $\mathcal{A} \in R^{(I_1 \times I_2 \times \dots \times I_N)}$ 和一系列矩阵 $U^{(n)} \in R^{(J_n \times I_n)}, J_n < I_n, n = 1, 2, \dots, N$, 则 \mathcal{A} 到张量子空间

$R^{(I_1 \times I_2 \times \dots \times I_N)}$ 的投影定义为 $\mathcal{A} \times_1 U^{(1)} \times_2 U^{(2)} \times \dots \times_N U^{(N)}$ 。

定义 5(张量的 Frobenius 范数) 张量 $\mathcal{A} \in R^{(I_1 \times I_2 \times \dots \times I_N)}$ 的 Frobenius 范数被定义为所有元素平方和的二次方根:

$$\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} a_{i_1, i_2, \dots, i_N}^2} \quad (4)$$

定义 6(张量的秩一分解) 张量 $\mathcal{A} \in R^{(I_1 \times I_2 \times \dots \times I_N)}$ 可以近似改写为:

$$\mathcal{A} \approx \sum_{r=1}^R \mathbf{u}_r^{(1)} \otimes \mathbf{u}_r^{(2)} \otimes \dots \otimes \mathbf{u}_r^{(N)} = \sum_{r=1}^R \prod_{n=1}^N \mathbf{u}_r^{(n)} \quad (5)$$

定义 7(张量的 Tucker 分解) 张量 $\mathcal{A} \in R^{(I_1 \times I_2 \times \dots \times I_N)}$ 可以近似改写为:

$$\mathcal{A} \approx \{\mathcal{G}; A_1, A_2, \dots, A_N\} = \mathcal{G} \times_1 A_1 \times_2 A_2 \times \dots \times_N A_N \quad (6)$$

其中, $\mathcal{G} \in R^{(I_1 \times I_2 \times \dots \times I_N)}$ 是核张量, $A_{(k)} \in R^{(I_k \times g_k)}$ 为因子矩阵, $k=1, 2, \dots, N$ 。

3 传统支持向量机与支持张量机

3.1 支持向量机

支持向量机是建立在统计学习理论中的一种机器学习方法,它通过有限的训练样本信息在模型的复杂性和学习能力之间寻求最佳折中,以求获得最好的推广能力。其核心内容是:如果在输入空间中样本是线性可分的,那么可以直接在原始空间(即原始的训练样本)中构造最优分类超平面;而构造最优分类超平面的过程在数学上可以归结为求解一个具有不等式约束的二次规划问题;如果输入空间中样本是线性不可分的,则需要通过将输入样本映射到高维特征空间,使其在高维空间中线性可分,再构造分类超平面。

对于二分类问题,给定训练样本 $T = \{(x_1, y_1), \dots, (x_m, y_m)\}, y_i \in \{-1, +1\}$,其中 $x_i \in R^n$ 是输入向量, $y_i \in \{-1, +1\}$ 是 x_i 的分类标志。分类超平面表示为 $\mathbf{w}^T \mathbf{x} + b = 0, \mathbf{w} = (w_1, w_2, \dots, w_d)$ 为法向量。想要获得最好的分类效果,就要使得分类超平面距离两类样本的距离最大,也即令 $2/\|\mathbf{w}\|$ 最大。最大化间隔,等价于求 $\|\mathbf{w}\|$ 最小。另外,为了提高超平面的推广泛化能力,引入松弛变量 $\xi > 0$ 。SVM 构造最优超平面的问题转变成求解如下凸二次规划问题:

$$\min \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i \quad (7)$$

$$\text{s. t. } y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, 2, \dots, m$$

其中, C 为惩罚因子,表示对错误的惩罚程度。对于该问题,可以采用拉格朗日乘子法得到其对偶问题:

$$L(\mathbf{w}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i (y_i (\mathbf{w}^T \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^m \beta_i \xi_i \quad (8)$$

令 $L(\mathbf{w}, b, \xi, \alpha, \beta)$ 对 \mathbf{w}, b 和 ξ 求偏导并取零:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^m (\alpha_i y_i) = 0 \quad (9)$$

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow \alpha_i + \beta_i = C, 1 \leq i \leq m$$

将式(9)代入到原问题式(8)中,可得对偶问题为:

$$\min \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j - \sum_{i=1}^m \alpha_i \quad (10)$$

$$\text{s. t. } \sum_{i=1}^m \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, m$$

求解以上优化问题,可以得到超平面参数 α 和 b ,进而可以得到分类函数式(11):

$$f(x) = \mathbf{w}^T \mathbf{x} + b = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b \quad (11)$$

3.2 支持张量机

在张量的学习框架下,我们对支持向量机进行了推广。设计支持张量机的目标规划为:

$$\min \frac{1}{2} \|\mathcal{W}\|^2 + C \sum_{i=1}^m \xi_i \quad (12)$$

$$\text{s. t. } y_i (\langle \mathcal{W}, \mathcal{Q}_i \rangle + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \dots, m$$

与支持向量机相比,输入参数从向量 \mathbf{x} 变为张量 \mathcal{X} ,法向量 \mathbf{w} 变为张量形式 \mathcal{W} ,向量的乘积变为张量的内积;当输入一阶张量时,该模型退化为支持向量机。

求解该问题时,首先要求出其对偶问题:

$$L(\mathcal{W}, b, \xi, \alpha, \beta) = \frac{1}{2} \|\mathcal{W}\|^2 + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i (y_i (\langle \mathcal{W}, \mathcal{Q}_i \rangle + b) - 1 + \xi_i) - \sum_{i=1}^m \beta_i \xi_i \quad (13)$$

求函数 $L(\mathcal{W}, b, \xi, \alpha, \beta)$ 关于 \mathcal{W} , b 和 ξ 的偏导并令其为零:

$$\frac{\partial L}{\partial \mathcal{W}} = 0 \Rightarrow \mathcal{W} = \sum_{i=1}^m \alpha_i y_i \mathcal{Q}_i$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^m (\alpha_i y_i) = 0 \quad (14)$$

$$\frac{\partial L}{\partial \xi} = 0 \Rightarrow \alpha_i + \beta_i = C, 1 \leq i \leq m$$

将式(14)代入式(13)中,可得其对偶问题:

$$\min \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle \mathcal{Q}_i, \mathcal{Q}_j \rangle - \sum_{i=1}^m \alpha_i \quad (15)$$

$$\text{s. t. } \sum_{i=1}^m \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, m$$

对于该模型,在支持向量机最大间隔的思想下,将权重参数以及输入数据转换成张量数据后就可以直接处理张量数据,其中张量内积 $\langle \mathcal{Q}_i, \mathcal{Q}_j \rangle$ 根据定义即可求出,然后采用 SMO 算法求出最终分类函数:

$$f(x) = \langle \mathcal{W}, \mathcal{X} \rangle + b = \langle \sum_{i=1}^m \alpha_i y_i \mathcal{Q}_i, \mathcal{X} \rangle + b \quad (16)$$

但这种朴素支持张量机的推广意义不大,它本质上只是经典支持向量机的一个线性映射,因为其结果与直接将数据展开成向量并采用支持向量机的结果是一样的。这种推广方式无法体现出张量数据原生的存在于张量内部的结构信息,而 STM 的核心是要尽可能多地保留张量结构信息。

基于这样的考虑,Cai 等提出了支持矩阵机,将二阶张量定义为 $\mathcal{W} \approx \mathbf{u} \times \mathbf{v}^T$ 。基于此方法,支持矩阵机使得排列方式不同的矩阵数据得到不同的结果。对于矩阵而言,其结构信息在结果中得以体现。

之后,Tao 等将支持张量机的张量权重限制在 CP 秩意义下的秩一张量中,得出了广为研究者所知的经典支持张量机模型。张量秩一分解如式(17)所示:

$$\mathcal{W} \approx \mathbf{w}_1 \otimes \mathbf{w}_2 \otimes \dots \otimes \mathbf{w}_m = \sum_{k=1}^m \mathbf{w}_k \quad (17)$$

对于该问题,Tao 等通过交替投影法求取 \mathcal{W} ,最后得出分

类决策函数:

$$f(x) = \langle \mathcal{W}, \mathcal{X} \rangle + b = \langle \sum_{k=1}^m \mathbf{w}_k, \mathcal{X} \rangle + b \quad (18)$$

$$\langle \sum_{k=1}^m \mathbf{w}_k, \mathcal{X} \rangle = \mathcal{X} \times_1 \mathbf{w}_1 \times_2 \mathbf{w}_2 \times \dots \times_m \mathbf{w}_m$$

但该问题把张量权重限制在 CP 秩意义下的秩为一的张量中。只用秩一分解对于张量数据有很大的局限性,为此 Hao 等在支持张量机的基础上将张量进行 CP 分解,形成多个秩一张量的和,将 CP 秩由一扩展为 R ,如式(19)所示:

$$\mathcal{W} \approx \sum_{i=1}^R \mathbf{w}_{i1} \otimes \mathbf{w}_{i2} \otimes \dots \otimes \mathbf{w}_{im} = \sum_{i=1}^R \sum_{k=1}^m \mathbf{w}_{ik} \quad (19)$$

对于张量内积 $\langle \mathcal{Q}_i, \mathcal{Q}_j \rangle$ 的求解问题,Hao 等将 \mathcal{Q}_i 和 \mathcal{Q}_j 进行 CP 分解再求取内积,如式(20)所示:

$$\langle \mathcal{Q}_i, \mathcal{Q}_j \rangle = \langle \sum_{p=1}^R \sum_{l=1}^m \mathbf{x}_{ip}, \sum_{q=1}^R \sum_{j=1}^m \mathbf{x}_{jq} \rangle = \sum_{p=1}^R \sum_{l=1}^m \prod_{n=1}^N \langle \mathbf{x}_{ip}, \mathbf{x}_{jq} \rangle \quad (20)$$

通过 SMO 算法计算出 \mathcal{W} ,最后求出决策函数:

$$f(x) = \langle \mathcal{W}, \mathcal{X} \rangle + b = \langle \sum_{p=1}^R \sum_{i=1}^m \mathbf{w}_{ip}, \sum_{q=1}^R \sum_{j=1}^m \mathbf{x}_{jq} \rangle + b \quad (21)$$

4 基于 Tucker 分解的半监督支持张量机算法

4.1 基于 Tucker 分解的 STM

通过对以上方法的研究与分析,本文在前人的基础上选用更具一般性的 Tucker 分解进行张量数据的处理。Tucker 分解的概念最早于 1963 年被提出,它将一个张量分解成一个核张量与一些矩阵的外积^[19-20]。经过后续研究,人们发现 Tucker 分解实际上就是高阶主成分分析。

对于一个三阶张量,Tucker 分解是指将一个张量表示成一个核张量沿着每一个模式乘以一个矩阵的形式。如图 1 所示,一个三阶张量可以分解成一个三阶核张量 \mathcal{G} 和 3 个因子矩阵乘积的方式,其中每个模式上的因子矩阵为张量在该模式上的主成分。对于 Tucker 分解,如果其核张量 \mathcal{G} 的各个维数相同且是对角的,则 Tucker 分解就退化成 CP 分解。本质上,CP 分解是 Tucker 分解的特殊形式^[21-22]。

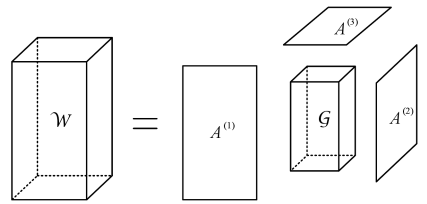


图1 三阶张量 \mathcal{G} 的 Tucker 分解

Fig. 1 Tucker decomposition of third-order tensor \mathcal{G}

CP 分解局限于只能把张量分解成向量乘积的方式;相比 CP 分解,Tucker 分解更具一般性。CP 分解可以看成是 Tucker 分解中核张量的一种特殊形式。

Tucker 分解如下所示:

$$\mathcal{W} \approx \{\mathcal{G}; A_1, A_2, \dots, A_N\} = \mathcal{G} \times_1 A_1 \times_2 A_2 \times \dots \times_N A_N \quad (22)$$

张量内积 $\langle \mathcal{Q}_i, \mathcal{Q}_j \rangle$:

$$\langle \mathcal{Q}_i, \mathcal{Q}_j \rangle \approx \langle \mathcal{G}_i \otimes_{k=1}^N A_{ik}, \mathcal{G}_j \otimes_{k=1}^N A_{jk} \rangle \quad (23)$$

利用式(23)求取张量内积后,STM 的对偶问题式(15)就可以采用 SMO 算法进行求解。根据式(14)求出 \mathcal{W} 和 b ,得到决策函数:

$$f(x) = \langle \mathcal{W}, \mathcal{X} \rangle + b = \langle \mathcal{G}_w \times_{k=1}^N A_{wk}, \mathcal{G}_x \otimes_{k=1}^N A_k \rangle + b \quad (24)$$

在张量分解时,核张量 \mathcal{G} 的选取至关重要。 \mathcal{G} 若过大,则失去了张量分解的意义,几乎保留了所有张量数据,没有凸显出张量的核心结构信息; \mathcal{G} 若过小,则会丢失过多的张量结构信息,使得张量信息丢失严重,分类性能下降。

核张量 \mathcal{G} 各维度的大小可以看作是张量信息对整个分类结果的贡献程度。核张量 \mathcal{G} 某一维度的数值越大,这个维度保留的结构信息就越多,但也保留了更多对分类结果无用的信息。因此,核张量的选择至关重要,过大或过小都会影响分类性能。本文在选取核张量 \mathcal{G} 时,采用逐一选取的方法,即每次固定其他维度,只改变一个维度的大小,然后选择使分类性能最佳的维度数值;之后使用该方法依次选出其他所有维度,最终得到最优核张量 \mathcal{G} 。

对于张量数据,采用 Tucker 分解后,对张量的内部信息加以改变,在保留张量的核心结构信息的同时削弱了张量内部与张量核心内容无关的信息,使得张量的内部结构信息能够在分类结果中凸显出来。

传统的 SVM 方法直接将张量数据转换成向量数据,使得张量数据的空间结构信息被破坏;而采用 Tucker 分解后,更多的张量结构信息被保存下来,使张量内部的不同结构能够在结果中得以体现。在求取张量内积时,对张量 $\mathcal{Q}_i, \mathcal{Q}_j$ 进行 Tucker 分解,分解后的张量的内积会相应改变,进而反映到张量的权值 \mathcal{W} 中;而且采用 Tucker 分解的方式,原问题可以利用通用 SMO 算法求解,降低了算法的时间开销。基于 Tucker 分解的支持张量机的具体算法如算法 1 所示。

算法 1 基于 Tucker 分解的支持张量机

输入: 样本 $T = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}, y_i \in \{-1, +1\}$, 权衡参数 C , 核张量 \mathcal{G}

输出: 预测测试数据的分类结果 y_i

Step 1 选取核张量 \mathcal{G} , 对所有数据进行 Tucker 分解, 计算张量内积。

Step 2 利用式(23)和 SMO 算法求解如下目标函数:

$$\begin{aligned} \min & \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j \langle \mathcal{Q}_i, \mathcal{Q}_j \rangle - \sum_{i=1}^m \alpha_i \\ \text{s. t.} & \sum_{i=1}^m \alpha_i y_i = 0, 0 \leq \alpha_i \leq C, i = 1, 2, \dots, m \end{aligned}$$

Step 3 利用式(14)求出 \mathcal{W} , 得到分类函数。

Step 4 用分类函数预测测试数据, 输出分类结果 y_i 。

4.2 基于 Tucker 分解的 S3TM

在 STM 的基础上, 为了克服监督学习无法利用未标记数据的缺点, 本文将半监督学习算法与 STM 结合, 采用自训练的方式提取未标记样本。

该算法的思想是首先根据已有的训练样本训练分类器; 然后用其对未标记样本进行类别标记, 再从未标记样本中选取若干个置信度较高的样本加入到训练样本集中, 得到一个新的训练样本; 接着, 用新的训练样本继续训练分类器, 不断重复上述训练过程, 直至达到迭代停止条件; 最后输出分类函数。

在确定高置信度的样本时, 首先利用已标记样本训练的分类器对未标记样本进行分类, 然后根据分类结果对所有样本由大到小排序, 排序靠前的数据距离分类超平面更远。根据排序结果设置置信度大小, 排序靠前的数据的置信度高, 排序靠后的置信度低。

该方法的优点是简单易行, 但也存在相应的问题: 如果分

类器标记的样本存在错误, 那么该错误会随着迭代次数的增加而不断加强, 最终导致过训练, 影响分类器效果。为了避免产生过训练, 一方面, 在提取未分类样本时要选取分类置信度较高的样本, 尽量避免超平面附近可能标错的样本; 另一方面, 在保证分类性能的前提下尽量减少迭代次数, 避免过训练, 减少迭代次数还可以降低训练时间开销, 具体算法如算法 2 所示。

算法 2 基于 Tucker 分解的半监督支持张量机

输入: 标记样本 T_1 , 未标记样本 T_2 , 权衡参数 C , 核张量 \mathcal{G} , 置信度和迭代次数 N

$$T_1 = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}, y_i \in \{-1, +1\}$$

$$T_2 = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$$

输出: 预测测试数据的分类结果 y_i

Step1 设置核张量 \mathcal{G} , 对所有数据进行 Tucker 分解, 计算张量内积。

for $i = 1 : N$ do

Step2 选取训练数据训练 STM 分类器, 采用 SMO 算法和式(24)求解优化问题式(15)。

Step3 利用式(14)求出 \mathcal{W} , 得到 STM 分类函数。

Step4 使用分类函数对未标记数据进行分类并标记。

Step5 提取置信度较高的部分加入到训练数据集, 得到新的训练数据集。

end for

Step6 得到最终分类函数式(24)并预测测试样本, 输出分类结果 y_i 。

5 实验及结果分析

5.1 实验环境与数据样本

本实验所有程序均在 Matlab2016a 软件平台上进行编译, 算法的运行环境为 Intel(R) Core(TM) i5-4570 3.2 GHz CPU, 16GB 内存, Microsoft Windows 10 操作系统。

测试数据来自 Corel 图像库, 本文选取老虎和猎豹两类图像数据进行分类测试。实验总计选取 530 张图像, 其中老虎图像和猎豹图像各占 265 张。本次实验随机挑选 206 张图像作为训练数据, 随机挑选 114 张图像作为测试数据, 其余 210 张图像作为未标记数据。

部分实验数据图像如图 2 和图 3 所示, 所有图像数据皆是在不同背景、光照、姿态下拍摄。原始图像大小各不相同, 同时考虑到训练时间问题, 本文在实验中将所有图像的分辨率压缩为 160×120 。



图 2 部分老虎数据集图片

Fig. 2 Partial tiger pictures



图 3 部分猎豹数据集图片

Fig. 3 Partia leopard pictures

5.2 实验结果分析

分别选取支持向量机(SVM)、基于 Tucker 分解的支持张量机(STM)、半监督支持向量机(S3VM)和基于 Tucker 分解的半监督支持张量机(S3TM)4 种分类算法进行对比,为了降低随机性的影响,每次实验进行 10 次并取平均值。本文实验中参数 C 的取值为 3,迭代次数 N 为 2,核张量 \mathcal{Q} 选取为 $10 \times 15 \times 30$,置信度大小在 $0.4 \sim 0.5$ 之间。

图 4 和图 5 给出了半监督学习迭代次数与置信度对分类器的影响。从图 4 中可以看出,起初,随着半监督学习算法迭代次数的增加,分类器获取的未标记样本增加,分类器性能逐渐提升;但随后进一步增加迭代次数时分类器性能反而下降,这是因为随着迭代次数的增加,错误标记被不断加强,最后错误标记所带来的负面影响逐渐超过数据量增加所带来的正面影响,分类器性能下降。

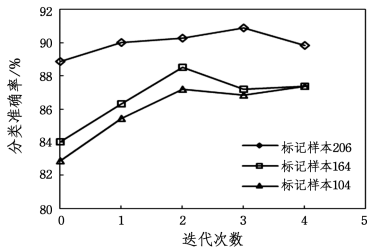


图 4 迭代次数对实验结果的影响

Fig. 4 Effect of number of iterations on experimental results

如图 5 所示,提取未标记样本时,置信度越高,提取的样本越准确,但是提取的样本数量越少。在置信度较高时,随着置信度的降低,分类器性能逐渐提升,此阶段的错误标记较少,提取的未标记样本数量大幅提升,样本数量起主导作用。但是随着置信度的继续降低,错误标记样本开始不断增加,并逐渐起到主导作用,随着样本数量的大量增加,分类性能不断下降。

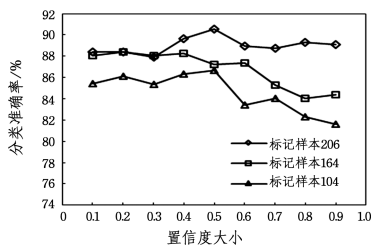


图 5 置信度对实验结果的影响

Fig. 5 Effect of confidence on experimental results

通过对半监督学习迭代次数与置信度的分析,综合考虑分类性能与时间开销,最终选取迭代次数 N 为 2,置信度大小在 $0.4 \sim 0.5$ 之间。

表 2 给出了在不同训练样本数量下各个算法的分类性能和学习时间。从表中可以看到,采用 Tucker 分解的 STM 的预测准确率明显优于 SVM,整体识别率提高了 2% 左右,验证了 Tucker 分解的效果;同时,S3TM 的准确率相比 STM 有 3% 左右的提升,预测精度达到 90.26%,验证了本文所提算法的有效性。

表 2 实验结果对比

Table 2 Comparison of experimental results

训练样本数	算法	准确率/%	学习时间/s
80	SVM	77.81	3.31
	STM	79.47	15.21
	S3VM	83.33	44.65
	S3TM	83.59	68.17
104	SVM	80.18	5.59
	STM	82.89	17.75
	S3VM	84.30	61.77
	S3TM	85.96	83.713
164	SVM	82.72	14.24
	STM	84.12	28.54
	S3VM	87.63	115.15
	S3TM	88.51	144.22
206	SVM	86.84	22.07
	STM	88.86	40.03
	S3VM	90.26	145.29
	S3TM	90.26	167.88

从表 2 还可以看出,采用 Tucker 分解的 STM 学习时间长于 SVM,相差 10~20s,这是由于对图像进行 Tucker 分解需要一定的时间。实验中,S3TM 的预测准确率相比 SVM 有很大的提高,但学习时间也有所延长。本实验 S3TM 算法进行了 2 次迭代,最长学习时间达到了 167.88s,一方面是因为该算法需要进行多次训练,另一方面是半监督学习引入了大量未标记数据,使得训练数据大大增加。

图 6 给出了 STM,SVM,S3TM,S3VM 算法的分类性能对比结果。从图中可以看出,采用 Tucker 分解的 STM 性能明显优于 SVM,S3TM 的分类性能优于 S3VM,且随着训练样本的逐渐增加,S3TM 与 S3VM 的差距逐渐减小,从而验证了在小样本下,使用 Tucker 分解的 STM 相比 SVM 更具优势,但随着样本数量的增加,性能差距逐渐缩小。相比于监督学习的 STM 与 SVM,采用半监督学习算法的 S3TM 与 S3VM 具有明显的分类优势,分类性能大幅提升,从而验证了半监督算法的有效性。与其他算法相比,本文所提 S3TM 具有绝对优势。

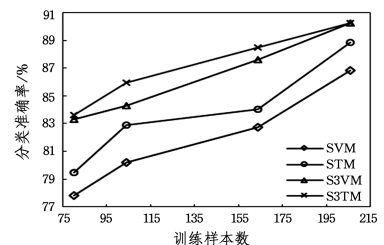


图 6 4 种算法的分类性能对比

Fig. 6 Classification performance comparison of four algorithms

结束语 本文分析了传统 STM 的发展过程,介绍了 SVM 与 STM 基本算法,并在此基础上提出了新的基于 Tucker 分解的 STM 算法。实验表明,使用 Tucker 分解的 STM 分类性能显著优于 SVM。同时,本文将半监督学习算法与 STM 相结合,提出了 S3TM 算法。实验分析结果表明:采用了 Tucker 分解的 S3TM 的分类效果显著,分类正确率达到 90.26%,具有绝对优势。但本文提出的算法仍然存在一些问题:该算法目前只能解决线性问题,不适合非线性分类问题;同时,该算法的训练时间相对较长,还须进一步改进算法,以降低时间开销,拓展非线性分类问题。

参考文献

- [1] GUO X, HUANG X, ZHANG L, et al. Support Tensor Machines for Classification of Hyperspectral Remote Sensing Imagery[J]. IEEE Transactions on Geoscience & Remote Sensing, 2016, 54(6): 3248-3264.
- [2] ZHANG J, HAN Y, JIANG J. Tucker decomposition-based tensor learning for human action recognition[J]. Multimedia Systems, 2016, 22(3): 343-353.
- [3] RAJWADE A, RANGARAJAN A, BANERJEE A. Image Denoising Using the Higher Order Singular Value Decomposition [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(4): 849-862.
- [4] BISWAS S K, MILANFAR P. Linear Support Tensor Machine with LSK Channels: Pedestrian Detection in Thermal Infrared Images[J]. IEEE Transactions on Image Processing, 2017, 26(9): 4229-4242.
- [5] CAI D, HE X F, WEN J R, et al. Support Tensor Machines for Text Categorization[J]. International Journal of Academic Research in Business & Social Sciences, 2006, 2(12): 2222-6990.
- [6] LUO L, XIE Y, ZHANG Z, et al. Support matrix machines [C] // International Conference on International Conference on Machine Learning. JMLR. org, 2015: 938-947.
- [7] TAO D, LI X, HU W, et al. Supervised Tensor Learning[C] // IEEE International Conference on Data Mining. IEEE, 2005.
- [8] KOTSIA I, PATRAS I. Support tucker machines [C] // The 24th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011). Colorado Springs, CO, USA, IEEE, 2011.
- [9] HAO Z, HE L, CHEN B, et al. A Linear Support Higher-Order Tensor Machine for Classification[J]. IEEE Transactions on Image Processing, 2013, 22(7): 2911-2920.
- [10] CHEN Y, LU L, ZHONG P. One-class support higher order tensor machine classifier[J]. Applied Intelligence, 2017, 47(7): 1-9.
- [11] CHEN Z, BATSELIER K, SUYKENS J A K, et al. Parallelized Tensor Train Learning of Polynomial Classifiers [J]. IEEE Transactions on Neural Networks & Learning Systems, 2016, PP(99): 1-12.
- [12] SHU T. Research on Regression Problem and Classification Problem Algorithm with Input Tensor Data[D]. Urumqi: Xinjiang University, 2017. (in Chinese)
舒托. 输入为张量的回归问题和分类问题算法研究[D]. 乌鲁木齐: 新疆大学, 2017.
- [13] YU J, LIU G S. Text Classification Research Based on Support Tensor[J]. Information Technology, 2016(9): 10-13. (in Chinese)
俞炯, 刘功申. 基于支持张量机的文本分类研究[J]. 信息技术, 2016(9): 10-13.
- [14] LIU J W, LIU Y, LUO X L. Semi-supervised learning algorithm [J]. Chinese Journal of Computers, 2015, 38(8): 1592-1617. (in Chinese)
刘建伟, 刘媛, 罗雄麟. 半监督学习方法[J]. 计算机学报, 2015, 38(8): 1592-1617.
- [15] CHEN Y H. Research on Semi-supervised learning classification algorithm[D]. Zhenjiang: Jiangsu University, 2017. (in Chinese)
陈英豪. 半监督学习分类算法的研究[D]. 镇江: 江苏大学, 2017.
- [16] WIDMANN N, VERBERNE S. Graph-based Semi-supervised Learning for Text Classification[C] // Acm Sigir International Conference on Theory of Information Retrieval. ACM, 2017.
- [17] HADY M F A, SCHWENKER F. Semi-supervised Learning [J]. Intelligent Systems Reference Library, 2013, 49(2): 215-239.
- [18] XU C, TAO W. Combining Active Learning and Semi-Supervised Learning by Using Selective Label Spreading [C] // IEEE International Conference on Data Mining Workshops. IEEE, 2017.
- [19] TUCKER L R. Extension of factor analysis to three-dimensional matrices[M] // Contributions to Mathematical Psychology. New York: Helt Rinehart & Winston, 1964.
- [20] TUCKER L R. Some mathematical notes on three-mode factor analysis[J]. Psychometrika, 1966, 31(3): 279-311.
- [21] YANG B. Research and Application of Machine Learning Method Based on Tensor Data[D]. Beijing: China Agricultural University, 2014. (in Chinese)
杨兵. 基于张量数据的机器学习方法研究与应用[D]. 北京: 中国农业大学, 2014.
- [22] ERFANI S M, BAKTASHMOTLAGH M, RAJASEGARAR S, et al. R1STM: One-class Support Tensor Machine with Randomised Kernel[C] // SIAM International Conference on Data Mining. Philadelphia: Society for Industrial and Applied Mathematics Conference, 2016.