
FSPMTL: 一个新的自步多任务学习框架

孙立健 周鋈

中国长沙国防科技大学信息系统工程重点实验室, 410072

摘要: 多任务学习 (Multi-Task Learning, MTL) 是在有限的条件下, 同时利用任务间的共性和差异来提高学习效果的一种方法。然而, 在实际问题中, 存在着大量的样本噪声, 这些噪声可能会严重影响 MTL 的性能。为了解决这一问题, 引入了自学习 (SPL) 方法, 通过逐步增加从最简单样本到最困难样本的实例数目来提高其性能。在目前的多任务自学习方法中, 大多数都是作为 SPL 项引入到优化过程中, 这对 SPL 和 MTL 的结合造成了很大的限制。本文提出了一种新的灵活学习框架, 将 MTL 和 SPL 相结合, 在学习过程中分为两个阶段, 使其更适合于学习困难的样本和任务。通过这个框架, 我们可以同时利用现有的 MTL 模型和 SPL 模型。通过对合成数据集和真实数据集的进一步实验表明, 与其他最新算法相比, 我们的方法具有更高的效率。

关键字: 多任务学习, 自主学习, 一种新的灵活框架, 多任务自主学习

本研究得到了国家自然科学基金 61703416 资助、湖南省自然科学基金 2018JJ3614 资助和湖南省研究生科研创新项目 CX20190041 资助

一、引言

在类人推理过程的启发下, MTL 不仅可以同时学习多个相关任务, 而且可以利用相关任务之间的共享表示来微调原始任务的广义模型[1,2].

在过去的十年中, 许多 MTL 方法被提出并应用于各种情况, 大致可分为三大类[3-6]。第一类假设所有任务共享一个公共的低秩特征表示[7-10]。第二类假设不同的任务在训练的模型中可能有共享的参数[5,8,11,12]。虽然上述两种策略都取得了良好的效果, 但都忽略了不同任务、不同学习样本在学习过程中的困难差异。

为了解决这一不足, 第三类 MTL 是最近发展起来的, 被称为自我进度多任务学习 (Self-Paced Multi-Task Learning)。具体来说, 该方法采用了一种类似人类的学习机制, 将模型从最简单的样本和任务训练到最困难的样本和任务。因此, 此方法可实现改进的性能, 例如 SPMTL[13]和 spMMTL[14]。由于这些模型在 SPL 和 MTL 之间都具有很强的耦合性, 这将使 SPL 和 MTL 在有限的场景下的可伸缩性问题更加严重。

本文介绍了一个灵活的多任务学习框架——灵活自步多任务学习 (FSPMTL), 用以解决这些方法的可伸缩性问题。我们的 FSPMTL 模型包含两个阶段, 可以灵活地嵌入不同类型的 SPL 模型和 MTL 模型。具体来说, 使用名为 the Balanced Self-Paced Learning (BSPL) 的 SPL 机制[15]我们的 FSPMTL 模型首先根据样本难度选取每个任务的样本, 得到样本难度水平。然后利用最新的 MTL 模型对不同难度水平的样本进行迭代训练, 得到最终的模型。

本文的主要贡献概括如下:

据我们所知, 这是第一个提出 SPL 和 MTL 结合的通用框架的工作。

提出了一种与传统学习机制不同的渐进式多任务学习机制。

我们对合成数据集和真实数据集进行了广泛的实验, 以证明我们提出的框架的有效性。

二、相关工作

MTL 作为当前的研究热点之一，通过在相关任务之间共享信息，可以有效地提高模型的整体性能，提高模型的鲁棒性。传统的 MTL 方法假设不同任务的目标函数参数应该相似[16]，或者多个相关任务应该共享相同的特征子集[17]。早期的 MTL 方法试图使用正则项约束来最小化相关任务之间的差异。然而，这些方法容易产生负迁移。因此，最近对 MTL 的研究主要基于稀疏表示[18,19]。

Argyriou 等人[7]提出了一个 MTL-FEAT 模型，该模型通过学习多个任务之间的稀疏表示来共享信息。Kang 等人[3]放松了 MTL-FEAT 模型的约束，提出了 DG-MTL 模型。基于 MTL-FEAT 和 DG-MTL 模型，Kumar 等人[20]提出了 GO-MTL 模型，在任务间有选择地共享信息。随后，基于之前的模型，Jeong 等人[21]提出了 VSTG-MTL 模型，实现了变量选择与学习任务之间的变群结构。与以往的 MTL 模型相比，VSTG-MTL 模型大大提高了模型预测的性能。

我们知道，样本质量也可能影响模型的性能，这可以在模型学习过程中考虑。课程学习 (CL) 模仿人类的认知过程，支持学习算法遵循从简单示例到更难示例的逻辑学习顺序[22]。这种“从小做起”的策略与人类从小到大的知识获取过程非常相似，在多模态学习中也得到了有效的证明，例如[23,24]和半监督学习[25]。CL 通常是在两个框架下实现的：Self-Paced Learning (SPL) [26]以及教学与学习 (TLLT) [27,28]。

SPL 在[26]它以简单的样本开始训练过程，然后逐渐将更难的样本带入训练中。最近的研究表明，SPL 是一种有效的鲁棒学习机制，并取得了迅速的发展[29,30]，如 SPMoR[31]和 C-SPCL[32]。Jiang 等人[33]实验证明，SPL 既考虑了训练前已知的先验知识，又考虑了训练过程中的学习进度，可以避免陷入局部最优。最近，Ren 等人[15s]提出了求解 SPL 中常见的不平衡分类问题的 BSPL 模型。BSPL 模型可以从不同的类别标签中按比例选择训练数据，避免了样本数据的类别标签率因样本难度分布的不同而发生较大变化。

自 2017 年以来，人们开始利用 SPL 和 MTL 的好处。Li 等人[13]以及 Murugesan 等人[14]提出了一种将 MTL 与 SPL 紧密结合的方法，通过简单到困难的 MTL 取得了较好的效果。此外，SPMTL[13]尝试通过同时考虑每个任务的任务和实例的复杂性以及[14]将任务选择嵌入到基于共享知识的模型学习中。在模型中，通过同时学习难度等级系数和系数矩阵，同时优化了 SPL 和 MTL 的参数。然而，优化方法导致模型的可扩展性和灵活性较低。

因此，我们提出了一个名为 FSPMTL 的两阶段框架来解决这个问题。在第一阶段中，我们使用 SPL 模型获得样本难度矩阵 \mathbf{E} ，其元素表示每个任务中样本的难度。第二阶段根据系数矩阵 \mathbf{E} 选取新的训练样本，更新优化后的 MTL 模型，得到系数矩阵 \mathbf{W} 。更多细节将在下一节讨论。

三、灵活自我进度多任务学习

1. 多任务学习的变量选择与任务分组

假设存在 \mathcal{T} 监督学习任务，每个任务都包含 D 变量和 N_t 训练实例。在

$X_t = \left[(x_t^1)^T, \dots, (x_t^{N_t})^T \right]^T \in \mathbb{R}^{N_t \times D}$ ， $x_t^n \in \mathbb{R}^D$ ，输出向量 $y_t = [y_t^1, \dots, y_t^{N_t}]^T \in \mathbb{R}^{N_t}$ 。之后，

我们可以用线性模型来描述输入和输出之间的关系，

$$y_t^n = f(w_t^T x_t^n)$$

其中 f 是二进制分类问题的 logit 函数 $y_t^n \in \{-1, 1\}$ 和 $w_t^T \in \mathbb{R}^D$ 表示第 t 个任务的系数向量。然后， \mathcal{T} 任务的系数向量 w_t 生成系数矩阵 $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_T]$ 。

存在一个低维的潜在空间，使得系数矩阵 \mathbf{W} 可以用低阶因子分解和稀疏空间来表示。我们将 \mathbf{W} 表示为两个低秩矩阵 \mathbf{U} 和 \mathbf{V} 的乘积，即 $\mathbf{W} = \mathbf{UV}$ 。其中 $\mathbf{U} \in \mathbb{R}^{D \times M}$ 是变量潜在矩阵， $\mathbf{V} \in \mathbb{R}^{M \times T}$ 是潜在任务矩阵， $M \ll \min(D, T)$ ， M 是潜在基的个数。对于 t -th 任务， $\mathbf{w}_t = \mathbf{U}\mathbf{v}_t$ 。

这个问题的优化函数是，

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{t=1}^T \frac{1}{N_t} L(y_t, \mathbf{X}_t \mathbf{U} \mathbf{v}_t) + \gamma_1 \|\mathbf{U}\|_1 + \gamma_2 \|\mathbf{U}\|_{1,\infty} + \mu \sum_{t=1}^T (\|\mathbf{v}_t\|_k^{sp})^2$$

$L(\cdot, ?)$ 用于二进制分类问题； $\sum_{n=1}^{N_t} \log(1 + \exp(-y_t^n \mathbf{v}_t^T \mathbf{U}^T \mathbf{x}_t^n))$ 是损失函数，

$\|\mathbf{U}\|_1 = \sum_{d=1}^D \sum_{m=1}^M |\mathbf{u}_{dm}|$ 是 ℓ_1 范数； $\|\mathbf{v}_t\|_k^{sp}$ 是 k 支持规范；其中 γ_1, γ_2, μ 是正则化参数。

2. 平衡的自我进度学习

在本节中，我们将对单个任务使用 BSPL 模型来获得训练样本的样本难度矩阵 $\mathbf{E} \in \mathbb{R}^{L \times T \times N_t}$ ，其行向量表示为 $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_L]^T$ ，其列向量表示为 $\mathbf{E} = [\mathbf{e}^1, \mathbf{e}^2, \dots, \mathbf{e}^T]$ 。在 \mathbf{E} 中的每个元素 \mathbf{e}_l^t 都可以表示为 \mathbb{R}^{N_t} 的向量，其值范围是离散集 $\{0, 1\}$ 。

就第 t 个任务而言，BSPL 模型的目标是通过最小化，共同学习模型参数 θ_l^t ，该参数是决策函数 g 的参数和潜在样本困难变量 $\mathbf{e}_l^t = [e_l^{t,1}, \dots, e_l^{t,N_t}]$ ：

$$\min_{\theta_l^t, \mathbf{e}_l^t} \sum_{n=1}^{N_t} e_l^{t,n} L(y_t^n; g(\mathbf{x}_t^n, \theta_l^t)) + \mu R(\theta_l^t) - \sum_{k=1}^K \sum_{\substack{nk \\ \mathbf{x}_t^{nk} \in Cl_k}} \lambda_k e_l^{t,nk}$$

其中 $e_l^{t,n} \in \{0, 1\}$ 是一个正则化参数，可以表示为

$R(\theta_l^t) = \sum_{n=1}^{N_t} G_\xi \left[A \left((\theta_l^t)^T \tilde{\mathbf{x}}_t^n \right) \right] - A \left((\theta_l^t)^T \mathbf{x}_t^n \right)$ ，其中 $\tilde{\mathbf{x}}_t^n$ 是噪声特征和 $G_\xi[\cdot]$ 是根据某个

分布的期望值。在我们的方法中，我们将高斯噪声添加到 $R(\theta_l^t)$ 。函数 $A(\cdot)$ 取决于特定的损失函数。 μ 表示相应的系数， Cl_k 表示第 k 个类。使用固定的 θ_l^t ，全局最优 $\mathbf{e}_l^{t*} = [e_l^{t,1}, \dots, e_l^{t,N_t}]$ 可以通过以下规则计算，

$$e_l^{t,n*} = \begin{cases} 1, & \text{if } L_k(y_t^n; g(\mathbf{x}_t^n, \theta_l^t)) < \lambda_k \\ 0, & \text{otherwise} \end{cases}$$

其中 $L_k(y_t^n; g(\mathbf{x}_t^n, \theta_l^t))$ 表示第 k 类中实例的损失。

3. 更新过程

在第 l 层的训练过程中， \mathbf{E} 的第 l 行向量 \mathbf{e}_l 表示训练样本的样本难度向量。因此，我们可以使用以下策略来训练模型参数 \mathbf{E}, \mathbf{W} 。

算法中参数的更新过程一般分两个阶段完成。在第一阶段，通过求解方程 $\text{ref}\{\text{equE}\}$ ，我们可以通过 BSPL 模型得到第 t 个任务的 l 层中的样本难度系数 \mathbf{e}_l^t 。具体计算见第 1 步和第 2 步。在第二阶段，使用 $\{\mathbf{e}_l^t\}$ 选择的训练样本，VSTG-MTL 模型通过求解方程 $\{\text{eqw}\}$ 来训练 \mathbf{W}_l 。迭代地，当所有 L 层的训练过程完成时，通过计算 \mathbf{W}_1 到 \mathbf{W}_L 的期望值，得到最终的训练系数矩阵 \mathbf{W} 。具体计算如步骤 3 和 4 所示。

具体来说，算法更新过程可分为四个步骤。当初始化 L 、 θ_l 、 \mathbf{U}_l^{init} 和 \mathbf{V}_l^{init} 时，将迭代完成以下四个步骤。

步骤 1：固定 θ_l^t ，更新 \mathbf{e}_l^t 。

首先，对于第 t 个任务，我们固定了 θ_l^t ，然后解决以下问题以更新 \mathbf{e}_l^t ：

$$\begin{aligned} \mathbf{e}_l^{t*} &= \arg \min \sum_{n=1}^{N_t} e_l^{t,n} L(y_t^n; g(\mathbf{x}_t^n, \theta_l^t)) - \sum_{k=1}^K \sum_{\substack{nk \\ \mathbf{x}_t^{nk} \in Cl_k}} \lambda_k \mathbf{e}_l^{nk} \\ &= \arg \min \sum_{k=1}^K \sum_{\substack{nk \\ \mathbf{x}_t^{nk} \in Cl_k}} \mathbf{e}_l^{t,nk} \left(L(y_t^{nk}; g(\mathbf{x}_t^{nk}, \theta_l^{t,nk})) - \lambda_k \right) \end{aligned}$$

步骤 2：固定 \mathbf{e}_l^t ，更新 θ_{l+1}^t 。

对于第 t 个任务，我们固定了 \mathbf{e}_l^t ，并通过解决以下问题来更新 θ_{l+1}^t ：

$$\theta_{l+1}^* = \arg \min \left(\sum_{n=1}^{N_l} e_l^{t,n} L(y_l^n; g(\mathbf{x}_l^n, \theta_l^t)) + \mu R(\theta_l^t) \right)$$

我们使用梯度下降算法更新上面的方程，然后增加 λ_k ，并返回到步骤 1 进行迭代执行，

直到选中所有实例。然后，我们可以得到第 t 个任务的样本难度系数向量 \mathbf{e}_l^t 。

但所有任务训练完成后，我们可以得到 $\mathbf{e}_l = [\mathbf{e}_l^1, \mathbf{e}_l^2, \dots, \mathbf{e}_l^T]$ 和 $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_L]^T$ 。

步骤 3：固定 \mathbf{e}_l ，更新 \mathbf{U}_l 。

对于每个级别的培训任务，我们通过 \mathbf{e}_l 选择培训示例，即。

$$\mathbf{X}_l = \mathbf{X}(\mathbf{e}_l \text{ is equal to } 1)$$

类似地，我们选择对应于 \mathbf{y}_l 的标签 \mathbf{X}_l 。然后，我们使用乘法器和提前停止的交替方向

方法更新 \mathbf{U}_l 。目标函数如下：

$$\mathbf{U}_l = \arg \min_{\mathbf{U}_l} \left(\sum_{t=1}^T \frac{1}{N_j} L(\mathbf{y}_l^t, \mathbf{X}_l^t \mathbf{U}_l \mathbf{v}_l^t) + \gamma_1 \|\mathbf{U}_l\|_1 + \gamma_2 \|\mathbf{U}_l\|_{1,\infty} \right)$$

步骤 4：固定 \mathbf{U}_l ，更新 \mathbf{V}_l ：

我们使用梯度下降算法解决下列公式，同时， $\mathbf{V}_l = [\mathbf{v}_l^1, \dots, \mathbf{v}_l^T]$ ：

$$\mathbf{v}_l^t = \arg \min_{\mathbf{v}_l^t} \left(\sum_{t=1}^T \frac{1}{N_j} L(\mathbf{y}_l^t, \mathbf{X}_l^t \mathbf{U}_l \mathbf{v}_l^t) + \mu \sum_{t=1}^T \left(\|\mathbf{v}_l^t\|_k^{sp} \right)^2 \right)$$

接下来，算法迭代执行步骤 3 和 4，直到 \mathbf{U}_l 和 \mathbf{V}_l 收敛。之后，我们计算 $\mathbf{W}_l = \mathbf{U}_l \mathbf{V}_l$ 。

最后，它将重复 L 次，直到所有 L 级别的样本都经过训练，然后 \mathbf{W}_1 到 \mathbf{W}_L 的期望值可能会被视为最终的计算结果 \mathbf{W} 。

4. 自定进度多任务学习的灵活框架

综上所述，我们可以提取出一个通用的灵活的多任务自学习框架，即 FSPMTL 算法，如下面的算法 1。

TABLE 1: The Flexible Self-Paced Multi-Task Learning (FSPMTL) Algorithm

Algorithm 1 Flexible Self-Paced Multi-Task Learning (FSPMTL)	
Input: Dataset \mathbf{X} and \mathbf{y}	The number of data difficulty levels L
Output: Coefficient matrix \mathbf{W}	
1. for $t=1:T$ do	
2. for $l=1:L$ do	
3. select SPL model to train to get e_l^t	
4. end for	
5. end for	
6. $\mathbf{e}_l = [e_l^1, e_l^2, \dots, e_l^T]^T$	
7. $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_L]^T$	
8. for $l=1:L$ do	
9. select training samples \mathbf{X}_l and \mathbf{y}_l from \mathbf{X} and \mathbf{y} according to \mathbf{e}_l	
10. select MTL model for training based on \mathbf{X}_l and \mathbf{y}_l to obtain \mathbf{W}_l	
11. end for	
12. $\mathbf{W} = \mathbb{E}[\mathbf{W}_1, \dots, \mathbf{W}_L]$	
13. return \mathbf{W} .	

设 O_e 为一次解方程 5 和 6 的时间，而 O_w 为一次解方程 8 和 9 的时间。因此，SPL 部分所用的时间是 $O_e * L * T$ ，MTL 部分所用的时间是 $O_w * L$ 。从算法 1 我们可以看到基于 FSPMTL 算法的模型每次运行需要 $O_e * L * T + O_w * L$ 时间。此外，如算法 1 所示，FSPMTL 算法的收敛依赖于 SPL 和 MTL 部分的收敛，这意味着 FSPMTL 算法将在 SPL 模型选择所有实例并且 MTL 模型达到两个残差阈值后停止[21].

四、实验

在这一部分中，我们旨在验证 FSPMTL 算法在不同实验设置下的有效性。我们方法的 Matlab 实现在 URL: <http://yzhou.github.io/#code>.

1.实验设置

为了验证该框架的有效性，基于 VSTG-MTL 模型实现了 FSPMTL 算法，并命名为 FSP-VSTG-MTL。在此，我们将 FSP-VSTG-MTL 与以下方法进行了比较：

BSPL-STL method: 这是一种基于高斯噪声平衡自学习的单任务学习方法。

VSTG-MTL method: 它将模型中的权重矩阵分解为两个低秩矩阵的乘积。这些矩阵将

同时执行任务之间的特征选择和学习任务之间的重叠组结构。

spMMTL 是自调整平均多任务学习的缩写，该模型根据每个任务难度的距离来提取简单任务。

FSP-VSTG-MTL 的参数初始化分为两部分。对于 BSPL 部分，我们在第一次迭代期间选择了一半的数据点，然后在下一次迭代期间更新了 λ_k 并增加了 10%，即 $L = 6$ 。然后我们最初设置 $e_i^{l:n} = 1 (n = 1, \dots, N_i)$ ，并运行相应的分类算法进行 5 次迭代，以获得 θ_i 的估计值。

对于 VSTG-MTL 部分，从搜索网格 $\{1, 3, 5, 7\}$ 中选择潜在碱基的数目 M 。我们将第三个正则化参数 μ 设置为等于第一个正则化参数 γ_1 。正则化参数是从搜索网格 $\{2^{-10}, \dots, 2^3\}$ 中选择的。矩阵 \mathbf{W}_i^{init} 的初始估计由 logistic 回归算法实现。通过 \mathbf{W}_i^{init} 的奇异值分解，给出了 \mathbf{U}_i^{init} 和 \mathbf{V}_i^{init} 的初始估计。

对于实验数据集，我们首先以 9: 1 的比例随机选取数据集中的数据，得到训练集和测试集。在训练集中，我们使用五重交叉验证方法来获得模型输出。然后，对测试集进行预测，得到最终分类效果。我们确保每个培训和测试过程中使用的数据集在不同的模型中是一致的。我们重复每个病例 10 次，并报告平均结果。

2 合成数据集

我们生成了四个合成数据集，它们具有不同数量的 D 维变量和 T 任务。实例 \mathbf{x}_i^n 从标准正态分布 $N(0, 1)$ 中采样，响应为 $y_i^n = \text{sign}(\mathbf{w}_i^T \mathbf{x}_i^n + \xi_i^n)$ 。为了创建困难实例，我们通过设置 $\xi_i^n = \sigma_i^n \theta_i^n$ ，将不同的噪声添加到实例中，其中 σ_i^n 从正态分布 $N(0, 5)$ 绘制，并且 θ_i^n 服从 $N(0, 1)$ 。在 $\mathbf{W}^* = [\mathbf{w}_1^*, \dots, \mathbf{w}_T^*]$ 中的 D 次 $M = \text{rank}(\mathbf{W}) = 5$ $\mathbf{U}\mathbf{V}$ ， $\mathbf{U} \in R^{D \times M}$ ， $\mathbf{V} \in R^{M \times T}$ 。每个合成数据集在两个矩阵 \mathbf{U} 和 \mathbf{V} 的结构上都不同。

生成数据集 1：生成数据集 1 有 25 个维度变量和 20 个任务。对于 $r = 1, \dots, M$ ，潜在基 \mathbf{u}_r 只具有从 $3r - 2$ 到 $3r + 3$ 分量的非零值。非零值由正态分布生成。类似地，对于 $r = 1, \dots, M$ ，加权向量 $\mathbf{v}_{4r-3}, \dots, \mathbf{v}_{4r}$ 分量上只有非零值。最后四个加权向量 $\mathbf{v}_{4M-3}, \dots, \mathbf{v}_{4M}$ 只在第 $M - 1$ 和 M 分量上具有非零值。非零值是通过从 1 到 1.5 的均匀分布产生的。

生成数据集 2：生成数据集 2 有 50 个维度变量和 20 个任务。对于 $r = 1, \dots, M$ ，潜在

基 \mathbf{u}_r 只具有从 $8r-7$ 到 $8r+8$ 分量的非零值。类似地, \mathbf{V} 的生成方式与生成数据集 1 相同。非零值由与生成数据集 1 中使用的分布相同的分布生成。

生成数据集 3: 生成数据集 3 有 25 个维度变量和 40 个任务。 \mathbf{U} 的生成方式与生成数据集 1 相同。类似地, 对于 $r=1, \dots, M$, 加权向量 $\mathbf{v}_{8r-7}, \dots, \mathbf{v}_{8r}$ 在 r 和 $r+1$ 分量上只有非零值。最后四个加权向量 $\mathbf{v}_{8M-7}, \dots, \mathbf{v}_{8M}$ 只在第 $M-1$ 和 M 分量上具有非零值。非零值由与生成数据集 1 中相同的分布生成。

生成数据集 4: 生成数据集 4 有 50 个维度变量和 40 个任务。类似地, \mathbf{U} 的生成方式与生成数据集 2 相同, \mathbf{V} 的生成方式与生成数据集 3 相同。非零值由与生成数据集 1 中相同的分布生成。

实例数目变化: 为了验证实例数目对 MTL 模型学习性能的影响, 我们在参数学习中改变了实例数目。对于每个任务, 我们通过每次添加 25 个实例, 将每个数据集的实例总数从 50 个增加到 300 个。每个实验将重复 10 次, 并用 F1 分数的平均值报告结果。图 1 总结了上述四个合成数据集的实验结果。可见, 对于单任务学习 (BSPL-STL), 实例数的增加会显著提高学习性能。但是, 对于 MTL 方法, 性能不仅取决于任务实例的数量, 而且还取决于维度的数量。

此外, 与最先进的模型 (VSTG-MTL、spMMTL) 相比, 我们提出的 FSP-VSTG-MTL 在大多数情况下都更好。值得注意的是, 我们的模型在四个综合数据集上的性能分别比 VSTG-MTL 模型高 2.20%、3.28%、1.55% 和 1.00%。此外, 为了验证我们提出的框架的灵活性, 我们使用我们的灵活学习框架扩展了 spMMTL, 看看它是否可以进一步改进。

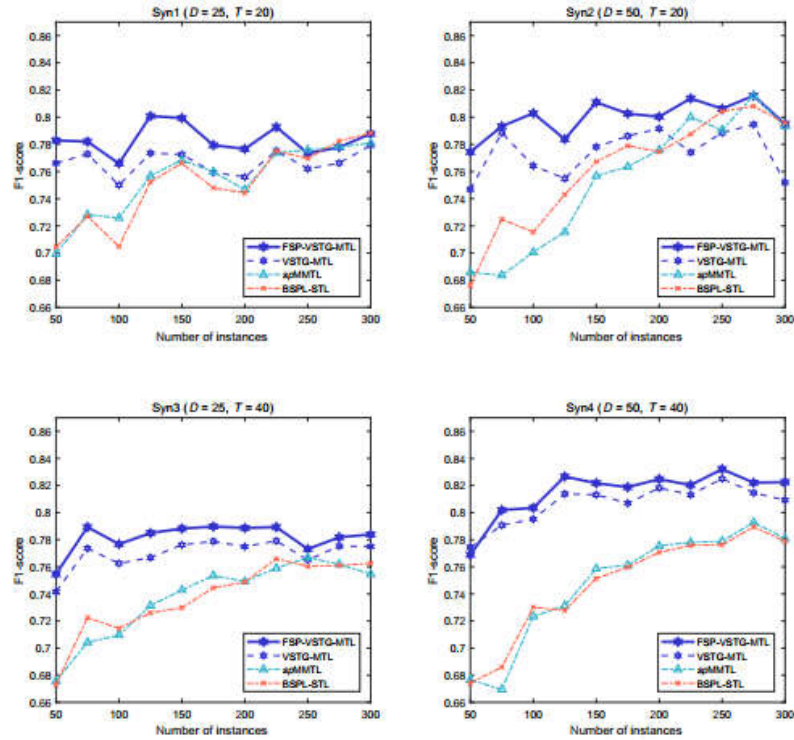


FIGURE 1: Results on four synthetic datasets with different number of instances, the dark blue solid line represents our proposed model.

所提出框架的灵活性测试: 为了进一步验证我们框架的灵活性, 我们将框架扩展到了

spMMTL 模型中, 并将其命名为 FSP-spMMTL 模型。这些模型的参数初始化是参照. 如图 1 所示, 实线总是比相应的虚线高。具体来说, 在四个综合数据集上, FSP-spMMTL 模型分别比 spMMTL 模型好 5.35%、7.92%、5.90%和 7.27%, 说明我们的灵活学习框架取得了很好的效果。因此, 在训练过程中加入 FSPMTL 框架可以有效地提高模型的性能。

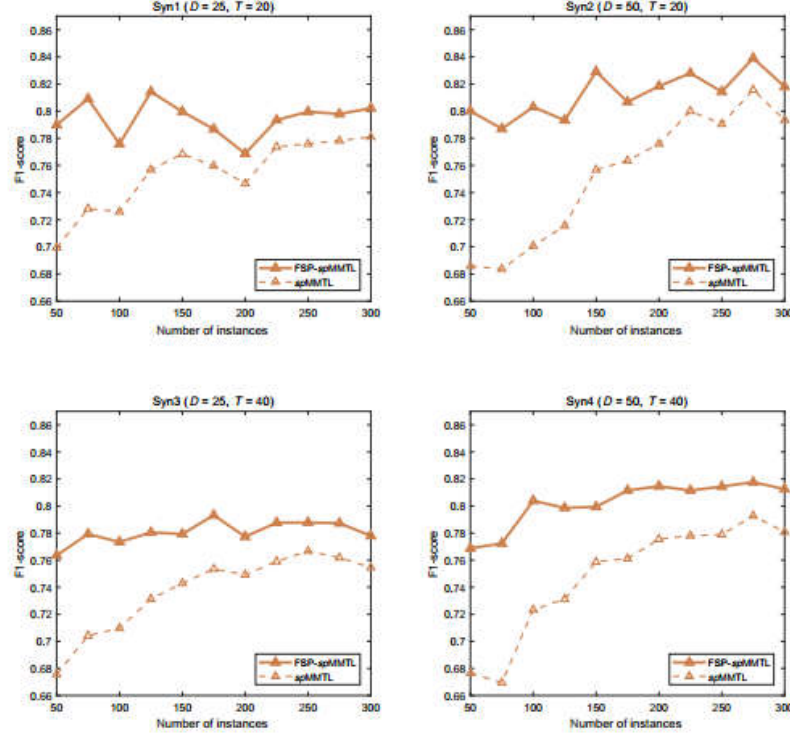


FIGURE 2: Results on four synthetic datasets with different number of instances, the solid lines represent the extended model FSP-spMMTL.

所提出框架的稳健性测试: 如果改变数据集上的噪声水平, 模型的学习难度也会随之改变。为了做到这一点, σ'' 的正态分布将从 $N(0, 1)$ 变为 $N(0, 10)$, 每次增加一个方差。同时, 我们为每个任务设置了 100 个实例。图 1 显示, 随着噪声的增加, 每个模型的性能都变得越来越差。当噪声保持在较低水平时, FSPMTL 框架对模型学习的影响很小。当 σ'' 的方差平均为 1-5 时, FSP-VSTG-MTL 的性能仅比 VSTG-MTL 好 1.03%、2.54%、1.28%和 -0.05%, 而 FSP-spMMTL 的性能比上述 4 个综合数据集上的 spMMTL 好 4.76%、10.14%、4.54%和 10.64%。然而, 随着噪声的增加, FSPMTL 框架的优点也逐渐显现出来。具体来说, 当 σ'' 的方差为 6-10 时, 平均而言, FSP-VSTG-MTL 的性能优于 VSTG-MTL 的 4.15%、6.44%、3.26%和 2.67%, 而 FSP-spMMTL 的性能优于 spMMTL 的 9.42%、13.91%、10.87%和 15.97%。结果证明了 FSPMTL 框架的优越性和灵活性。

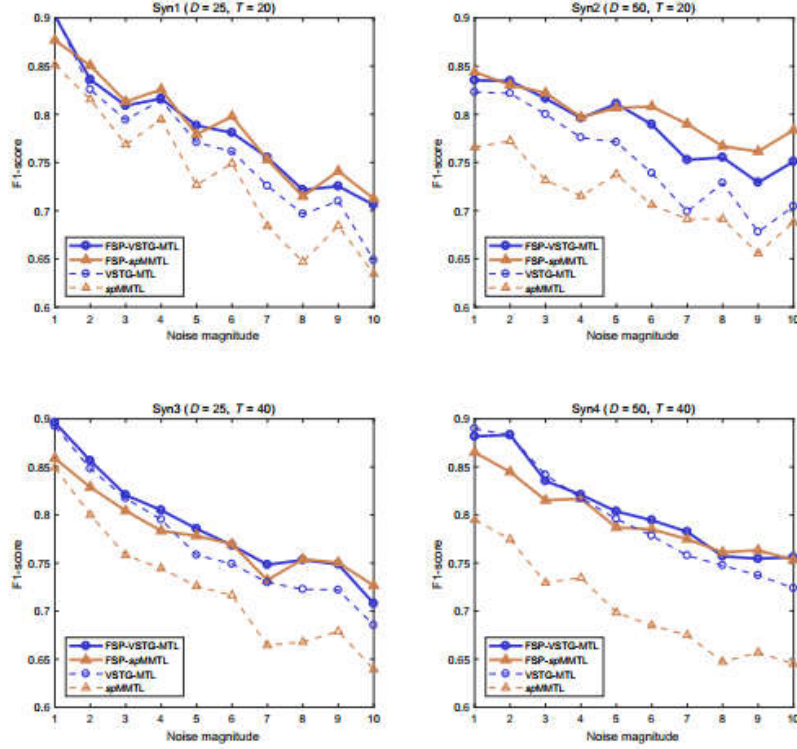


FIGURE 3: Results on four synthetic datasets with 10 different sample noises, two solid lines represent the extended models FSP-VSTG-MTL and FSP-spMMTL.

所选样本的可视化: 为了使我们的实验更加清晰,我们以第一个任务为例,使用我们的 FSP-VSTG-MTL 算法对所选样本进行可视化。当任务包含 100 个实例时,我们首先在 $l=1$ 时选择 50% 的样本,然后将每个级别的样本数增加 10%。对于选定的样本,我们使用主成分分析 (PCA) 将原始数据投影到一个较低维的子空间中,并将前两个维可视化。如图 4 所示,当 $l=1$ 时,正样本分布和负样本分布之间有一个清晰的边界。随着 l 的增加,所选样本的数量逐渐增加,二维样本分布的重叠区域也逐渐增加,这意味着所选样本越来越难以区分。这就是为什么我们的 FSPMTL 算法可以获得更高的分类精度和鲁棒性的原因。

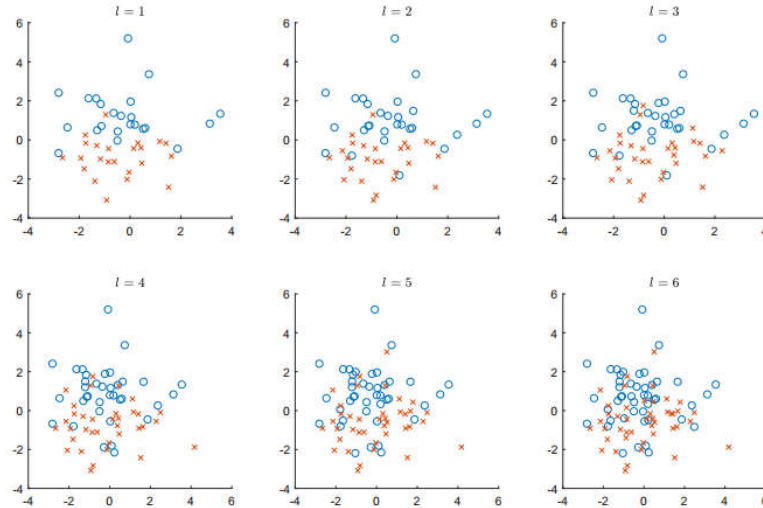


FIGURE 4: 2-D distribution of the selected samples in different levels with our FSP-VSTG-MTL algorithm in different sample difficulty levels. Here “o” represents positive samples and “x” represents negative samples in the classification problem.

3.真实世界数据集

London School Data(School)为了进一步验证该方法的可行性,我们使用由经典的 School 数据集生成的分类数据集进行了实验。school 数据集是伦敦教育局内部获得的回归数据集,包括 1985 年至 1987 年三年间伦敦 139 所中学 15362 名学生的考试成绩。该数据集包含 139 个任务和 15362 个观察结果,分别对应于不同学校和学生的测试。每项观察由 3 个连续变量和 23 个二元变量组成,代表学校和学生的专业属性。在这个实验中, school 数据集被离散化。得分高于 20 分的阳性样本 6984 份,占样本量的 45.46%,低于或等于 20 分的阴性样本 8378 份,占 54.54%。阳性和阴性样本的比例接近 1: 1。

TABLE 2: The F1-score of different methods on the real school dataset. The statistically best models are highlighted in bold.

Methods	BSPL-STL	VSTG-MTL	FSP-VSTG-MTL	spMMTL	FSP-spMMTL
F1-score	0.6269±0.0133	0.7234±0.0097	0.7272±0.0103	0.7138±0.0104	0.7234±0.0108

表 2 显示我们的模型在真实数据集上的结果。可以看出, FSP-VSTG-MTL 在 school 数据集中优于所有其他模型,从而证实了我们提出的 FSPMTL 框架的有效性。具体来说,BSPL-STL 是最新的单任务学习模型之一,但是它在处理 school 数据集方面很弱。近两年提出的 MTL 方法都取得了比单任务学习模型更好的分类预测效果。此外,本文提出的 FSP-VSTG-MTL 和 FSP-spMMTL 模型优于最新的 VSTG-MTL 和 FSPspMMTL 模型,分别为 0.53%和 1.39%。综上所述,通过将自主学习机制融入 MTL,我们的方法在这些实验中是有效的。

五、结论

本文提出了一种灵活的多任务学习框架,该框架采用松散耦合的方法将 MTL 模型与 SPL 模型结合起来。这样就可以灵活地将不同的 MTL 模型嵌入到 SPL 模型中。大量实验表明,该框架不仅有效地提高了传统模型的性能,而且提高了模型的灵活性和鲁棒性。在未来的工作中,我们希望在框架中引入先验知识,并将此方法应用到实际应用中。

参考文献

- [1] R. Caruana, “Multitask learning,” Machine Learning, vol. 28,no. 1, pp. 41–75, 1997. [Online]. Available: <https://doi.org/10.1023/A:1007379606734>
- [2] Y. Zhang and Q. Yang, “A survey on multi-task learning,” CoRR,vol. abs/1707.08114, 2017. [Online]. Available: <http://arxiv.org/abs/1707.08114>
- [3] Z. Kang, K. Grauman, and F. Sha, “Learning with whom to share inmulti-task feature learning,” in Proceedings of the 28th InternationalConference on Machine Learning, ICML 2011, Bellevue, Washington,USA, June 28 - July 2, 2011, 2011, pp. 521–528. [Online]. Available:https://icml.cc/2011/papers/344_icmlpaper.pdf
- [4] J. Pu, Y. Jiang, J. Wang, and X. Xue, “Multiple task learning usingiteratively reweighted least square,” in IJCAI 2013, Proceedings of the23rd International Joint Conference on Artificial Intelligence, Beijing,China, August 3-9, 2013, 2013, pp. 1607–1613. [Online].

Available:<http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6857>

[5] J. Pu, J. Wang, Y. Jiang, and X. Xue, "Multiple task learning with flexible structure regularization," *Neurocomputing*, vol. 177, pp. 242–256, 2016.

[Online]. Available: <https://doi.org/10.1016/j.neucom.2015.11.029>

[6] Z. Shi, P. Jian, Y. G. Jiang, F. Rui, and X. Xue, "Flexible multi-task learning with latent task grouping," *Neurocomputing*, vol. 189, no. C, pp. 179–188, 2016.

[7] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[8] Y. Zhang and D. Yeung, "Transfer metric learning by learning task relationships," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, July 25–28, 2010, 2010, pp. 1199–1208. [Online]. Available: <https://doi.org/10.1145/1835804.1835954>

[9] Y. Yang, J. Yang, J. Yan, S. Liao, D. Yi, and S. Z. Li, "Salient color names for person re-identification," in *Computer Vision - ECCV2014 - 13th European Conference*, Zurich, Switzerland, September 6–12, 2014, *Proceedings, Part I*, 2014, pp. 536–551. [Online]. Available: https://doi.org/10.1007/978-3-319-10590-1_35

[10] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," 2009.

[11] A. Schwaighofer, V. Tresp, and K. Yu, "Learning gaussian process kernels via hierarchical bayes," in *Advances in Neural Information Processing Systems 17* [Neural Information Processing Systems, NIPS2004, December 13–18, 2004, Vancouver, British Columbia, Canada], 2004, pp. 1209–1216. [Online]. Available: <http://papers.nips.cc/paper/2595-learning-gaussian-process-kernels-via-hierarchical-bayes>

[12] R. K. Ando and T. Zhang, "A framework for learning predictive structures from multiple tasks and unlabeled data," *Journal of Machine Learning Research*, vol. 6, no. Nov, pp. 1817–1853, 2005.

[13] C. Li, J. Yan, F. Wei, W. Dong, Q. Liu, and H. Zha, "Self-paced multi-task learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, February 4–9, 2017, San Francisco, California, USA., 2017, pp. 2175–2181. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14535>

[14] K. Murugesan and J. G. Carbonell, "Self-paced multitask learning with shared knowledge," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, Melbourne, Australia, August 19–25, 2017, 2017, pp. 2522–2528. [Online]. Available: <https://doi.org/10.24963/ijcai.2017/351>

[15] Y. Ren, P. Zhao, Z. Xu, and D. Yao, "Balanced self-paced learning with feature corruption," in *2017 International Joint Conference on Neural Networks, IJCNN 2017*, Anchorage, AK, USA, May 14–19, 2017, 2017, pp. 2064–2071. [Online]. Available: <https://doi.org/10.1109/IJCNN.2017.7966104>

[16] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 109–117.

[17] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in neural information processing systems*, 2007, pp. 41–48.

[18] Y. Zhou, J. Wang, C. Zhu, and W. Zhang, "Multiple dags learning with non-negative matrix factorization," in *Proceedings of the 3rd Workshop on Advanced Methodologies for Bayesian*

-
- Networks, AMBN 2017, Kyoto, Japan, September 20-22, 2017, 2017, pp. 81–92. [Online]. Available: <http://proceedings.mlr.press/v73/zhoul7a.html>
- [19] D. Zhang, J. Han, L. Yang, and D. Xu, “SPFTN: A joint learning framework for localizing and segmenting objects in weakly labeled videos,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 475–489, 2020. [Online]. Available: <https://doi.org/10.1109/TPAMI.2018.2881114>
- [20] A. Kumar and H. D. Iii, “Learning task grouping and overlap in multi-task learning,” *Computer Science*, vol. 2, 2012.
- [21] J. Jeong and C. Jun, “Variable selection and task grouping for multi-task learning,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2018, pp. 1589–1598. [Online]. Available: <https://doi.org/10.1145/3219819.3219992>
- [22] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, 2009, pp. 41–48. [Online]. Available: <https://doi.org/10.1145/1553374.1553380>
- [23] C. Gong, “Exploring commonality and individuality for multi-modal curriculum learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, 2017*, pp. 1926–1933. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14205>
- [24] C. Gong, J. Yang, and D. Tao, “Multi-modal curriculum learning over graphs,” *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 4, pp. 35:1–35:25, 2019. [Online]. Available: <https://doi.org/10.1145/3322122>
- [25] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, “Multimodal curriculum learning for semi-supervised image classification,” *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3249–3260, 2016.
- [Online]. Available: <https://doi.org/10.1109/TIP.2016.2563981>
- [26] M. P. Kumar, B. Packer, and D. Koller, “Self-paced learning for latent variable models,” in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada, 2010*, pp. 1189–1197. [Online]. Available: <http://papers.nips.cc/paper/3923-self-paced-learning-for-latent-variable-models>
- [27] C. Gong, D. Tao, J. Yang, and W. Liu, “Teaching-to-learn and learning-to-teach for multi-label propagation,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, 2016*, pp. 1610–1616. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11780>
- [28] C. Gong, D. Tao, W. Liu, L. Liu, and J. Yang, “Label propagation via teaching-to-learn and learning-to-teach,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 6, pp. 1452–1465, 2017. [Online]. Available: <https://doi.org/10.1109/TNNLS.2016.2514360>
- [29] K. Ghasedi, X. Wang, C. Deng, and H. Huang, “Balanced self-paced learning for generative adversarial clustering network,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 4391–4400. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Ghasedi_Balanced_Self-Paced_Learning_for_Generative_Adversarial_Clustering_Network_CVPR_2019_paper.html
- [30] K. Wang, Y. Wang, Q. Zhao, D. Meng, and Z. Xu, “Splboost: An improved robust boosting

algorithm based on self-paced learning,” CoRR, vol. abs/1706.06341, 2017. [Online]. Available: <http://arxiv.org/abs/1706.06341>

[31] L. Han, D. Zhang, D. Huang, X. Chang, J. Ren, S. Luo, and J. Han, “Self-paced mixture of regressions,” in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, 2017, pp. 1816–1822.

[Online]. Available: <https://doi.org/10.24963/ijcai.2017/252>

[32] D. Zhang, J. Han, L. Zhao, and D. Meng, “Leveraging prior knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework,” Int. J. Comput. Vis., vol. 127, no. 4, pp. 363–380, 2019. [Online]. Available: <https://doi.org/10.1007/s11263-018-1112-4>

[33] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, “Self-paced curriculum learning,” in Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA., 2015, pp. 2694–2700. [Online]. Available: <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9750>