



元自步学习

束俊¹, 孟德宇^{1,2*}, 徐宗本¹

1. 西安交通大学数学与统计学院, 西安 710049

2. 澳门科技大学澳门系统工程研究所, 澳门 999078

* 通信作者. E-mail: dymeng@mail.xjtu.edu.cn

收稿日期: 2020-01-06; 接受日期: 2020-03-01; 网络出版日期: 2020-06-10

国家自然科学基金(批准号: 61661166011, 11690011, 61603292, 61721002, U1811461)资助项目

摘要 自步学习是近年来机器学习领域提出的一种启发于人和动物“由易到难”学习过程的学习机制. 尽管自步学习已取得可喜的理论与应用进展, 但是当前的自步学习算法仍存在超参数选择的瓶颈问题. 针对该问题当前主要采用一些启发式的手工设计方法或者交叉验证方法, 然而此类方法效率很低, 缺乏理论性指导, 难以推广应用到广泛的实践问题中. 针对这一挑战性问题, 本文提出一种基于元学习机理的自步学习算法, 该方法能使自步学习中涉及的超参数以数据驱动的方式自动习得, 从而大大减弱了自步学习的这一核心问题. 特别地, 我们针对 3 种典型的自步学习实现格式, 将所提元学习策略实质性嵌入, 通过回归和分类实验验证了所提算法的准确性和泛化性, 特别验证了相比于传统超参设置方法的显著优越性.

关键词 自步学习, 元学习, 样本加权, 噪声标记下学习, 超参数选择

1 引言

课程学习^[1] 尝试模拟一般生物通常采用的“从易到难”学习机制, 其执行模式为预先将数据按照某种预设“课程”从易到难排序, 然后按此顺序将数据逐步输入学习模型进行增量式学习. 近年来, 课程学习策略被广泛应用于弱监督学习^[2]、多任务学习^[3,4]、迁移学习^[5]、多模态学习^[6]、自然语言理解^[7]等. 课程学习方法论因而逐渐引起机器学习领域更多的研究关注, 成为较为热点的研究课题.

虽然课程学习在一些应用中体现出良好的性能, 但其性能依赖于预先主观制定的课程质量, 这需要用户对领域拥有足够全面的先验知识. 对于现实中常见的缺乏充分领域先验认知的应用领域, 课程学习的有效执行仍面临着很大困难. 为了克服课程学习中自适应课程设置问题, 自步学习^[8] 提出将课程难易的度量准则嵌入到优化模型当中, 一边基于当前样本排序更新模型参数, 一边基于学习的效果对所有样本更新这一度量从而获得样本的新一轮难易排序, 最终达到自动化更新课程、自适应样本排序的目的.

引用格式: 束俊, 孟德宇, 徐宗本. 元自步学习. 中国科学: 信息科学, 2020, 50: 781-793, doi: 10.1360/SSI-2020-0005
Shu J, Meng D Y, Xu Z B. Meta self-paced learning (in Chinese). Sci Sin Inform, 2020, 50: 781-793, doi: 10.1360/SSI-2020-0005

考虑一个标准的监督学习问题: 给定训练数据集 $D = \{x_i, y_i\}_{i=1}^N$, 其中 $x_i \in \mathbf{X}$ 代表样本 (\mathbf{X} 表示样本空间), $y_i \in \mathbf{Y}$ 代表标记 (\mathbf{Y} 表示标记空间), N 是所有样本的数目. 如果 \mathbf{Y} 是连续的, 该问题为回归问题; 若 \mathbf{Y} 是离散的, 该问题为分类问题. 机器学习的目标是学习一个预测函数 $f_{\mathbf{w}}: \mathbf{X} \rightarrow \mathbf{Y}$, $\mathbf{w} \in \Gamma$, 其中 \mathbf{w} 为预测函数的参数. 获得此决策函数后, 我们即可对任意新样本进行预测. 该问题的求解一般需要构建如下的优化模型:

$$\min_{\mathbf{w} \in \Gamma} \sum_{i=1}^N \mathcal{L}_{\text{Train}}(f_{\mathbf{w}}(x_i), y_i) + R(\mathbf{w}), \quad (1)$$

其中 $\mathcal{L}_{\text{Train}}(\cdot, \cdot)$ 为训练数据集上的损失/误差项, 其作用在于度量预测与真实标记之间的差异程度, 常用的误差形式包括交叉熵损失 (分类)、均方误差 (回归) 等; $R(\mathbf{w})$ 为正则项, 其作用在于编码需要求解 \mathbf{w} 的预知先验信息.

自步学习的基本执行模式为: 在传统机器学习模型 (1) 中, 通过嵌入衡量每个样本难易程度或对学习目标重要性进行刻画的权重变量 v_i , 通过求解如下优化问题实现对该参数与模型参数 \mathbf{w} 的共同优化, 从而达到样本课程与学习模型共同学习的目的 [8]:

$$\min_{\mathbf{w} \in \Gamma, \mathbf{v} \in [0,1]^N} \sum_{i=1}^N (v_i \mathcal{L}_{\text{Train}}(f_{\mathbf{w}}(x_i), y_i) + g(v_i, \lambda)), \quad (2)$$

其中 λ 是年龄参数, $g(v, \lambda)$ 是自步正则, 用于编码自步学习由易到难的学习模式, 其定义及需要满足的条件可以参见文献 [9,10], 详细阐述见第 3 部分. 由于 $R(\mathbf{w})$ 对分析自步正则没有本质影响, 故在下文中省略该项. 在之后的介绍中, 为了简化符号表示, 我们也常常将第 i 个样本的误差 $\mathcal{L}(f_{\mathbf{w}}(x_i), y_i)$ 简记为 ℓ_i .

求解问题 (2) 可以采用交替迭代算法对参数 \mathbf{w} 和 \mathbf{v} 进行更新, 具体如下.

步骤 1. 固定参数 \mathbf{w}^* , 求解 \mathbf{v} , 即求解如下问题:

$$v_i^* = \arg \min_{v \in [0,1]} v_i \mathcal{L}_{\text{Train}}(f_{\mathbf{w}^*}(x_i), y_i) + g(v_i, \lambda), \quad i = 1, 2, \dots, n. \quad (3)$$

步骤 2. 固定参数 \mathbf{v}^* , 求解 \mathbf{w} , 即求解如下问题:

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \sum_{i=1}^N v_i^* \mathcal{L}_{\text{Train}}(f_{\mathbf{w}}(x_i), y_i). \quad (4)$$

步骤 3. 更新完 \mathbf{v} 与 \mathbf{w} , 增加 λ 的值, 使更多样本进入学习过程.

完整算法如算法 1 所示. 在算法的不断迭代过程中, 会将误差较大的标记数据作为不可靠性样本通过设置较小甚至零权重的方式将其在训练过程中的作用减弱甚至消除, 从而减小或者避免了此类样本对于分类器学习的负面影响. 进一步地, 可以证明, 在自步学习迭代优化样本权重与分类器参数的执行过程中, 其实存在一个以分类器参数为变量的隐藏目标函数 $F_{\lambda}(\ell)$, 也在同步单调下降 [11], 其定义为

$$F_{\lambda}(\ell) = \int_0^{\ell} v^*(\ell; \lambda) d\ell. \quad (5)$$

这一隐藏目标函数能够更为本质地说明自步学习这种算法模式的稳健性本质 [11], 如图 1 所示.

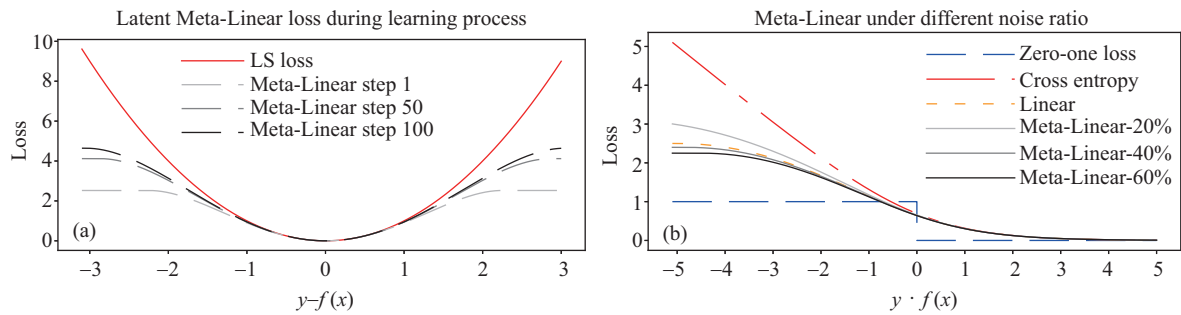


图 1 (网络版彩图) 元自步学习在学习过程中的隐式损失函数变化。(a) 和 (b) 分别展示回归和分类实验上的实验结果

Figure 1 (Color online) Variations of learned latent loss functions during learning process. (a) and (b) show the results of regression and classification experiments, respectively

算法 1 自步学习求解算法

输入: 数据 $\{x_i, y_i\}_{i=1}^N$, 最大迭代次数 T , 批样本数 n ;

输出: 模型参数 w ;

初始化: $w^0, v^0, \lambda_0, t = 0$;

- 1: if $t < T$ then
- 2: $t = t + 1$;
- 3: 求解 (3) 更新 $v^{(t)}$;
- 4: 求解 (4) 更新 $w^{(t)}$;
- 5: 增加 λ ;
- 6: end if

注意到算法中变量 v 和 w 的优化均可保证目标函数 (2) 的单调下降, 而 λ 的更新是基于由易到难的课程学习模式启发式制定的, 对其更新并不能保证目标函数 (2) 的递减性 (事实上, 往往反而导致该目标的递增效果). Meng 等^[11]指出, 在学习的后期阶段, 随着年龄参数的逐步增大, 异常值或者无意义的噪声样本会参与到训练过程中, 从而对模型学习效果产生损害. 针对这一问题, 文献中一般采用“中间截停”策略终止算法, 而选取终止的 λ 参数可以通过验证集的验证策略进行估计^[9]. 另一方面, 年龄参数 λ 在每一次迭代之后采用何种机制上升, 也是需要谨慎思考的问题. 如果将其设置的初值过大或增速过快, 则可能进入学习的训练样本过快过多, 从而导致无法充分执行课程学习从易到难的逐步学习模式, 也可能导致噪声或异常样本在学习过程中的过早侵入, 从而产生训练效果的偏差; 而如果将其设置的初值过小或增速过慢, 则可能进入样本过少, 导致算法学习速度过慢. 更为严重的是, 随着迭代次数的进行, 模型学习越来越准确, 导致所有样本的损失整体下降, 此时 λ 有可能并非要增大, 反而需要逐步减小. 为了克服这一超参数选择问题, Meng 等^[11]提出一种启发式的设定参数 λ 的方法, 即将 λ 选择转换为样本个数的选择, 通过逐次增加样本进入的个数, 使得课程参数的选择更加直观. 但是为了有效控制进入样本的个数变化, 往往需要了解数据更多的先验知识, 否则算法的鲁棒性和可重复性往往难以保证. 该超参数 (即年龄参数 λ) 的合理设定问题已成为自步学习面向真实应用的瓶颈问题.

为了解决上述问题, 本文提出一种新型的元自步学习算法. 该算法能够将改善年龄参数与优化模型参数放置于同一个优化框架中, 从而实现从数据中动态调节年龄参数的学习效果. 我们将 3 种经典的自步学习实现格式嵌入到所提的元学习算法中, 针对其构建了超参数的自动学习策略. 通过回归和分类实验, 验证了所提元学习算法相比于传统方法更优的学习效果, 从而说明了所提算法的有效性.

本文组织结构如下: 第 1 节介绍了自步学习及其年龄参数设定目前存在的问题, 第 2 节回顾自步学习及其超参数学习的相关工作, 第 3 节构建针对自步学习超参数自动选择的元学习算法, 并在 3 种典型的自步学习实现格式中本质嵌入, 第 4 节通过实验分析论证了所提算法的有效性, 第 5 节对本文进行了总结.

2 相关工作简介

2.1 自步学习

2009 年, Bengio 等^[1]首次提出了课程学习的概念. 其基本思想是通过组织训练样本从易到难逐步地输入学习模型, 从而达到良好的运算效果. 针对其课程标准自动化设置问题, 2010 年出现了自步学习^[8]的思想方法, 提出把用以课程排序的难易度量嵌入到优化模型, 自动学习该度的基本策略. 由于该类学习方法论的一般性和广泛适用性, 自步学习被广泛用于不同的任务, 如多媒体事件检测^[9]、显著性检测^[12]、物体检测^[13]、图片分类^[14]、物体追踪^[15]等, 并获得了良好的应用效果. 自步学习的本质也不断从深度和广度方面被推进. 例如, Jiang 等^[14]提出了自步课程学习, 通过老师 - 学生共同的教学 - 学习功能实现从易到难的课程学习目标. Meng 等^[11]证明了自步学习的迭代优化算法完全对应于一个鲁棒损失函数的 majorization-minimization (MM) 求解算法的迭代过程, 并在此后深入证明了自步学习收敛性的结论^[16]. 这项工作揭示了自步学习具有鲁棒学习的本质内涵. 进一步地, Liu 等^[17]阐明了自步学习模型与凹共轭形式的本质等价性, 并在此框架下, 诱导出更多一般形式的自步正则. 然而, 所有这些工作都未能对自步学习的超参数选择提供理论较为完善、具有广泛适用性的一般算法.

2.2 超参数优化

超参数优化试图针对一个优化模型的超参数进行自适应选择或学习, 从而使其获得良好的推广应用性能. 典型的超参数选择方法包含网格搜索、随机搜索^[18]、贝叶斯 (Bayes) 优化^[19]、交叉验证^[20]等. 最近, 基于元学习的超参数优化^[19~21]逐步引起关注并被普遍尝试使用. 例如, 基于元学习的策略, 针对样本权值这一复杂超参数出现了多种数据驱动学习策略^[22~24]. 实验验证了这种自适应的元学习策略的确能够有益于增强原始优化问题对于复杂数据偏差的鲁棒性与准确性. 本文可以视为该方法论在自适应学习自步学习的年龄参数的探索.

3 元自步学习

3.1 典型的自步学习算法

自步学习模型 (2) 中, 自步正则 $g(v, \lambda)$ 旨在诱导出一个从易到难的学习模式, 其严格数学化定义^[9,10]如下.

定义 1 (自步正则) 对于优化模型 (2), 当 $g(v, \lambda)$ 满足以下条件时, 称之为一个自步正则项:

- (1) $g(v, \lambda)$ 关于其变量 $v \in [0, 1]$ 为凸函数;
- (2) $v^*(\ell, \lambda)$ 为变量 ℓ 的单调递减函数, 且成立 $\lim_{\ell \rightarrow 0} v^*(\ell, \lambda) = 1, \lim_{\ell \rightarrow \infty} v^*(\ell, \lambda) = 0$;
- (3) $v^*(\ell, \lambda)$ 为变量 λ 的单调递增函数, 且成立 $\lim_{\lambda \rightarrow 0} v^*(\ell, \lambda) = 0, \lim_{\lambda \rightarrow \infty} v^*(\ell, \lambda) \leq 1$;

其中 $v^*(\ell, \lambda) = \arg \min_{v \in [0, 1]} v\ell + f(v, \lambda)$.

基于上述定义, 我们可以构造出很多体现数据内涵的自步正则格式. 这里我们给出 3 个典型的自步正则格式及其对应的权值求解公式, 如下:

$$f^H(v; \lambda) = -\lambda v, \quad v^*(\lambda; \ell) = \begin{cases} 1, & \text{if } \ell < \lambda, \\ 0, & \text{if } \ell \geq \lambda, \end{cases} \quad (6)$$

$$f^L(v; \lambda) = \lambda \left(\frac{1}{2} v^2 - v \right), \quad v^*(\lambda; \ell) = \begin{cases} -\ell/\lambda + 1, & \text{if } \ell < \lambda, \\ 0, & \text{if } \ell \geq \lambda, \end{cases} \quad (7)$$

$$f^M(v; \lambda) = \frac{\gamma^2}{v + \gamma/\lambda}, \quad v^*(\lambda, \gamma; \ell) = \begin{cases} 1, & \text{if } \ell < \left(\frac{\lambda\gamma}{\lambda + \gamma} \right)^2, \\ 0, & \text{if } \ell \geq \lambda^2, \\ \gamma \left(\frac{1}{\sqrt{\ell}} - \frac{1}{\lambda} \right), & \text{otherwise,} \end{cases} \quad (8)$$

其中 λ, γ 表示需要设置的超参数, 统一记为 Λ .

以上自步正则项分别构造于文献 [8~10] 中, 分别称为 Hard, Linear 和 Mixture 自步正则. 从上述权重求解公式可以看出, 当一个样本在当前模型参数下, 对应误差 ℓ 比年龄参数 λ 或者 λ^2 大时, 对应样本倾向于被认为是相对可靠性较低的难训练噪声样本, 其对应权重为 0, 其作用为消除这些样本对训练过程产生的影响; 反之, 对应误差 ℓ 比年龄参数 λ 或者 λ^2 小时, 这样的样本可被认为是分类可靠性较高的易训练相对纯净样本. 相比这些自步正则的加权格式, Hard 自步正则对每个样本要求选或者不选 (即“硬”加权), 而 Linear 自步正则形式赋予每个样本“软”加权以更为客观准确地反映样本的重要性程度, 而 Mixture 自步正则能够结合二者的优势, 对非常重要的一批样本赋予权重 1, 又对误差较小的样本赋予软加权, 从而有望能够更为充分体现样本重要性, 获得更好的学习效果.

3.2 元自步学习算法

正如第 1 节所述, 如何选取合适的年龄参数是影响自步学习学习效果及增强其可复现性的关键因素. 受当前元学习技术 [24~29] 的启发, 本文提出元自步学习方法, 旨在实现自适应调整自步学习超参数的功能.

具体地, 当我们在训练集 D 上优化自步学习目标函数 (2) 时, 因为训练数据集的分布形式 (即干净样本和噪声样本的分布) 未知, 传统的若干启发式解决方案, 特别是使用验证集进行参数评估验证的策略, 在现实的应用中存在诸多困难. 例如, 验证集该如何选择和充分利用, 自步学习迭代中止标准如何设定, 自步学习年龄参数该如何动态变化等. 为了克服这些问题, 元学习方法的基本思路为, 将传统验证集的收集标准提高, 对其质量进行严格要求, 从而保证其能够较为合理地体现数据背后真实分布的形态. 在这样的要求下, 该验证集就构成能够对训练数据进行良态引导的元数据集, 其一方面具有较小的量, 而另一方面对于问题相比训练数据具有更高的可靠性 [22~24].

具体方法步骤如下. 预先收集少量的元数据 (干净样本) $D_{\text{meta}} = \{x_i^{(m)}, y_i^{(m)}\}_{i=1}^M$, 通过在元数据集上的损失指导年龄参数的学习, 如下式所示, 其中 M 是元样本的个数, 且 $M \ll N$:

$$\Lambda^* = \arg \min_{\Lambda} \sum_{i=1}^M \mathcal{L} \left(f_{\mathbf{w}(\Lambda)}(x_i^{(m)}), y_i^{(m)} \right), \quad (9)$$

其中,

$$\mathbf{w}(\Lambda) = \arg \min_{\mathbf{w}} \sum_{i=1}^N (v_i \mathcal{L}(f_{\mathbf{w}}(x_i), y_i) + g(v_i, \Lambda)), \quad (10)$$

其中 \mathcal{L} 表示在训练数据上的损失函数, 本文在分类问题上用交叉熵误差, 回归问题上用均方误差.

因此, 元自步学习算法所优化的是一个双边优化问题^[25,30], 其中外目标函数是问题 (9), 内目标函数是问题 (2). 由于该问题很难获得准确结果, 借助文献 [23,24] 中采用的技术, 我们采取如下在线近似策略联合交替更新年龄参数集 Λ , 权重 \mathbf{v} 和模型参数 \mathbf{w} .

步骤 1. 固定参数 \mathbf{w}^* , 求解 Λ , 即优化目标函数 (9). 这里基于上一步的模型参数 $\mathbf{w}^{(t-1)}$ 和超参数 $\Lambda^{(t-1)}$, 采用随机梯度下降方法在 D_{meta} 上选取 m 个批样本求解, 即

$$\Lambda^{(t)} = \Lambda^{(t-1)} - \beta \sum_{i=1}^m \nabla_{\Lambda} \left(f_{\hat{\mathbf{w}}^{(t)}(\Lambda)}(x_i^{(m)}), y_i^{(m)} \right) \Big|_{\Lambda^{(t-1)}}, \quad (11)$$

其中 $\hat{\mathbf{w}}^{(t)}(\Lambda)$ 的形式通过在 D 上选取 n 个批样本按如下方式构建:

$$\hat{\mathbf{w}}^{(t)}(\Lambda) = \mathbf{w}^{(t-1)} - \alpha \sum_{i=1}^n \nabla_{\mathbf{w}} v_i(\Lambda) \mathcal{L}_{\text{Train}}(f_{\mathbf{w}}(x_i), y_i),$$

α, β 是随机梯度下降算法的学习率.

步骤 2. 基于上一步的模型参数 $\mathbf{w}^{(t-1)}$ 和更新获得的超参数 $\Lambda^{(t)}$, 求解 \mathbf{v} :

$$v_i^{(t)} = \arg \min_{v \in [0,1]} v_i \mathcal{L}_{\text{Train}}(f_{\mathbf{w}^{(t-1)}}(x_i), y_i) + g(v_i, \Lambda^{(t)}), \quad i = 1, 2, \dots, n. \quad (12)$$

针对我们选取的 3 种典型自步学习算法, 当模型参数 \mathbf{w} 固定后, 均具有闭式解形式. 此时只需要将 $\Lambda^{(t)}$ 带入式 (6) 中的权重求解公式, 即可得到 $\mathbf{v}^{(t)}$.

步骤 3. 基于得到的 $\mathbf{v}^{(t)}$, 求解 $\mathbf{w}^{(t)}$. 采用随机梯度下降方法, 在 D 上选取 m 个批样本进行求解, 即

$$\mathbf{w}^{(t)}(\Lambda) = \mathbf{w}^{(t-1)} - \alpha \sum_{i=1}^n \nabla_{\mathbf{w}} v_i^{(t)} \mathcal{L}_{\text{Train}}(f_{\mathbf{w}}(x_i), y_i) \Big|_{\mathbf{w}^{(t-1)}}. \quad (13)$$

完整元自步学习算法如算法 2 所示. 可以看出, 相比于原自步学习算法, 该算法通过元数据指导超参数 Λ 进行自适应的调整和学习. 这一算法可以很方便整合到任何分类或者回归模型当中, 因而具有一般的可用性. 特别地, 当自步正则采取 (6) 的 Hard, Linear 和 Mixture 形式时, 记对应的元自步学习算法分别为 Meta-Hard, Meta-Linear 和 Meta-Mixture.

算法 2 元自步学习求解算法

输入: 数据 $\{x_i, y_i\}_{i=1}^N$, 元数据集 $\{x_i^{(m)}, y_i^{(m)}\}_{i=1}^M$, 最大迭代次数 T , 批样本数 n 和 m ;

输出: 模型参数 \mathbf{w} ;

初始化: $\mathbf{w}^0, \mathbf{v}^0, \Lambda_0, t = 0$;

```

1: if  $t < T$  then
2:    $t = t + 1$ ;
3:   求解 (11) 更新  $\Lambda^{(t)}$ ;
4:   求解 (12) 更新  $\mathbf{v}^{(t)}$ ;
5:   求解 (13) 更新  $\mathbf{w}^{(t)}$ ;
6: end if

```

4 实验结果

本节将通过仿真回归实验和 CIFAR-10, CIFAR-100 数据集上的分类实验验证所提元自步学习算法的有效性.

表 1 仿真回归实验中各对比方法性能比较 (LS 误差)

Table 1 Performance comparison (LS error) on simulation experiment of all compared methods

Methods	LS	Hard	Meta-Hard	Linear	Meta-Linear	Mixture	Meta-Mixture
Test error	86.23	3.26	2.32	2.40	1.56	1.87	1.35

4.1 仿真回归实验

我们采用文献 [11] 中的实验设计生成仿真数据. 该数据集包含 1000 个数据点, 通过以下分布生成:

$$\sum_{i=1}^3 \pi_i p_i(X, Y) = p(X) p_i(Y|X),$$

其中,

$$\begin{aligned} p(X) &= U(\Omega_1), \quad \Omega_1 = \{x|x \in [-1, 1]\}, \\ \pi_1 &= 0.3, \quad p_1(Y|X) = N(f(X), 1), \\ \pi_2 &= 0.4, \quad p_2(Y|X) = \text{Laplacian}(f(X), 1), \\ \pi_3 &= 0.3, \quad p_3(Y|X) = U(\Omega_2), \quad \Omega_2 = \{\varepsilon|\varepsilon \in [f(X) - 20, f(X) + 20]\}, \end{aligned}$$

$f(x) = 0.5x + 1$, $N(\mu, \sigma^2)$ 为均值 μ 方差 σ^2 的高斯 (Gauss) 分布, $\text{Laplacian}(\mu, \beta)$ 表示均值为 μ 尺度为 β 的拉普拉斯 (Laplace) 分布. 此问题最优回归曲线为 $f(x)$. p_1, p_2 的作用为试图生成噪声, p_3 为生成异常值. 元数据集和测试集分别由真实回归曲线上随机生成的 1000 个点构成. 我们通过最小二乘回归 [20] 拟合曲线, 其中损失函数以最小二乘损失 (LS) 作为对比. 通过应用 Hard, Linear, Mixture, Meta-Hard, Meta-Linear 和 Meta-Mixture 的基本策略, 验证各自步学习算法的有效性.

表 1 给出了各个方法在测试集上的性能, 可以看出, 元自步学习算法相比于手工设计参数的原自步学习方法, 性能得以稳步提升, 其中 Meta-Mixture 算法获得了最佳的预测效果. 图 1(a) 描绘出训练过程中 Meta-Linear 的隐式损失函数随着迭代次数的变化曲线, 可以看出, 随着迭代次数的增加, 年龄参数逐步增大, 更多的难样本参与训练, 说明元自步学习算法可以通过自适应学习年龄参数近似实现了自步学习预期的由易到难的学习过程.

4.2 噪声标记分类实验

数据集. 我们选取 CIFAR-10 和 CIFAR-100 数据集用以评估各种自步学习算法在噪声标记数据集上分类的效果. 两个数据集由 32×32 大小的彩图组成, 分别包含 10 个类和 100 个类. 其训练集和测试集分别由 50000 张和 10000 张图片构成. 我们从其数据集中给定的验证集随机选取 1000 张图片作为元数据.

噪声设置. 在实验中采取两种类型的标记噪声: 对称噪声和非对称噪声. 对称噪声标记通过对给定比例的训练样本均匀转换到其他类别的标记上形成 [31]. 非对称噪声标记通过将给定比例的样本仅转换到特定类别的标记上构成 [24].

对比方法. 我们选取以下方法作为对比方法: (1) CE, 利用交叉熵损失在噪声数据集上训练分类器. (2) Forward [32] 通过标签的转移矩阵矫正噪声标记. (3) GCE [33] 通过引入额外的参数使得新设计的损失函数相比于 CE 具有噪声鲁棒性. (4) Meta-Weight-Net [24] 利用元数据通过单隐层的 MLP 学

表 2 所有对比方法在 CIFAR-10 和 CIFAR-100 上不同噪声比例下的测试准确率

Table 2 Test accuracy comparison on CIFAR-10 and CIFAR-100 with varying noise rates of all compared methods^{a)}

Datasets	Methods	Symmetric noise				Asymmetric noise	
		Noise rate η				Noise rate η	
		0	0.2	0.4	0.6	0.2	0.4
CIFAR-10	CE	92.89±0.32	76.83±2.30	70.77±2.31	63.21±4.22	76.83±2.30	70.77±2.31
	Forward	93.03±0.11	86.49±0.15	80.51±0.28	75.55±2.25	87.38±0.48	78.98±0.35
	GCE	90.03±0.30	88.51±0.37	85.48±0.16	81.29±0.23	88.55±0.22	83.31±0.14
	Meta-Weight-Net	92.04±0.15	89.19±0.57	86.10±0.18	81.31±0.37	90.33±0.61	87.54±0.23
	Hard	91.72±0.12	85.84±0.06	81.10±0.20	74.49±0.09	84.54±0.10	81.20±0.31
	Meta-Hard	92.20±0.35	88.77±0.23	85.74±0.23	80.67±0.61	87.14±0.15	85.21±0.35
	Linear	90.62±0.54	88.49±0.40	82.99±0.08	75.63±0.55	85.56±0.13	81.97±0.48
	Meta-Linear	91.71±0.36	89.71±0.27	86.61±0.30	82.12±0.22	89.89±0.14	86.94±0.23
	Mixture	91.51±0.16	88.63±0.18	83.77±0.33	77.77±0.38	88.52±0.37	82.58±0.45
	Meta-Mixture	91.90±0.15	89.78±0.30	86.82±0.13	82.20±0.12	90.41±0.24	87.62±0.05
CIFAR-100	CE	70.50±0.12	50.86±0.27	43.01±1.16	34.43±0.94	50.86±0.27	43.01±1.16
	Forward	67.81±0.61	63.75±0.38	57.53±0.15	46.44±1.03	63.28±0.23	57.90±0.57
	GCE	67.39±0.12	63.97±0.43	58.33±0.35	41.73±0.36	62.07±0.41	55.25±0.09
	Meta-Weight-Net	69.13±0.33	64.22±0.28	58.64±0.47	47.43±0.76	64.22±0.28	58.64±0.47
	Hard	68.68±0.11	58.13±0.31	53.57±0.57	39.96±0.24	60.54±0.56	51.04±0.87
	Meta-Hard	68.83±0.13	63.79±0.18	55.87±0.95	44.94±1.09	63.08±0.42	56.39±0.38
	Linear	68.07±0.23	60.13±0.14	55.13±0.71	43.28±0.59	62.11±0.72	53.73±0.70
	Meta-Linear	68.42±1.02	64.86±0.13	60.03±0.27	48.46±0.41	64.27±0.16	57.68±0.09
	Mixture	69.05±0.19	61.20±0.16	56.52±0.06	43.60±1.12	62.50±0.11	54.97±0.05
	Meta-Mixture	69.17±0.36	65.24±0.21	61.12±0.30	52.09±0.58	64.71±0.19	58.86±0.05

a) The best results are highlighted in bold.

习数据背后的加权格式, 代表着最先进的样本加权方法. (5) Hard, (6) Linear 和 (7) Mixture 表示典型的自步学习算法. 其中 (5)~(7) 将元数据作为验证集, 用来选择年龄参数.

网络结构及实验步骤. 我们使用 ResNet-32^[34] 作为分类器网络. 实验中, 使用 SGD 训练神经网络, 初始学习率为 0.1, 动量为 0.9, 权值衰减为 1×10^{-3} , 批次样本大小为 100. 网络一共训练 60 epoch, 其中在 40 epoch 和 50 epoch 学习率分别衰减至 0.01 和 0.001.

表 2 给出了 CIFAR-10 和 CIFAR-100 在对称和非对称噪声不同噪声水平下的测试集上的分类准确率. 可以看出: (1) 元自步学习算法通过自适应学习年龄参数, 可以稳步地提升通过验证集启发式获得年龄参数的方法, 并且基本上针对不同数据集及不同噪声水平都有显著提升. (2) Meta-Mixture 和 Meta-Linear 性能超过 Meta-Weight-Net, 主要因为前者明确假设加权函数单调下降, 对噪声的压制更加明显, 而 Meta-Hard 因为硬加权, 所以容易出现将部分脏样本误判为干净样本的偏差, 从而导致性能下降. (3) Meta-Mixture 在噪声情形下性能超过其他所有对比方法, 包括 Meta-Linear. 相比于 Meta-Linear, Meta-Mixture 会对部分非常置信的样本赋予权重 1, 而 Meta-Linear 会对这些样本赋予小于 1 的权重. 这使得 Meta-Mixture 更可能充分地获取样本蕴含的有效信息, 从而得到更为准确的分类结果.

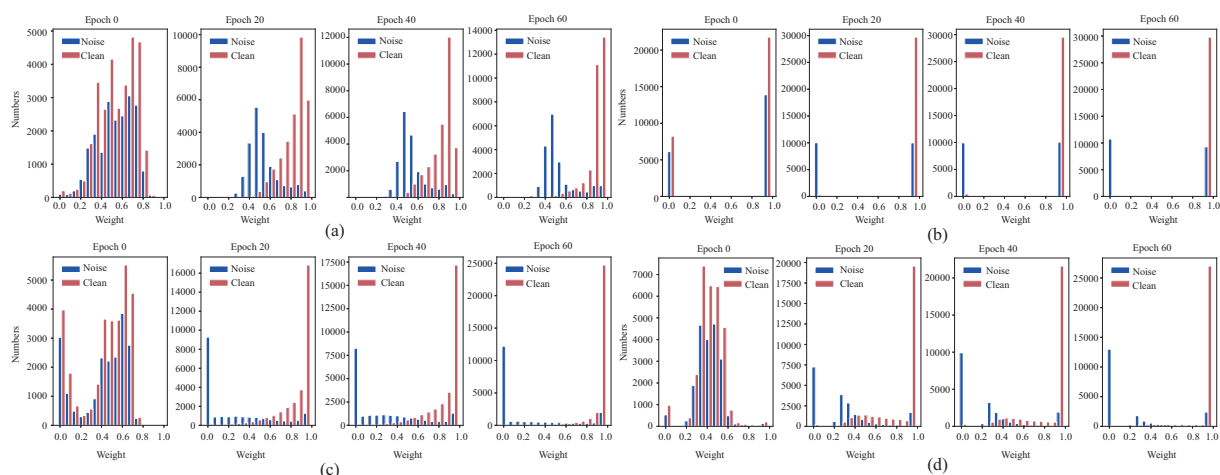


图 2 (网络版彩图) 在 40% 噪声比例下, 采用不同方法在 CIFAR-10 上随着训练过程样本权重分布的变化趋势展示. (a)~(d) 分别表示由 Meta-Weight-Net, Meta-Hard, Meta-Linear 和 Meta-Mixture 生成的样本权重分布

Figure 2 (Color online) Illustrations of sample weight distribution's variations during learning process on CIFAR-10 40% noise among different methods. (a)~(d) are generated by Meta-Weight-Net, Meta-Hard, Meta-Linear and Meta-Mixture, respectively

4.3 理解元自步学习算法

元自步学习如何自适应处理噪声. 为了理解元自步学习算法自适应拟合噪声水平的基本原理, 我们画出 CIFAR-10 数据集上不同噪声比例下学习到的隐藏目标函数 (如式 (5) 定义), 如图 1(b) 所示. 可以看出: (1) 当损失函数值较小时, 隐藏目标函数基本和 CE 损失一致; 而当损失函数值较大, 超过某个阈值, 隐藏目标函数会变得平坦, 从而对标记错误的样本 (通常损失值较大) 产生压制作用. (2) 当噪声水平增加时, 隐藏目标函数更容易提前变得平坦. 这意味着元自步学习算法可以根据噪声水平的强弱自适应调整损失函数的形态, 使其能够更好地编码并消除噪声标记样本的影响. 这种根据对噪声水平具有自适应性的学习能力也进一步解释了元自步学习在实验中表现出更好鲁棒性效果的本质原因.

样本加权机制可视化. 为了更好地理解元自步学习算法怎样学习更加鲁棒的模型, 我们分别绘制训练过程中 CIFAR-10 和 CIFAR-100 数据集上正确标记样本和噪声标记样本学习的权重分布变化, 如图 2 和 3 所示. 另外, 我们也将 Meta-Weight-Net 的权重分布变化作为比较. 可以看出, 相比于 Meta-Weight-Net, Meta-Linear 和 Meta-Mixture 可以更好地区分正确标记样本和噪声标记样本, 从而自然诱导更加鲁棒的学习效果, 而由于 Meta-Hard 学习的是硬权重, 导致部分噪声标记样本被误判为正确标记样本, 从而引起性能下降. 另一方面, 由于 Meta-Mixture 允许更多的样本权重为 1, 相比于 Meta-Linear, 更多正确标记样本被赋予更大的权重, 因此 Meta-Mixture 性能会略高于 Meta-Linear. 总体来说, 在元自步学习算法迭代过程中, 正确标记样本学习的权重会逐步变大, 而噪声标记样本学习的权重会逐步变小, 所以噪声标记样本的影响会随着迭代逐步减弱.

加权机制是否由易到难? 传统自步学习算法在设置年龄参数变化机制时, 通常采用由易到难的基本设定, 即通过逐步增加年龄参数的值来引导学习过程. 但是最优样本的加权机制真的必须采用“由易到难, 由少到多”的学习机制吗? 为了说明该问题, 我们以 Meta-Linear 算法为例, 在 CIFAR-10 数据集 40% 对称噪声下, 通过改变年龄参数的初值, 观察学习过程中年龄参数及样本权重统计的变化趋

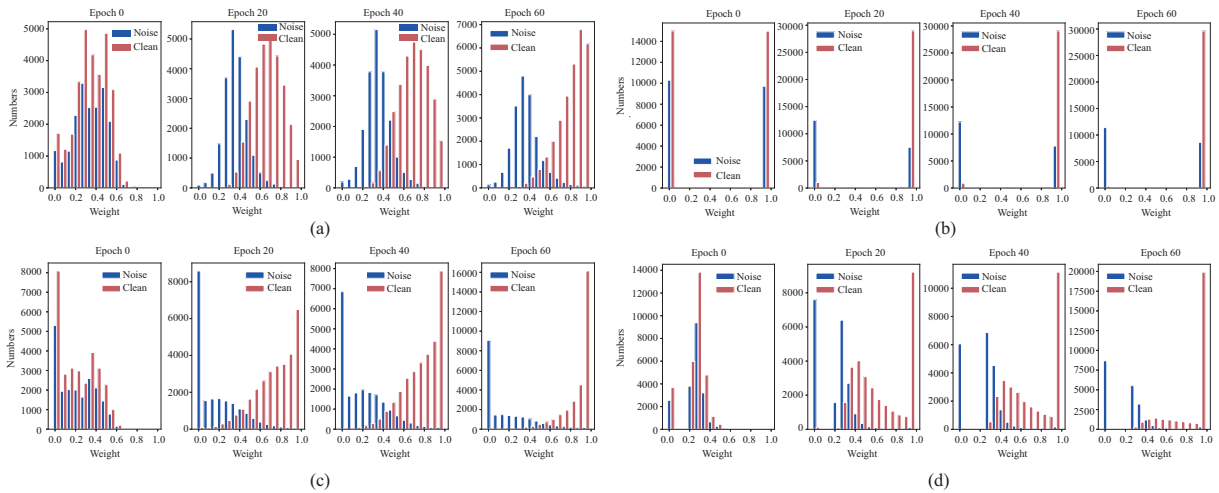


图 3 (网络版彩图) 在 40% 噪声比例下, 采用不同方法在 CIFAR-100 上随着训练过程样本权重分布的变化趋势展示. (a)~(d) 分别表示由 Meta-Weight-Net, Meta-Hard, Meta-Linear 和 Meta-Mixture 生成的样本权重分布

Figure 3 (Color online) Illustrations of sample weight distribution’s variations during learning process on CIFAR-100 40% noise among different methods. (a)~(d) are generated by Meta-Weight-Net, Meta-Hard, Meta-Linear and Meta-Mixture, respectively

表 3 不同初值下 Meta-Linear 在具有 40% 对称噪声 CIFAR-10 数据集上的测试准确率

Table 3 Test accuracy comparison on CIFAR-10 with 40% symmetric noise of different initial values

Initial values	$\lambda = 4$	$\lambda = 5$	$\lambda = 6$	$\lambda = 7$	$\lambda = 8$	$\lambda = 9$	$\lambda = 10$
Test accuracy (%)	85.34	86.10	86.83	86.75	86.54	85.93	85.64

势, 如图 4 所示, 其对应的在测试集的准确率如表 3 所示. 可以看出, 算法迭代过程中, 年龄参数整体遵循先升后降的变化趋势. 当年龄参数初值较小时, 需要更多的 epoch 上升, 而当其初值变大时, 上升以后很快下降, 在训练的后后期阶段逐渐趋于收敛. 而从其对应的样本权重, 可以看出, 当初值较小时, 对学习起到重要作用的样本 (权重大于 0.7) 基本上遵循先快速增加后趋于稳定的模式, 最后接近干净样本的个数 (约等于 3000), 而当年龄参数初值较大时, 会遵循先快速增加后下降直至趋于稳定的模式, 这是由于迭代起始较大的年龄参数导致部分噪声样本进入训练过程, 从而后期算法又逐渐将部分此类异常样本剔除. 这也解释了表 3 初值过大会带来性能轻微下降的本质. 这能够说明, 年龄参数的变化并不能完全反应学习的难易变化, 传统启发式设置的年龄参数逐渐递增的模式并非一定合理的策略. 而这里所构建的元自步学习可以通过自适应根据学习效果不断更改策略, 当一开始进入样本过少时, 学习会遵循由易到难模型, 并最终趋于稳定, 避免噪声样本的进入破坏学习效果; 而当一开始进入样本过多时, 一开始学习时会有部分噪声样本进入学习, 在学习后期, 模型试图自适应压制这些样本的作用, 保证最后学习效果尽可能鲁棒.

5 结论

针对原始自步学习算法超参数 (年龄参数) 选择困难的瓶颈问题, 本文提出了元自步学习算法的

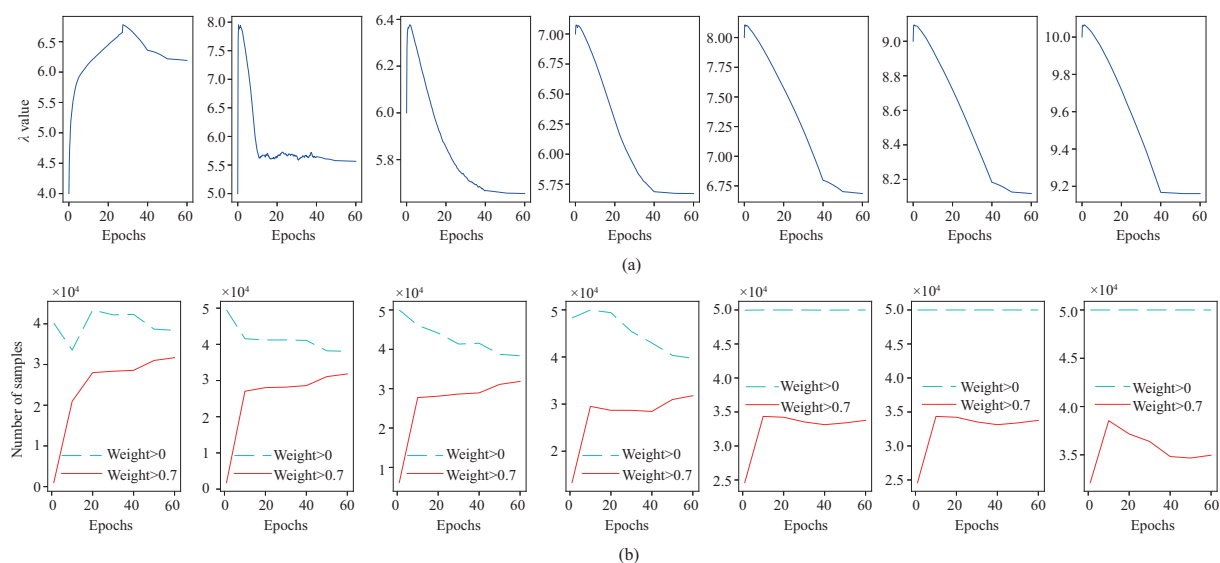


图 4 (网络版彩图) 在 40% 噪声比例下, Meta-Linear 算法在 CIFAR-10 上随着训练过程学习到的年龄参数 (a) 及实质进入训练过程的样本个数 (b) 统计变化趋势, 每个子图表示不同初值的学习效果, 从左到右初值依次为 4, 5, 6, 7, 8, 9, 10. 其中, (b) 的每个子图中展现的两条曲线分别对应样本权值大于 0 与样本权值大于 0.7 的样本个数变化趋势

Figure 4 (Color online) Varying trend of learned age parameter during learning process (a) and the number of essential samples involved in training (b) on CIFAR-10 40% noise of Meta-Linear method, where each subfigure shows the results of different initial values, including 4, 5, 6, 7, 8, 9, 10 in order from left to right. Two curves of (b) represent varying trend of the number of samples whose sample weights are greater than 0 and 0.7, respectively

基本策略. 通过元数据的指导, 所提算法能够自适应地学习自步学习中的年龄参数. 通过选取 3 种典型的自步学习算法将其嵌入到所提元自步学习算法, 其实验效果相比原自步学习方法呈现显著提升, 因而验证了所提元学习算法的有效性. 下一步工作我们将考虑将这种自适应学习超参数的学习范式应用于更多的机器学习问题中, 提升其有效应用的范畴.

参考文献

- 1 Bengio Y, Louradour J, Collobert R, et al. Curriculum learning. In: Proceedings of International Conference on Machine Learning, 2009. 41–48
- 2 Guo S, Huang W, Zhang H, et al. Curriculumnet: weakly supervised learning from large-scale web images. In: Proceedings of European Conference on Computer Vision, 2018. 135–150
- 3 Graves A, Bellemare M G, Menick J, et al. Automated curriculum learning for neural networks. In: Proceedings of International Conference on Machine Learning, 2017. 1311–1320
- 4 Pentina A, Sharmanska V, Lampert C H. Curriculum learning of multiple tasks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2015. 5492–5500
- 5 Shu Y, Cao Z, Long M, et al. Transferable curriculum for weakly-supervised domain adaptation. In: Proceedings of the 33th AAAI Conference on Artificial Intelligence, 2019. 4951–4958
- 6 Gong C, Tao D, Maybank S J, et al. Multi-modal curriculum learning for semi-supervised image classification. IEEE Trans Image Process, 2016, 25: 3249–3260
- 7 Turian J, Ratino L, Bengio Y. Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010. 384–394
- 8 Kumar M P, Packer B, Koller D. Self-paced learning for latent variable models. In: Proceedings of Conference and

- Workshop on Neural Information Processing Systems, 2010. 1189–1197
- 9 Jiang L, Meng D, Mitamura T, et al. Easy samples first: self-paced reranking for zero-example multimedia search. In: Proceedings of ACM International Conference on Multimedia, 2014. 547–556
 - 10 Zhao Q, Meng D, Jiang L, et al. Self-paced learning for matrix factorization. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, 2015. 3196–3202
 - 11 Meng D, Zhao Q, Jiang L. A theoretical understanding of self-paced learning. *Inf Sci*, 2017, 414: 319–328
 - 12 Zhang D, Meng D, Han J. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE Trans Pattern Anal Mach Intell*, 2016, 39: 865–878
 - 13 Dong X, Zheng L, Ma F, et al. Few-example object detection with model communication. *IEEE Trans Pattern Anal Mach Intell*, 2018, 41: 1641–1654
 - 14 Jiang L, Meng D, Zhao Q, et al. Self-paced curriculum learning. In: Proceedings of the Association for the Advance of Artificial Intelligence, 2015. 2694–2700
 - 15 Supancic J S, Ramanan D. Self-paced learning for long-term tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2013. 2379–2386
 - 16 Ma Z, Liu S, Meng D, et al. On convergence properties of implicit self-paced objective. *Inf Sci*, 2018, 462: 132–140
 - 17 Liu S Q, Ma Z L, Meng D Y, et al. Understanding self-paced learning under concave conjugacy theory. *Commun Inf Syst*, 2018, 18: 1–35
 - 18 Bergstra J, Bengio Y. Random search for hyper-parameter optimization. *J Mach Learn Res*, 2012, 13: 281–305
 - 19 Snoek J, Larochelle H, Adams R P. Practical bayesian optimization of machine learning algorithms. In: Proceedings of Conference and Workshop on Neural Information Processing Systems, 2012. 2951–2959
 - 20 Bishop C M. *Pattern Recognition and Machine Learning*. Berlin: Springer, 2006
 - 21 Maclaurin D, Duvenaud D, Adams R. Gradient-based hyperparameter optimization through reversible learning. In: Proceedings of International Conference on Machine Learning, 2015. 2113–2122
 - 22 Jiang L, Zhou Z, Leung T, et al. MentorNet: learning data-driven curriculum for very deep neural networks on corrupted labels. In: Proceedings of International Conference on Machine Learning, 2018. 2309–2318
 - 23 Ren M, Zeng W, Yang B, et al. Learning to reweight examples for robust deep learning. In: Proceedings of International Conference on Machine Learning, 2018. 4331–4340
 - 24 Shu J, Xie Q, Yi L, et al. Meta-weight-net: learning an explicit mapping for sample weighting. In: Proceedings of Conference and Workshop on Neural Information Processing Systems, 2019. 1917–1928
 - 25 Franceschi L, Frascioni P, Salzo S, et al. Bilevel programming for hyperparameter optimization and meta-learning. In: Proceedings of International Conference on Machine Learning, 2018. 1563–1572
 - 26 Schmidhuber J. Learning to control fast-weight memories: an alternative to dynamic recurrent networks. *Neural Comput*, 1992, 4: 131–139
 - 27 Shu J, Xu Z, Meng D. Small sample learning in big data era. 2018. ArXiv: 1808.04572
 - 28 Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of International Conference on Machine Learning, 2017. 1126–1135
 - 29 Thrun S, Pratt L. *Learning to Learn*. Berlin: Springer, 1998
 - 30 Colson B, Marcotte P, Savard G. An overview of bilevel optimization. *Ann Oper Res*, 2007, 153: 235–256
 - 31 Zhang C Y, Bengio S, Hardt M, et al. Understanding deep learning requires rethinking generalization. In: Proceedings of International Conference on Learning Representations, 2017
 - 32 Patrini G, Rozza A, Menon A K, et al. Making deep neural networks robust to label noise: a loss correction approach. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2017. 1944–1952
 - 33 Zhang Z L, Sabuncu M. Generalized cross entropy loss for training deep neural networks with noisy labels. In: Proceedings of Conference and Workshop on Neural Information Processing Systems, 2018. 8778–8788
 - 34 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778

Meta self-paced learning

Jun SHU¹, Deyu MENG^{1,2*} & Zongben XU¹

1. *School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China;*

2. *Macau Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China*

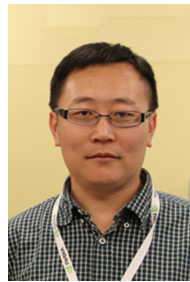
* Corresponding author. E-mail: dymeng@mail.xjtu.edu.cn

Abstract Self-paced learning (SPL) is a learning regime, inspired by human and animal learning processes, that gradually incorporates simple to more complex samples into a training dataset. Recently, SPL has seen significant research progress. However, current SPL algorithms still have critical limitations, such as how to determine the age hyper-parameters (especially the age parameters). Some heuristic strategies based on cross-validation have been designed. In addition, setting these parameters manually has been proposed. However, such strategies are very inefficient, not supported by theoretical evidence, and are very difficult to apply generally in practice. To address these issues, we propose a meta-learning regime for adaptively learning age parameters involved in SPL. Three types of typical SPL algorithms are integrated into the proposed regime, and their accuracy and generalization capability are substantiated through regression and classification experiments, and compared to conventional SPL paradigms that do not include adaptive age parameter tuning.

Keywords self-paced learning, meta learning, sample re-weighting, learning with noisy labels, hyper-parameter tuning



Jun SHU is currently a Ph.D. candidate at Xi'an Jiaotong University, under the supervision of Prof. Deyu MENG and Prof. Zongben XU. His current research interests include machine learning and computer vision, particularly small sample learning, learning to learn, and meta learning.



Deyu MENG is currently a professor with the Institute for Information and System Sciences, Xi'an Jiaotong University, and a guest professor with Macau Institute of Systems Engineering, Macau University of Science and Technology. From 2012 to 2014, he was a visiting scholar at Carnegie Mellon University. His current research interests include self-paced learning, noise modeling, tensor sparsity, and meta learning.



Zongben XU is a member of the Chinese Academy of Sciences, and the chief scientist of the National Basic Research Program of China (973 Project). His current research interests include intelligent information processing and machine learning.