论文题目：自步学习研究综述

学生姓名：刘仕琪

指导教师：孟德宇

摘　要

大数据时代下，大量的信息包含于海量的不完美数据中。为了从这些数据中获得有用的信息，机器学习方法需要具有自适应筛选有效数据并从中学习的能力。针对这一问题，近年来通过借鉴人类自适应的从易到难的课程学习原理，出现了自步学习这一新型学习模式。自步学习主要在半监督学习，弱监督学习，稳健学习，极少样本学习等机器学习领域上取得了快速发展和广泛应用。该论文调研了最近自步学习在主流机器学习领域上取得的成果和相关应用。论文回顾了课程学习和自步学习这两种基本学习模式，进一步总结了自步学习在自步课程学习，多样性自步学习，自步协同训练等方法中，所利用的内在先验知识和外在的课程先验知识。本文介绍了自步学习的优化理论的进展，并利用自步学习的凹共轭性将现有关于自步学习优化理解的研究统一化。特别的，我们在该论文里首次引入自步学习的概率理解，基于信息论得到了自步学习最优表现的上界形式和相应的影响因子，并提出了自步学习通用的概率框架。本论文最后展望了所提出的自步学习概率理论在封闭和开放学习环境下的发展。自步学习概率理论有望为设计出解决自步学习难易判定的有效方案、利用自步学习实现新的样本筛选方法、自步学习理论和其他学习理论的融合等提供启发，这有益于进一步推广自步学习在实际问题中的应用。

关键词：自步学习；课程学习；机器学习；人工智能

Title: A Survey On Self-paced Learning

Name: Shiqi Liu

Supervisor: Deyu Meng

ABSTRACT

In the era of big data, a lot of information is contained in a large amount of imperfect data. In order to extract useful information from those data, machine learning algorithms should be provided with the abilities of both data selection and data learning. To solve this problem, drawing lessons from the principle of human curriculum learning, which is to learn from simple to difficult adaptively, a new learning regime called self-paced learning(SPL) has appeared in recent years. SPL has developed rapidly and made extensive applications in the fields of machine learning, mainly including semi-supervised learning, weakly supervised learning, robust learning, and very few samples learning. This paper surveys recent achievements and related applications of SPL in machine learning fields. We review the classical curriculum learning and SPL regimes, further, summarize the internal prior knowledge utilized by the SPL and external curriculum prior knowledge utilized by the self-paced curriculum learning, SPL with diversity and self-paced co-training. Current advancements of optimization theory regarding SPL are introduced and united together under the concave conjugacy of SPL. In particular, we first raise the probability viewpoint for SPL, obtain the upper bound of the optimal performance as well as the influence factor of SPL and render a general probability framework for SPL. In the end, we look into the future of the SPL in the closed and open learning environment. The general probability viewpoint could motivate the design of the easiness standard, the formulation of novel data selection principles, the integration with other learning algorithms regarding the self-paced methodologies which would further benefit the promotion and applications of SPL in practical problems.

KEY WORDS: Self-paced Learning; Curriculum Learning; Machine Learning; Artificial Intelligence

目 录

主要符号表

$R_{SP}(v,\lambda)$	自步正则项
$R_{\mathcal{F}}(f)$	模型关于 f 在空间 $\mathcal{F}$ 中的正则项
$p_{model}(y\|x)$	模型/代理参数化的条件分布的简写
$H(p)$	分布 p 的熵
λ	年龄参数
Q_λ	年龄参数为 λ 时的课程，学习样本的分布函数
V_λ	年龄参数为 λ 时的权重，学习样本的权重函数
D	KL 散度

1 绪论

我们生活在大数据的时代，科学、技术、工程和人们的日常生活在产生着数以PB级和EB级的数据。这些大数据具有极度的多样性，不确定性。对于给定的任务，精良标记的数据往往是少数的并且高成本的，人们能获得轻易大量的相关数据，但是获得数据集的绝大部分往往存在以下问题：

- **缺少和任务直接相关的监督信息**，例如目标检测任务中，只有图片，但由于人力与物力的限制缺乏对图片的标记信息；
- **只拥有和任务部分相关的监督信息**，例如在目标定位问题中，只有图片和图片的标记，却未知目标在图片中的具体定位信息；
- **拥有很多错误标识和不准确的信息**，例如网页上对于视频和图片的文字内容，评论内容，不完整的追踪图片；
- **拥有少量与任务完全无关的数据信息**，例如动物图像识别的数据集中出现了家具的图片；
- **构成结构十分不均衡**，例如类别成分不均匀。

尽管如此，这些不完美却大量的数据中却可以蕴含海量的信息，这也催生了针对不同的数据特性的很多机器学习（Machine learning）的主题，例如半监督学习[①]（Semi-supervised learning）和弱监督学习（Weakly supervised learning）[②]，它们意在利用这些不完美的数据，从中学习出一个对于任务有很好性能的代理（Agent）[③]。其中一些的学习方法需要不断地利用较好的数据来对于大量不完美数据的不确定信息进行推断和学习，然而这直接导致机器学习方法有效性十分依赖其对于数据的学习过程。从狭义上讲，改变对于指定任务中范例的学习顺序会对于学习成效产生影

① 半监督学习泛指从少量有标注数据和大量无标注数据中学习
② 弱监督学习泛指从少量有标注数据和大量存在非直接关联的标注数据中学习
③ 代理泛指以任务为导向的有自主能力的实体。简单的代理可以是一个回归器，分类器等简单的决策体，复杂的代理可以是机器人或者游戏的AI程序，例如AlphaGo，深蓝等。

响。从广义上讲，改变对于不同任务的学习的顺序也会产生不同的学习表现。因此，现有的很多机器学习的研究开始关注于设计一系列被用于学习的数据分布，它们有时可以被视为一系列课程（Curriculum），依照课程次序来对代理进行训练，以使得其能取得一个相对于给定的任务更好的成绩。这些努力可以被大致归结为课程学习（Curriculum learning）[1] 和迁移学习（Transfer learning）[2]。这种人工地对训练数据的学习顺序的排序（或者对数据分布的操控）是十分重要的，但同时又大量消耗人力成本的，并且反映出了现有学习算法的局限性: 他们不能够有效的提取以及重新审视从原始数据中获得的信息。课程学习也因此成为了一种利用人的智能和先验知识去补充这种不足的替代品。为了去拓展机器学习的自适应程度（简单程度），为了使得更加新奇的模型和应用能够被更快捷的建造，更重要地，为了使得代理具备通用的智能，学习算法需要更少的依赖于人工的操作。一个人工智能必须要本质的理解我们周围的世界，并且我们断言他必须具有能对学习还是不学习以及学习什么来做决定的能力。

而最近新出现的一种模拟人类自适应的从简单到难的课程学习原理，自步学习（Self-paced learning），有希望赋予代理这种能力。自步学习的严格定义为：通过从原始的数据中逐步的自适应的确定所学习的数据分布（可以理解为自适应的确定需要学习的课程）来达到一个更好的学习效果的一种方法论。在半监督学习中，代理先对未标注的样本进行伪标注，然后优先选择其中可靠的样本来学习，这种学习模式就利用了自步学习的思想。自步学习的有效性度量现阶段主要依据其在任务上最终的表现，但是也可以用学习的速率和所利用的监督信息量来做衡量标准。

本文调研这个快速发展的领域并将侧重点放在最近的一些进展和我们的工作上。文章主要会分为四个部分：

- **第一部分**涵盖了自步学习的发展史（包括自步学习的前身课程学习的出现，和自步学习的提出），并介绍自步学习在当前在学术领域的现状，和自步学习在应用上取得的成功；
- **第二部分**详细介绍现有的课程学习、自步学习、以及自步课程学习（Self-paced curriculum learning）等方法论的数学描述，以及它们所利用的通用的先验知识（Prior knowledge），并对方法论进行一定的哲学探讨；
- **第三部分**包含自步学习的前沿理论，其中会重点介绍由我们提出的自步学习的统

一优化理论：自步学习的凹共轭性（Concave conjugacy），并深入探讨自步学习程序的统一化：本文提出的自步学习的概率化理解。这将涵盖概率化理解下的通用自步学习模型、难易程度的标准，样本筛选的格式，先验知识的嵌入和影响；

- **第四部分**简要总结了自步学习的核心思想和现阶段主要的作用，并把重点讨论了自步学习发展方向。

同时本文会以问题作为文章逻辑导向的补充，这里将会包含一些读者较为关心的问题，也会涵盖一些推动自步学习领域研究前进的一些很基本的问题，它们包括：

- 我们为什么应该关注自步学习？
- 自步学习的方法论会比较适用于那些学习任务？
- 自步学习的简单到难，从少到多的标准是什么？
- 怎样才能使得学习的过程更好，代理真的应该遵循一种从简单到难，从少到多的学习格式吗？
- 自步学习在优化什么？

2 我们为什么应该关注自步学习?

2.1 自步学习的发展史

相比于随机的学习样例，人类和动物能够通过从简单到困难的学习一系列系统性课程达到一个更加好的学习效果。受此启发，2009 年，Bengio 等人[1]，提出了课程学习的方法论。通过设计一系列课程（学习样本的分布，和对应训练的标准），他们的实验表明，在一些情况下，课程学习可以加快学习的速度，并且提高代理① 的学习表现。然而，启发式手工的课程设计方法，会消耗大量的人力、物力，并且有效性也难以得到保障。这使得如何设计课程成为了课程学习的最大的问题之一。

2010 年，为了能够更好的处理含隐变量的机器学习问题，Kumar[3] 等人，提出一种依据损失来自适应的确定样例的学习顺序的方法论，她们把这种方法命名为自步学习方法论。自步学习的出现给出了一种基于损失自适应的课程设计方法。她们的实验表明，在一些情况下，优先学习损失小的样本，然后逐步学习损失大的样本，有助于达到一个更好的学习效果。相比于那些达到前沿水平的学习方法，自步学习第一次出现就在名词短语的指代消解（自然语言处理），模式发现 (计算生物学)，手写数据识别和目标定位（计算机视觉）等任务中展现出了很好的性能。她们随着弱标注② （Weakly labelled）数据的数量的快速增长，自步学习会这个领域收获巨大的成功。

2.2 自步学习学术领域现状

现在自步学习已经自成机器学习社区当中的一个领域，同机器学习各种主题相互结合，并在其下很多应用中取得了最新的水平，相关的文章出现在很多顶级的会议和期刊上其中包括 NIPS，ICCV，CVPR，IJCAI，ICML，PAMI 等。尽管学习方式遵从于人类课程学习的原则，即从简单到困难，是自步学习相关论文一个很重要的故事部分，但很多先验知识在学习过程中也十分有意思且可以被很简单的获得（这将会在章

① 代理泛指以任务为导向的有自主能力的实体。简单的代理可以是一个回归器，分类器等简单的决策体，复杂的代理可以是机器人或者游戏的 AI 程序，例如 AlphaGo，深蓝等。
② 弱标注指与学习任务存在非直接关联的标注

节 3.2.3.1被讨论）。在学术界上，自步学习快速增长的科研活动伴随着引人注目的一串在实际应用经验上的、以及理论上的进步（将会在章节 4讨论）。下面，我们简要的强调它在各种学习主题下的一些应用上的重要成功。

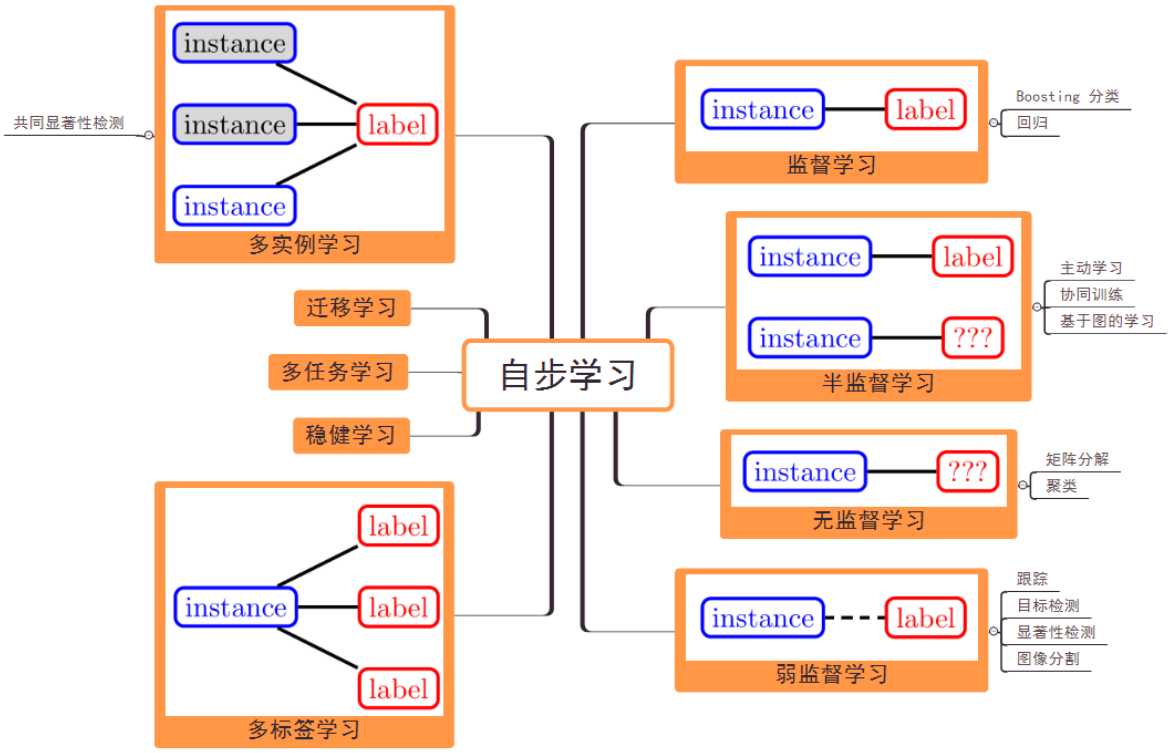

图 2-1　自步学习与机器学习主题结合图

2.3　自步学习在应用中的取得的成果

我们将以学习主题为划分，简要介绍自步学习在各主题中的应用，和一些重要的成果。它们大多数从经验上或者理论上，显示了自步学习有效性，并也预示着自步学习在人工智能领域的潜力。更加详细应用内容介绍可以在附录 5.4.3中找到。

在弱监督学习①领域中, 自步学习有助于对于弱关联标注数据的推断从而使得其在图像分割[4]，共同显著性检测[5]，追踪定位[6,7]，目标检测[8,9] 等应用领域取得很好的表现，其中一些达到了该应用领域的最新水平。

在稳健学习②领域中，自步学习有助于从带有大噪音的样本中提取出可靠地监督信息。最新涌现出来利用自步学习的提升稳健性的研究包括 boosting 分类[10]，噪声数据下利用流形正则的半监督分类[11]，多媒体事件检测[12]，含大规模噪音的网络数据

① 弱监督学习泛指从少量有标注数据和大量存在非直接关联的标注数据中学习
② 稳健学习泛指从带有大量标记噪声，甚至含有一定量外来样本的有数据集上的学习

的概念学习[13] 等。其中对媒体事件检测中利用的自步重排序方法达到了该领域的最优水平。

在半监督学习① 领域中，自步学习有助于对于未标注样本的标注稳健的推断，这使得自步学习和许多著名半监督学习方法论相结合包括，协同训练，自适应的拉普拉斯图割[14]，主动学习[15] 等。特别地，协同学习与自步学习的结合使得所需的监督信息大大减少。在数据集 Pascal VOC2007 的目标检测任务中，Dong 等人②，在仅利用了极少的监督信息，每类 3 到 4 个有标注的图像，但达到了与利用了大部分标注信息的学习方法相近的表现。

在无监督学习③（Unsupervised learning）领域中，自步学习与矩阵分解[16]，多视角聚类[16] 等方法结合，被期许帮助达到一个更好的局部极优解。

在监督学习④（Supervised learning）领域中，研究兴趣集中在利用自步学习方法达到一个极度非凸监督学习问题的一个更好的局部极优解。Avramova 等人[17], 在大型有标注图像数据集中利用从难到易样本学习过程来训练深度卷积神经网络，取得相较于传统训练方法更好的表现。Li 等人[18,19]，将自步学习与多标记和多任务学习相结合使得学习的过程不仅可以依据样本从易到难的学习，同时也依据标记和任务进行从易到难的学习。

此外，自步学习也被用在数据筛选任务中，包括数据类别均衡化[12,20]。在高度类别不均衡的数据集上任务学习中，自步学习引入使得代理可以自适应的调整所学习的数据类别成分，使学习更加全面。

2.4 自步学习比较适用的学习任务

从自步学习的应用成果可以发现，相比于简单的学习任务，自步学习只有在一些较为复杂任务上才能显现出方法论上的优势。相比于普通的精良标注的数据集上的学习任务，不完美数据集上的学习任务往往是较为困难的。因此，半监督学习，弱监督学习，稳健学习，样本筛选等，可能会是目前已知自步学习较为合适的学习任务。

而对于给定的任务和方法论，利用数学的语言对其进行描述是建模过程中一个很

① 半监督学习泛指从少量有标注数据和大量无标注数据中学习
② 该文章尚未公布互联网版本
③ 无监督学习泛指对于没有标注信息的数据进行学习
④ 监督学习泛指对于有标注的数据进行学习

重要的步骤。课程学习和自步学习的理论也由一套严格的数学描述而建立。

3 学习过程的建立：课程学习与自步学习

在人类学习中，人们通过系统性的学习一系列课程，以获取一些技能和认识。相比于直接对于样例的学习和总结，经验上，通过课程学习的过程会更加简单，快捷，稳定和可靠。

对于机器学习，同样的方法论，课程学习可以通过逐渐的改变所学样本的权重来实现。这也被视为对于样本的分布进行了不断地操作后用于代理的训练学习。特别的，从一些列被指定学习顺序的训练样本中学习可以被视为课程学习的一种特例。让那些简单的、重要的，可靠的样本在学习过程中更受关注 (拥有更高的权重)，并且逐步让所有目标内的学习样本拥有更加客观的权值可能会最终助力于更好的学习效果。

3.1 课程学习

抽象的来说，课程是有序的一系列训练标准。每一个学习标准关联着一个被操控的训练样本的分布。最开始“简单”的样本，会被设以高的采样的概率，进行训练。然后接下来的训练标准逐渐发生改变，会增加“难”样本的采样概率。最终采样的分布将会变成目标训练的分布。

3.1.1 课程的定义

课程学习的提出者，Bengio 等人[1] 给出了当时关于课程序列的严格定义。

条件 3.1 *设 z 是样本的随机变量，$P(z)$ 为目标训练的分布. 设 $0 \leq V_\lambda(z) \leq 1$ 表示在 λ 步的课程序列中，被操控的样本 z 权重, 并且 $V_1(z) = 1$。那么在 λ 步被操控的训练分布是，*

$$Q_\lambda(z) \propto V_\lambda(z)P(z) \; \forall z \tag{3-1}$$

满足 $\int Q_\lambda(z)dz = 1$。特别的,

$$Q_1(z) = P(z) \; \forall z。 \tag{3-2}$$

定义 3.1 (课程和课程序列) *设 $\{Q_\lambda, 0 \leq \lambda \leq 1\}$ 为分布序列, 遵循公式3-1 和公式3-2。我们称 Q_λ 为一个* ***课程***，*$\{Q_\lambda, 0 \leq \lambda \leq 1\}$ 为* ***课程序列***，*如果这些分布满足它们的熵增加*

$$H(Q_\lambda) < H(Q_{\lambda+\varepsilon}),\ \forall \varepsilon > 0 \tag{3-3}$$

并且 $V_\lambda(z)$ 关于 λ 单调上升, 即,

$$V_{\lambda+\varepsilon} \geq V_\lambda(z)\ \forall z, \forall \varepsilon > 0。 \tag{3-4}$$

3.1.2　课程学习的实施

课程的定义给出了一种设立课程的方向性指导。例如，弱监督学习中，对于给定图片和标识来预测标识对象的位置信息 (即，包含有标注对象的框) 的学习任务，一个很重要的方法是让代理将更多的精力优先关注在那些简单的图片上 (即，只含有很少交叠对象的图片，以及那些只有单一对象的图片) 然后逐渐的让更多更加复杂的训练样本被学习。于是就可以依照之前提到的方法，人工的安排课程序列 $\{Q_\lambda, 0 \leq \lambda \leq 1\}$，依据最小化期望损失来找到最优的决策函数①，

$$E_{z\sim Q_\lambda} Loss(f, z) = \sum_z Q_\lambda(z) Loss(f, z) \tag{3-5}$$

$$= \sum_z V_\lambda(z) P(z) Loss(f, z) \tag{3-6}$$

$$= E_{z\sim P(z)} V_\lambda(z) Loss(f, z) \tag{3-7}$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} V_\lambda(z_i) Loss(f, z_i)。 \tag{3-8}$$

λ 步时最优的决策函数因此为

$$f_\lambda^* = \arg\min_{f\in\mathcal{F}} \sum_{i=1}^{N} V_\lambda(z_i) Loss(f, z_i) + R_\mathcal{F}(f), \tag{3-9}$$

或者依据贝叶斯方法求得最优的模型的概率分布

$$p_\lambda^* = \arg_{p\in\mathcal{P}} \min D(Q_\lambda || p) + R(p) \tag{3-10}$$

（其中 D 指代概率分布之间的一种相似度量）。从而实现课程学习方法。

① 决策函数是一个给定输入后输出一个决策的函数，常见的决策函数包括分类器，回归器等

然而，当训练集比较大，人工的去标定各个样本的难易程度不仅会十分昂贵，而且使得这种容易程度的标记和这种课程的设计十分启发式。

因此，一些自适应安排课程的方式是迫切被需要的。针对这个问题，自步学习孕育而出。

3.2 自步学习

当人们自主学习的时候，从经验上，人们会贪婪的并且谨慎的倾向于学习那些他们在当前认知水平能够理解的样例。

对于机器学习，样本的难易程度可以被视为是代理在当前状态下对于这个样本进行预测所对应损失的函数[3]。这可以表现为对于不确定信息（见章节 3.2.3.2）和可靠程度信息（见章节 3.2.3.4）的先验的利用。

3.2.1 基于损失的难易程度标准

- **容易程度**

  如果对于样本的损失很小，那么有理由去相信这个样本对于代理来说很简单去学习。

- **可靠性**

  如果代理已经对于任务十分的专业并且样本的损失很高。那么久有理由任认为这个样本是一个外来的和任务无关样本。

用数学的描述，这个想法可以被如下方式编码，即对于某个样本的权重函数依赖于其在当前状态 i 下其预测这个样本的损失，

$$V_{\lambda i}(z) = V_{\lambda}(Loss_i(z))。 \tag{3-11}$$

这引导出了自步学习的格式。

3.2.2 自步学习的数学描述

对于传统的机器学习问题，代理意在通过最小化经验损失函数，学习一个分类或者回归的决策函数，并且为了去避免过拟合的问题，以及引入一些先验知识，一些正则项也会被加入到目标函数之中。

自步学习通过向目标函数引入自步正则项，使得代理能有基于损失能够自适应确定从简单到困难并编码了可靠性信息先验的系列课程。

- **自步学习模型**

$$\inf_{f\in\mathcal{F},v\in[0,1]^n} E(f,v;\lambda)=\inf_{f\in\mathcal{F},v\in[0,1]^n}\sum_{i=1}^{n} v_iL(f,z_i)+R_{SP}(v,\lambda)+R_{\mathcal{F}}(f) \tag{3-12}$$

其中

– **权重函数**

$v=(v_1,v_2,\cdots,v_n)$ 表示各样本权重构成的向量。

– **任务函数**

f 可以是决策函数，例如分类或者回归函数。f 也可以是模型需要学习的概率分布函数，例如分类任务中的条件概率函数，回归任务中的噪音分布函数等。

– **损失函数**

L 代表损失算子（如果函数 f 由 w 参数化，那么 L 就是关于 w 和样本 z 的函数）。l 表示损失的向量 $(L(f,z_1),\cdots,L(f,z_n))^T$. 这样模型可以简化写成

$$\inf_{f\in\mathcal{F},v\in[0,1]^n}\langle v,l\rangle+R_{SP}(v,\lambda)+R_{\mathcal{F}}(f)。 \tag{3-13}$$

– **年龄参数**

λ 是一个控制学习进程的参数，它可以对应到可靠性先验或者不确定性先验的的接受学习的阈值，因此它也控制了加入训练的样本数量和可靠性。它的逐渐增加过程对应到了样本逐渐加入学习的动态过程。

– **自步正则项**

$R_{SP}(v,\lambda)$ 被称作自步正则项，它编码了可靠信息信息先验和从简单到困难，从少到多的学习过程。

– **模型正则项**

$R_{\mathcal{F}}(f)$ 是模型正则项。它嵌入了对于模型的先验知识，同时也起到避免模型过拟合的作用。

- **交替优化方法**

一种比较普遍求解这个目标函数的方法，是去对于任务函数 f 和权重向量 v 进行交替的优化.

优化 f

$$f^k = \inf_{f \in \mathcal{F}} \langle v^{k-1}, l(f) \rangle + R_{\mathcal{F}}(f) \tag{3-14}$$

优化 v

$$v^k = \inf_{v \in [0,1]^n} \langle v, l(f^k) \rangle + R_{SP}(v, \lambda) \tag{3-15}$$

依据这种迭代优化的步骤，代理能够自适应的依据每个样本的损失去生成课程权重。紧接着，依据当前的课程，代理更新它所学习的任务函数。

- **正则项公理**

为了明确自步学习从易到难的学习过程，对于自步正则项的功能进行界定，自步正则项的公理被提出[21]。

定义 3.2 (自步正则项) . *假设 v 是一个权重向量, l 是损失向量属于 $\mathcal{R}_+^n$, 并且 λ 是年龄参数。$R_{SP}(v, \lambda)$ 被称作自步正则项，如果*

1) $R_{SP}(v, \lambda)$ 关于 $v \in [0,1]^n$ 是凸的;

2) $v_i^*(\lambda, l)$ 关于 l_i 单调下降, 并且满足 $\lim_{l_i \to 0} v_i^*(\lambda, l) \leq 1$ 和 $\lim_{l_i \to +\infty} v_i^*(\lambda, l) = 0$

3) $v_i^*(\lambda, l)$ 关于 λ 单调增加, 并且满足 $\forall i \in \{1, 2, \cdots, n\}$, $\lim_{\lambda \to 0} v_i^*(\lambda, l) = 0$ 和 $\lim_{\lambda \to +\infty} v_i^*(\lambda, l) \leq 1$

其中

$$v^*(\lambda, l) = \arg \inf_{v \in [0,1]^n} \{\langle v, l \rangle + R_{SP}(v, \lambda)\}. \tag{3-16}$$

上述的三个条件编码了，下述的信息，

1) $\{0 \leq V_\lambda(z) \leq 1, 0 \leq \lambda \leq +\infty\}$;

2) 权重和损失的对偶关系，损失大的样本的权重应该小（其难度大，或者可靠程度低）；

表 3-1　自步正则项功能关系表

自步正则项	权重函数	非凸损失	名称	数据筛选
$R_{SP}(v,\lambda)$	$v(\lambda,l)$	$F_\lambda(l)$	LOSS	$v(l\geq\lambda)=0$
$-\lambda(v-1)$	$1_{(0,\lambda)}$	$min(l,\lambda)$	Hard	$\surd$
$\frac{\lambda(1-v)^2}{2}$	$(1-\frac{l}{\lambda})_{(0,\lambda)}$	$min(l-\frac{l^2}{2\lambda},\frac{\lambda}{2})$	Soft	$\surd$
$\lambda v(\log v-1)+\lambda$	$e^{-\lambda^{-1}l}$	$\lambda-\lambda e^{-\lambda^{-1}l}$	Exponential	$\times$
$\lambda\log v$	$\min(1,\lambda l^{-1})$	$min(\lambda+\log\frac{l}{\lambda},l)$	Fraction	$\times$

$F_\lambda(l)=(-R_{SP}(v,\lambda))^{*_v}$ 为自步正则项关于 v 凹共轭。

3) 随着年龄参数的增加，样本权重会增大，更多的样本会被加入到训练。

- **与课程学习的关系**

λ 变化对应了从简单到困难，从少到多的过程，而交替迭代的优化过程形成了一系列子课程，

$$\{V_\lambda^k(z)=v^*(\lambda,l(f^k,z)),k\in N\}。\tag{3-17}$$

这种自适应生成的系列课程编码了来自于代理自身的可靠性先验和不确定信息的先验。

Algorithm 1 自步学习的过程

初始化代理任务函数 f, 初始化年龄参数 λ

repeat

　repeat

　　更新课程：依照代理当前状态, 计算其在各样本上的损失，确定各样本学习权重 V_λ（归一化得到 Q_λ）

　　更新代理：依据课程 Q_λ, 训练代理，更新代理任务函数 f

　until 加权损失接近收敛

　增大年龄参数 λ

until 满足训练任务的终止条件

通过对于常用的自步学习方法的总结，本文发现了一些使得学习过程更加好的经验。

3.2.3　怎样让学习过程更好

在机器学习领域，缺少对于周围环境和世界的先验将会使得对于智能的基本结构和训练过程的探索变得十分困难。而两种主要在自步学习过程中被编码的先验是从简单到困难的学习和从小规模到大规模的学习。

1) 自步学习的先验

当处理单元在速度和数量受限，那么从少到多的哲学是不可避免的，并且更重要的是，学习材料的内容在质量上是可以有很大差异的。这种哲学启发我们将少量的样本取为高质量的样本的想法，而这样一个较好的初始点对于非凸优化问题是至关重要的。相关的讨论会在出现在章节 3.2.3.2和 3.2.3.4。

从简单到困难的哲学是一个比较普遍的人类和动物遵循的原则。最基本的想法是先学习简单的样本/任务/概念等，然后其与难样本/任务/概念的关联性也许将有利于代理对于难样本/任务/概念的学习。正因如此，简单是对于代理的主观概念，对它的评价往往会取决于代理已积累的知识。

对于不同的任务，它们的难度可能拥有不同维度。对于本文开头提到的不完美数据，它对应的学习上的困难维度，可以主要归结为，数据中存在不确定性和数据的可靠性。而更好的利用这些难度维度上的标准，和已有的通用先验知识将会让代理的学习过程更好。

2) 不确定性信息

在训练的过程中，经验上，当同时学习有标注样本和大量无标注样本的时候，众多的半监督学习算法容易陷入一个糟糕的局部极优点。无标注样本的引入使得问题，经验上，变得更加的非凸和难以优化。然而，这其中有一些重要的先验被忽视了。事实上，学习了有标注样本的代理可以基于其当前的状态能够提供一些对于无标注样本不确定性的估计，同时一些对于不确定性直接的外部先验知识也能提供帮助。相比于将所有的无标注样本加入训练，将那些被代理自身和先验知识认为是低不确定性的无标注样本优先加入训练是更好的选择。对于分类问题，这是指，如果代理能够利用自身的学习的条件概率函数来对无标注的样本进行各个可能类别的概率估计，那么他可以依据有自身和先验给出的条件熵①，$H(p(y_{predict}|x, prior))$，选择不确定性最小的样本进行学习。如果代理使用的是打分函数，那么最大的打分间隔也是一种类似的标准。然后代理可以对于这些较为确定的样本优先进行学习训练。

① 条件熵是在给定条件下对于随机变量的不确定性的一种度量标准。当随机变量取值为离散的情况下它同时对应于在给定条件下描述这个随机变量所需要的最短字节长度。特别的，离散情况下，条件熵为零，意味着随机变量由该条件完全确定。

3) 确定性信息

有很多确定性的先验信息十分合适于编码到模型中，例如不同类别样本的比例。当面对半监督学习，偏标记学习[22] 时，这种先验信息可以指导伪标注的过程，并且对于模型的性能从经验上会有显著的提升。

4) 可靠程度

相比于不确定性先验，可靠性先验是一种很相似的概念但面向的对象不同。可靠性先验主要针对获得了不准确标签的样本，例如一些针对半监督学习的基于直推式学习算法产生的大量的伪标注样本，大量具有一定标签噪音，或者大量冗余的弱标注样本。这时代理的决策函数或者打分函数可以对这些样本提供一个可靠性的先验($p_{model}(y|x)$)。这些基于代理的可靠性先验可以为当前学习还是不学习该样本的决策提供重要的信息，并且也成为从小到多学习的重要指导: 那些有标注的样本和高可靠性的样本可以成为“小”的那部分集合被优先学习。同时，这个过程可以由动态的改变接受样本的阈值来进一步细化：当样本满足 $p_{model}(y|x) > threshold$，那么它将会被加入训练过程中。相应的模型将会在章节4.2.4.2被讨论。同时，很多外部额外的可靠性先验，例如偏序型先验（将会在章节 3.3.1.1和 4.2.7被讨论），也在经验上[23] 和实验上 (参见 附录 1.6) 被证明十分有效。

5) 关联性

在一些情况下，学习的一些特征会具有一些条件独立性关系，或者相互补充的关系，在一些情况下也可以得到很多不同角度的模型，这时候这些模型本身相互补充，即不同的代理之前的不确定性信息的先验以及可靠性的先验可以相互融合和相互指导。利用这种类型的先验，相比于一个单一的模型，通过训练几个不同的代理可以获得更好的表现。通过利用这种先验，自步协同训练（参见 附录 1.3.0.7）取得了协同训练的最新水平。与之相关的自步学习的集成概率模型也会在章节4.2.9.2被讨论。

3.3 自步课程学习

对于学习内容，除了代理自身的先验知识外，还存在很多外部的先验信息，Jiang等人[24] 将这些外部的先验信息和代理自身的先验知识的利用融合在一起，提出了自步课程学习的学习范式。其中自步学习利用自身的先验知识来不断地生成的系列课程

（需要学习的样本分布）被视作是以学生驱动主导的学习过程，而外部的其他各种先验知识的嵌入对应的系列课程的被视作以老师驱动为主导的学习过程，而两种过程的结合，学生驱动和老师指导共同作用成为了自步课程学习的过程。

3.3.1 课程先验

作为自步学习方法论的自然延伸，自步课程学习能够将更多外部先验信息融合到学习过程中，这也使得其在广泛的应用[5,8,12,13,17,25]中取得了很好的效果。其中一些常见的通用外部先验包括可靠性先验，不确定性先验，关联性先验，总体来源性先验，多样性先验，时空光滑性先验等。

1) 偏序关系先验

偏序关系先验是一种可靠性先验，例如如果我们知道样本 1 比样本 2 更加可靠, 其对应权重 v_1 与 v_2 就应该满足 $v_1 \geq v_2$ 的关系。而这样的权重可行域限制关系 $\Psi = \{(v_1, v_2, \cdots, v_n) | v_1 \geq v_2\}$ 被嵌入到自步学习的课程生成中，则构成偏序关系先验。其对应的自步学习模型为

$$\inf_{f \in \mathcal{F}, v \in [0,1]^n \cap \Psi} \sum_{i=1}^{n} v_i L(f, z_i) + R_{SP}(v, \lambda) + R_{\mathcal{F}}(f)。 \tag{3-18}$$

偏序型先验有着很广泛的应用[13,24], 偏序先验加入的学习过程的影响的理论分析会在章节 4中被讨论。

2) 多样性先验

所学习的样本分布的类别成分信息是一种重要的先验信息，这种多样性先验的嵌入使得学习过程的课程（需要学习的样本分布）各类别样本的比例更均衡。其中若各样本根据类别分组为 $v^{(1)}, v^{(2)}, \cdots, v^{(b)}$ 并排列，两种常用的鼓励样本分布多样性的自步正则项分别是 Jiang 等人[12] 提出的负的 $l_{2,1}$ 范数 : $-\|v\|_{2,1} = -\sum_{j=1}^{b} \|v^{(j)}\|_2$ 和 Zhang 等人[5] 提出的负的 $l_{0.5,1}$ 范数 : $-\|v\|_{0.5,1} = -\sum_{j=1}^{b} \|v^{(j)}\|_{0.5}$。其对应的自步学习模型

$$\inf_{f \in \mathcal{F}, v \in [0,1]^n} \sum_{i=1}^{n} v_i L(f, z_i) + R_{SP}(v, \lambda) + \eta R_{diversity}(v) + R_{\mathcal{F}}(f), \tag{3-19}$$

其中 $R_{diversity}(v)$ 可以为 $-\|v\|_{0.5,1}$ 或者 $-\|v\|_{2,1}$。将这两种正则取负，是常用的诱导组稀疏的正则项，这样就不难理解他们本身对应到鼓励样本的类别的多样性的功能了。

这样的的多样性先验也可以理解为让边缘化的学习样本的类别分布和指定的类别

分布尽可能相同，更具体的多样性先验对于优化过程的影响和深入的概率理论解释将会在章节 4中被进一步讨论。

3)　关联性先验

利用多个代理依据样本不同角度的特征来进行协同训练的时候，多个代理之间给出的可靠性判断能够相互的补充。基于此，Ma 等人①提出了关联性先验，他们将两个代理视角的下对于样本的组权重 $v^{(1)}$ 和 $v^{(2)}$ 利用负内积的形式关联起来提出了关联性的自步正则项 $-{v^{(1)}}^T v^{(2)}$, 其对应的自步学习模型

$$\inf_{f\in\mathcal{F},v\in[0,1]^n} \sum_{j=1}^{2}\sum_{i=1}^{n}(v_i^{(j)}L(f_j,z_i)+R_{\mathcal{F}}(f_j))+R_{SP}(v,\lambda)-\eta{v^{(1)}}^T v^{(2)}, \tag{3-20}$$

可以发现如果两个代理视角下的对应样本的组的权重越接近那么关联性的自步正则项将会取更小的值。

这种关联性先验的嵌入使得不同代理被集成起来，更详细的自步集成学习讨论将会在章节 4进行。

4)　光滑性先验

当样本可靠性信息关于其在空间中的分布具有连续性或者更好的光滑性等拓扑关系，或者说被加入学习可靠样本的概率分布在时空中因具有一定的连续性和光滑性，Meng 等人[21] 提出，那么其可以通过样本之间的相似度拉普拉斯矩阵 L 进行编码得到 $v^T Lv$，即时空连续或光滑性先验的自步正则项。

5)　总体性先验/任务内一致性先验

如果当处理的问题是回归类型的问题时，获得样本的可靠性与其对应来源分布的精度存在对应关系。当样本的来源可以为几个总体所划分时，来自相同总体的样本的权重可以进行共享，这对应到了总体性先验。假设已知样本来源于 k 个总体，$\{i_1,i_2,\cdots,i_{s_i}\}$ 为来自于第 i 个总体的成员们。那么其对应的自步学习模型

$$\inf_{f\in\mathcal{F}}\{\inf_{v_1\in[0,1]}\{s_1R_{SP}(v_1,\lambda)+v_1\sum_{j=1}^{s_1}L(f,z_{1j})\}+\cdots+\inf_{v_k\in[0,1]}\{s_kR_{SP}(v_k,\lambda)+v_k\sum_{j=1}^{s_k}L(f,z_{kj})\}+R_{\mathcal{F}}(f)\}。 \tag{3-21}$$

从另一种角度，如果代理需要解决的学习任务由多个加性的子任务构成，同时我

① 该文章被 ICML2017 收录但互联网版本尚未公开

们如果知道每个子任务中，学习样例的重要程度是相同的，那么其也可以类似对应到上述模型的表达形式，即同一个任务中学习样本的权重是共享的。

总体性先验和任务内一致性先验的利用可以大大减少权重参数的数量，也有助于避免过拟合现象的发生。更详细的概率理解下的总体性先验的讨论将会在章节 4进行。

各种先验的嵌入势必为模型引入了更多的参数，带来更为复杂的优化性质。这为模型的求解，模型的应用，带来了更大的挑战，也需要人们能够对于问题保持着清醒的认识。为了更好的理解自步学习的内涵，更加深入的理论理解是被迫切需要的。

4　自步学习与课程学习的理论前沿

4.1　自步学习在优化什么？

自步学习在应用层面的经验上的成功，引发了人们对于自步学习的理论层次的更深入的研究兴趣和思考。Meng 等人[21]，从优化的角度出发探讨了自步学习在优化什么。他们发现自步学习的交替迭代算法等价于在利用 MM 方法 (优化最小化方法) 优化一个稳健的隐藏目标函数，$F_\lambda(l)=\int_0^l v(\lambda,j)\,dj$。而这个隐藏的目标函数的表达形式与统计和机器学习领域中非凸正则惩罚项十分类似，并且一些常见自步正则项对应的隐式目标函数能和一些常用的非凸正则惩罚项能够实现一一对应。这样的发现有助于本质的理解自步学习方法的稳健性，特别是当一些特定的学习机，例如支持向量机[26]和自步学习方法结合时。

Fan 等人[27] 从另一个角度指出自步学习的过程完全由各个样本的权重函数 $v^*(\lambda,l)$（参见公式 3-16）所确定, 他们通过凸共轭和半二次优化将不同的权重函数和自步学习正则项建立起对应关系，并诱导出了自步学习一种隐式的正则项。

不管是从自步学习的优化目标函数出发，还是从自步学习的权重函数出发，都为了更好的帮助我们理解自步学习模型优化过程中形态。能否从优化的角度找到一种统一且更加简约的理论分析方法，来帮助我们理解自步学习的隐藏目标函数，帮助我们理解学习过程和权重函数的关系呢？

答案是可以的，接下来本文会介绍凹共轭理论，它能够达到上述的需求。不仅如此，它

- 能够更进一步说明自步学习优化的隐藏目标函数的导出不依赖于 MM 优化算法，
- 能够将权重函数，隐藏目标函数、和自步正则项建立直接对应关系，
- 能够直接推广到自步课程学习的优化框架之中，解释课程先验对于优化过程的影响，
- 能够说明自步正则项的等价性，

- 能够为自步正则项的设计提供直接且全面的指导。

我们将自步学习与凹共轭理论的完美结合，归结为自步学习的凹共轭性。

4.1.1 自步学习的凹共轭性

1) 凹共轭

定义 4.1 (凹共轭) *函数 $g(v)$ 的凹共轭定义为*

$$g^*(l) = \inf_{v\in\mathcal{R}^n}\{\langle v,l\rangle - g(v)\}。 \tag{4-1}$$

从公式 3-12可以发现，自步学习的目标函数关于权重变量 v 的优化完全可以对应到在计算负的自步正则项 $-R_{SP}(v,\lambda)$ 凹共轭函数

2) 模型等价性

记 $g_\lambda(v) = -R_{SP}(v,\lambda)$，那么自步学习的目标函数等价于

$$\inf_{f\in\mathcal{F},v\in[0,1]^n} E(f,v;\lambda) \tag{4-2}$$

$$= \inf_{f\in\mathcal{F}} R_{\mathcal{F}}(f) + \inf_{v\in[0,1]^n}\sum_{i=1}^{n} v_i L(f,z_i) + R_{SP}(v,\lambda) \tag{4-3}$$

$$= \inf_{f\in\mathcal{F}} g_\lambda^*(l(f)) + R_{\mathcal{F}}(f) \tag{4-4}$$

$$= \inf_{f\in\mathcal{F}} F_\lambda(l(f)) + R_{\mathcal{F}}(f) \tag{4-5}$$

其中为了符号形式和隐藏目标函数统一，我们记 $F_\lambda(l) = g_\lambda^*(l)$。

从上述推导也容易看出自步学习最终的目标函数可以由 $-R_{SP}(v,\lambda)$ 的凹共轭函数 $F_\lambda(l)$ 确定，而具有相同的凹包的函数具有相同的凹共轭函数，这说明对于自步正则项来说具有相同凸包的自步正则项所对应的优化目标函数是相同的，故只需考虑当中的那个凸函数代表即可，这也解释了自步正则项定义当中对于正则项凸性要求的合理性。

下面的定理说明在一维情况下，$-R_{SP}(v,\lambda)$ 的凹共轭函数数 $F_\lambda(l)$ 正好与 Meng 等人[21] 发现的自步学习在 MM 算法下的隐藏目标函数 $\int_0^l v(\lambda,j)\,dj$ 等价。

定理 4.1 (模型等价性) *如果 v 是一维的, 且 $R_{SP}(v,\lambda)$ 关于 v 严格凸，下半连续，同时*

$dom_v R_{SP}(v,\lambda) \subset [0,1]$ 且 $0,1 \in cl(dom_v R_{SP}(v,\lambda))$ 那么

$$F_\lambda(l) = \int_0^l v(\lambda, j)\, dj + C(\lambda) \tag{4-6}$$

其中 $C(\lambda)$ 是 λ 的函数。

证明参见 附录 2.2.

从定理中也可以发现，这种隐藏目标函数的获得其实一方面由权重函数决定，一方面又完全对应自步学习的正则项的凹共轭函数。而隐藏目标函数的导出可以独立于优化最小化方法，其事实上对应了自步学习本质的优化目标函数。

3) 权重函数，自步正则项，与隐藏目标函数的本质联系

下述定理进一步阐明了，从不同角度出发的自步学习的优化理解的紧密且确定的联系。

定理 4.2 (关联性) *如果 $R_{SP}(v,\lambda)$ 关于 v 凸，下半连续，同时 $dom_v R_{SP}(v,\lambda) \subset [0,1]$ 且 $0,1 \in cl(dom_v R_{SP}(v,\lambda))$，那么，*

$$l_\lambda(v) = \partial_v(-R_{SP}(v,\lambda)) \tag{4-7}$$

$$v(\lambda, l) = l_\lambda^{-1}(l) \tag{4-8}$$

$$v(\lambda, l) = \partial F_\lambda(l) \tag{4-9}$$

$$F_\lambda(l) = \langle v(\lambda,l), l\rangle + R_{SP}(v(\lambda,l),\lambda) \tag{4-10}$$

$$R_{SP}(v,\lambda) = \langle v, l_\lambda(v)\rangle - R_{SP}(v,\lambda)(l_\lambda(v)) \tag{4-11}$$

如果 v 和 l 是一维的, 且 $R_{SP}(v,\lambda)$ 关于 v 严格凸, 我们可以进一步得到

$$F_\lambda(l) = \int_0^l v(\lambda, j)\, dj + C(\lambda) \tag{4-12}$$

$$R_{SP}(v,\lambda) = -\int_0^v l_\lambda(j)\, dj + C(\lambda) \tag{4-13}$$

其中 ∂ 是指函数的次梯度，∂_v 是指函数关于 v 的次梯度。

这个定理是性质 附录 2.1的直接结果，而后面两个不等式是与定理 4-2对称的结果。

上述定理完美的阐述了自步学习的隐藏目标函数，权重函数，自步正则之间的关系。而利用这种关系我们可以十分便捷的将先验信息通过自步正则项或者是权重函数或者是隐藏目标函数嵌入到优化模型之中。这为自步学习模型的设计提供全面的优化层次的指导。对于更进一步的外部信息的编码，凹共轭理论可以直接推广到对于自步课程学习优化角度的解释。

4) 自步课程学习

自步课程学习通常通过对于权重变量的可行域限制，例如偏序型先验，或者通过增加一些额外的正则项来实现，例如多样性先验，关联性先验，光滑性先验等先验中的正则项。

我们将这两种添加先验的格式称为课程区域与课程函数，而它们的加入对自步学习隐藏目标函数的影响可以利用凹共轭变换下的最大卷积运算来进一步研究。

定义 4.2 (最大卷积) *函数 f 和 g 的最大卷积定义为*

$$f \oplus g(v) = \sup_{v_1+v_2=v} \{f(v_1) + g(v_2)\}。 \tag{4-14}$$

课程区域Ψ

$$F^{new}(l) \triangleq \inf_{v\in\Psi}\{\langle v, l\rangle + R_{SP}(v)\} = F \oplus \delta^*(\cdot|\Psi)(l) \tag{4-15}$$

其中 $\delta^*(l|\Psi)$ 是课程区域上的指示函数 $\delta(v|\Psi)$ 的凹共轭函数。

课程函数 $R_{CL}(v)$

$$F^{new}(l) = \inf_{v\in[0,1]^n}\{\langle v, l\rangle + R_{SP}(v) + R_{CL}(v)\} = F \oplus (-R_{CL})^*(l)。 \tag{4-16}$$

利用凹共轭理论中最大卷积的性质，可以说明常见的偏序型先验的加入将会减少与偏序先验描述对应不一致的损失区域对于隐藏目标函数下降量的贡献。这会倾向于使得所求得的解所对应的损失落在偏序先验描述所对应区域中。

同时利用凹共轭理论也可以很好的分析总体性先验加入模型的影响。依据公式 3-21可以进一步得到

$$F^{new}(l) = (s_1 \star \bar{F_\lambda})(\sum_{j=1}^{s_1} l_{1j}) + \cdots + (s_k \star \bar{F_\lambda})(\sum_{j=1}^{s_k} l_{kj}) \tag{4-17}$$

其中

$$s_1 \star \bar{F}_\lambda(x) \triangleq s_1 \bar{F}_\lambda(s_1^{-1}x), \quad \bar{F}_\lambda(l) = \inf_{v\in[0,1]} vl + R_{SP}(v,l)。 \tag{4-18}$$

从中可以看到同属一个总体的样本的损失会经过平均后再被用于计算隐藏目标函数的损失。

而关联性先验和光滑性先验对应的课程函数，也可能可以从凹共轭的角度来进一步分析其对优化过程的影响。

4.1.2 自步学习的收敛性

本文提出的凹共轭理论对于更好的自步学习在优化方面的性质提供了帮助，而自步学习到底会收敛到哪也受到了一定的关注，Ma 等人[28] 证明了当年龄参数固定的时候，利用优化最小化方法得到的解将会收敛到隐藏目标函数关于模型参数的一个驻点上。

如果对于模型优化层次的理解被当做数学工具包的大显身手，那么对于问题本质的认识可能需要从物理和概率的层次出发。

4.2 自步学习的概率理解

本文将首次从统一的概率框架下对于自步学习进行重新的认识和讨论。而本文将会从对于自步学习难易程度标准的讨论，开启自步学习的概率理解。

4.2.1 难易程度基于损失还是基于概率？

基于损失的自步学习的难易程度、可靠性程度等标准会随着任务，损失函数，学习机的表现评价标准的变化而改变，这使得自步学习标准的物理意义受制于任务、学习机和损失函数的局限性，也使得其缺少通用性的解释。例如，在应用中，代理（Agent）① 的损失可能是根据实际任务的奖励函数来确定的，而并不是对于单纯类别或者状态的概率推断，这时损失和样本的可靠性程度的对应关系将会受到很大影响。一个简单的生活理解，我们不能因为疾病发生后的代价大而认为我们不太可能得病，否则将会出现讳疾忌医的现象。

但是如果学习的主体是在学习某种概率的分布函数，学习条件概率分布，例如

① 代理泛指以任务为导向的有自主能力的实体。简单的代理可以是一个回归器，分类器等简单的决策体，复杂的代理可以是机器人或者游戏的 AI 程序，例如 AlphaGo，深蓝等。

$p_{model}(y|x)$，那么用概率的统一表现形式可以很好的将可靠性程度，不确定性等先验利用统一的形式进行编码。

1) 损失和概率分布的对应关系

常用的很多经验和期望损失最小化模型的损失函数可以很好的被联系到最大似然框架或者最大后验框架，从而可以和概率分布形成对应，为了避免歧义，下面的概率分布的下标采用完整形式。首先是期望损失最小化模型与最大化 log 似然函数的对应关系，

$$E_{(x,y)\sim p_{XY}} L(f,(x,y)) \propto E_{(x,y)\sim p_{XY}} \log e^{\alpha L(f,(x,y))-\beta} \tag{4-19}$$

$$\arg\min_{f\in\mathcal{F}} E_{(x,y)\sim p_{XY}} L(f,(x,y)) = \arg\max_{f\in\mathcal{F}} E_{(x,y)\sim p_{XY}} \log e^{-\alpha L(f,(x,y))+\beta} \tag{4-20}$$

其中 $\alpha \in R_+, \beta \in R$。

生成模型下损失和联合概率分布的对应关系： $p_{model\,XY}(x,y) = e^{-\alpha L(f,(x,y))+\beta}$,

$$\arg\min_{f\in\mathcal{F}} E_{(x,y)\sim p_{XY}} L(f,(x,y)) \tag{4-21}$$

$$= \arg\max_{f\in\mathcal{F}} E_{(x,y)\sim p_{XY}} \log p_{model\,XY}(x,y) \tag{4-22}$$

$$= \arg\min_{f\in\mathcal{F}} D(p_{XY}||p_{model\,XY}) \tag{4-23}$$

其中 D 表示 KL 散度。

判别模型下损失和条件概率分布的对应关系： $p_{model\,Y|X}(y|x) = e^{-\alpha L(f,(x,y))+\beta}$

$$\arg\min_{f\in\mathcal{F}} E_{(x,y)\sim p_{XY}} L(f,(x,y)) \tag{4-24}$$

$$= \arg\max_{f\in\mathcal{F}} E_{(x,y)\sim p_{XY}} \log p_{model\,XY}(x,y) \tag{4-25}$$

$$= \arg\min_{f\in\mathcal{F}} E_{x\sim p_X} D(p_{Y|X=x}||p_{model\,Y|X=x})。 \tag{4-26}$$

一些常见的损失都可以很好的对应到某种概率判别模型或者生成模型，参见表 4-1。

常见的支持向量机也可以转化成概率输出模型 [29]。

表 4-1　常见损失和概率分布对应表

损失	分布	名称	应用
$(y-f(x))^2$	$\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(y-f(x))^2}{2\sigma^2}}$	Gauss	连续回归
$\lvert y-f(x)\rvert$	$\frac{1}{2b}e^{\frac{-\lvert y-f(x)\rvert}{b}}$	Laplace	连续回归
$\lVert y-f(x)\rVert_{L^p}$	$\frac{1}{\Gamma(1+\frac{1}{p})b^{\frac{1}{p}}}e^{\frac{-\lVert y-f(x)\rVert_{Lp}^p}{b}}$	L^p	连续回归
$\log(1+e^{-\frac{yf(x)}{b}})$	$\frac{e^{\frac{yf(x)}{2b}}}{\sum_{i\in\{-1,1\}}e^{\frac{if(x)}{2b}}}$	Logistic	二分类
$\log(\frac{e^{\langle\beta_y,f(x)\rangle}}{\sum_{i\in\mathcal{Y}}e^{\langle\beta_i,f(x)\rangle}})$	$\frac{e^{\langle\beta_y,f(x)\rangle}}{\sum_{i\in\mathcal{Y}}e^{\langle\beta_i,f(x)\rangle}}$	Multinomial Logistic	多分类

2)　容易程度和可靠性的概率化定义

基于损失转化成基于概率的自步学习能够提供更加直接容易程度和可靠性的物理意义。一般情况我们认为一个规律简单，可以理解为我们头脑中存在一种对于这个规律很简短的描述。而我们认为一个事件可靠则是由于预测其发生的概率比较高。于是我们可以重新审视自步学习当中的简单和可靠地定义。

- **容易程度**

  代理对于随机变量$Y|X=x$的描述长度，$H(p_{model Y|X=x})$。这等价于章节3.2.3.2中提到的不确定信息的先验的编码。

  如果代理在当前状态下对于$Y|X=x$的描述长度较长，等价于代理认为随机变量的不确定性更显著，不易于做出判断，而描述长度很短的时候，等价于代理认为很容易做出判断。

- **可靠性**

  代理对于事件$(Y=y|X=x)$或$(X=x,Y=y)$的发生概率的预测值$p_{model XY}(x,y)$,或者$p_{model Y|X}(y|x)$。

  如果代理已经对于对象的联合分布或者条件分布有着很好的学习，但是代理预测该事件发生的概率很小。那么就有理由认为这是一个外来的样本和或者和任务无关的样本。

需要注意的是这两种先验都是基于代理的主观推断，当主观推断和实际事实出现较大偏差的时候，单纯的自步学习方法将会朝着错误的方向优化和学习。

4.2.2 自步学习的最优表现的上界

而从信息论角度理解自步学习方法论能达到的最好效果将会被不确定性因素约束，并可以有如下形式化的展现。

标记噪声设定

假设我们获得的数据的标注是有噪声的，我们需要依据代理初始的经验 $Agent_{initial}$，一些外部的先验 $Prior$ 和现有的数据 X 和其有噪声的标注 Y_{noise}, 构造一个决策函数 $\bar{Y} = decision(X, Y_{noise}, Agent_{initial}, Prior)$ 去预测 Y_{true}, 那么由信息论中的法诺不等式[30], 可以导出判别误差 $P_{error} = Pr(Y_{true} \neq \bar{Y})$ 与条件熵 $H(Y_{true}|X, Y_{noise}, Agent_{initial}, Prior)$ 的如下关系，

$$H(P_{error}) + P_{error} \log |\mathcal{Y}| \geq H(Y_{true}|\bar{Y}) \geq H(Y_{true}|X, Y_{noise}, Agent_{initial}, Prior) \quad (4\text{-}27)$$

和它的弱化形式的关系，

$$P_{error} \geq \frac{H(Y_{true}|X, Y_{noise}, Agent_{initial}, Prior) - 1}{\log |\mathcal{Y}|}。 \quad (4\text{-}28)$$

半监督设定

假设我们获得的数据大部分缺乏标注，我们需要依据代理初始的经验 $Agent_{initial}$，一些外部的先验 $Prior$ 和现有的无数据 X 和少数有标注数据 $(X_{labelled}, Y_{labelled})$, 构造一个决策函数 $\bar{Y} = decision(X, X_{labelled}, Y_{labelled}, Agent_{initial}, Prior)$ 去预测 $Y_{unlabelled}$, 那么由信息论中的法诺不等式[30], 可以导出判别误差 $P_{error} = Pr(Y_{unlabelled} \neq \bar{Y})$ 与条件熵 $H(Y_{unlabelled}|X, X_{labelled}, Y_{labelled}, Agent_{initial}, Prior)$ 的如下关系，

$$H(P_{error}) + P_{error} \log |\mathcal{Y}| \geq H(Y_{true}|\bar{Y}) \geq H(Y_{unlabelled}|X, X_{labelled}, Y_{labelled}, Agent_{initial}, Prior) \quad (4\text{-}29)$$

和它的弱化形式的关系，

$$P_{error} \geq \frac{H(Y_{unlabelled}|X, X_{labelled}, Y_{labelled}, Agent_{initial}, Prior) - 1}{\log |\mathcal{Y}|}。 \quad (4\text{-}30)$$

从上述问题设定和法诺不等式导出的预测误差的下界（模型性能的上界）中可以看出条件熵越大，那么判别精度的上界越小，自步学习方法能达到的最好效果因此也

受限。考虑到条件熵的形式，可以发现**代理初始的经验，拥有的先验知识，和数据本身的性质**，例如有标注的信息量，标记噪音的程度等都会影响最终性能的上界。

基于概率和信息论理解的下的自步学习格式有助于我们重新审视课程（被操控的样本分布）的生成方式。

4.2.3　通用自步学习概率模型

假设我们学习的课程分布为 Q_λ, 它的字母表为所有样本点 $\{z_i|i=1,\cdots,n\}$, 它在每个样本点上的概率取值为 $Q_\lambda(z_i)=v_i$，我们的代理学习的模型分布为 P_{model}, 我们将课程的设定和课程的先验知识描述为对于 Q_λ 的可行域的限制（课程区域）$\mathcal{Q}_\lambda$ 和正则项（课程函数）$R_{\mathcal{Q}_\lambda}(Q_\lambda)$，对于模型分布的先验描述成可行域上的限制 $\mathcal{P}_{model}$ 和正则项 $R_{\mathcal{P}}(P_{model})$, 于是我们得到通用的自步学习的概率模型，

通用自步学习概率模型

$$\min_{Q_\lambda\in\mathcal{Q}_\lambda, P_{model}\in\mathcal{P}_{model}} \quad D(Q_\lambda||P_{model})+R_{\mathcal{Q}_\lambda}(Q_\lambda)+R_{\mathcal{P}}(P_{model})。 \tag{4-31}$$

在给定一个代理的初始经验，即模型初始分布的情况，通过交替对于模型的概率分布和学习的课程分布进行优化，并逐渐的改变课程设定 λ, 那么就导出了通用的自步学习进程。

Algorithm 2 概率化自步学习的过程

初始化模型分布 P_{model}, 初始化年龄参数 λ

repeat

　repeat

　　更新课程：固定代理当前状态 P_{model}，求得自适应确定的学习的课程 Q_λ

　　更新代理：固定选定训练课程 Q_λ，训练代理，更新模型分布 P_{model}

　until 目标函数接近收敛

　增大年龄参数 λ

until 满足训练任务的终止条件

自步学习的概率模型需要结合具体问题和具体的先验知识来进一步确定课程的设定方式，课程函数的形式，模型正则项的形式，等等。为此我们更加细致的讨论标记噪声与外来样本情况下的自步学习概率模型，和半监督学习任务下的自步学习概率模型。

4.2.4 标记噪声与外来样本

假设我们学习的课程分布为 Q_λ, 它的字母表为所有样本点 $\{(x_i, y_i)|i = 1, \cdots, n\}$, 它在每个样本点上的概率取值为 $Q_\lambda(x_i, y_i) = v_i$，我们的学习代理分布 $p_{model Y|X;W}$ 由 W 参数化，那么我们可以将对于标记噪声和外来样本的学习问题表示成

标记噪声与外来样本的自步学习概率模型

$$\min_{Q_\lambda \in \mathcal{Q}_\lambda, W \in \mathcal{W}} \quad E_{x\sim Q_{\lambda X}} D(Q_{\lambda Y|X=x}||p_{model Y|X=x;W}) + R(Q_\lambda) + R_{\mathcal{W}}(W)。 \tag{4-32}$$

我们讨论一些特殊的课程设定和特殊的正则下的情形。

1) 负熵正则：$R_{\mathcal{Q}_\lambda}(Q_\lambda) = -H(Q_\lambda)$

负的熵正则的引入会使得对于原始数据点分布的估计会更加接近均匀分布，即鼓励更加充分的利用所有的数据。也使得模型的可以依据 KL 散度的性质快速求解。

课程设定：指定学习样本的数量

$$\begin{aligned} \min_{Q_\lambda, W \in \mathcal{W}} \quad & E_{x\sim Q_{\lambda X}} D(Q_{\lambda Y|X=x}||p_{model Y|X=x;W}) - H(Q_\lambda) + R_{\mathcal{W}}(W) \\ s.t. \quad & v_i \geq 0, \quad \forall i = 1, \cdots, n \\ & \textstyle\sum_{i=1}^{n} v_i = 1 \\ & \|V\|_0 = \lambda \quad 。 \end{aligned}$$

其中 λ 代表指定样本数量取值范围为 1 到 n，变化从少到多。

对应求解过程（交替下降 Q_λ,W）：当对 Q_λ 求解时, 相当在寻找最大的 λ 个的 $p_{model Y|X;W}(y_i|x_i)$ 加和，只需对于 $p_{model Y|X;W}(y_i|x_i)$（样本依据其可靠程度排序）进行从大到小排序然后选择前 λ 个指标对应的 $v_{i_k} = \frac{p_{model Y|X;W}(y_{i_k}|x_{i_k})}{\sum_{j=1}^{\lambda} p_{model Y|X;W}(y_{i_j}|x_{i_j})}$, 其余设定成 0 即得到最优解。

课程设定：指定学习样本的数量和等可靠性

$$\begin{aligned} \min_{Q_\lambda, W \in \mathcal{W}} \quad & E_{x\sim Q_{\lambda X}} D(Q_{\lambda Y|X=x}||p_{model Y|X=x;W}) - H(Q_\lambda) + R_{\mathcal{W}}(W) \\ s.t. \quad & v_i \geq 0, \quad \forall i = 1, \cdots, n \\ & \textstyle\sum_{i=1}^{n} v_i = 1 \\ & v_i = 0 \ or \ v_i = \frac{1}{\lambda} \quad \forall i = 1, \cdots, n。 \end{aligned}$$

其中 λ 代表指定样本数量取值范围为 1 到 n，变化从少到多。

对应求解过程（交替下降 Q_λ,W）：原问题相当于指定了选择的样本数量，同时样本数量之间权重相等。当对 Q_λ 求解时，只需对于 $p_{modelY|X;W}(y_i|x_i)$（样本依据其可靠程度排序）进行从大到小排序然后选择前 λ 个指标对应的 $v=\frac{1}{\lambda}$, 其余设定成 0 即得到最优解。

2) 负 L^1 正则：$R_{\mathcal{Q}_\lambda}(Q_\lambda)=\log\lambda\|Q_\lambda\|_1$

当我们更关心，所学习的样本的可靠性达到标准，而不太关心样本分布的归一化时，负 L^1 正则和他的一些变种是很好的选择。即假设 $Q_\lambda(x_i,y_i)=C_\lambda v_i, C_\lambda$ 为归一化系数。

课程设定：指定学习样本的可靠性阈值

$$\min_{v\in[0,1]^n,W\in\mathcal{W}} \quad \sum_{i=1}^{n} -v_i\log p_{modelY|X;W}(y_i|x_i)+R_{\mathcal{W}}(W)+(\log\lambda)\|V\|_1。 \tag{4-33}$$

其中 λ 代表指定接受可靠性阈值取值范围为 0 到 1，变化从大到小。

对应求解过程（交替下降 v,W）：求解得到，$v_i(p_{modelY|X;W}(y_i|x_i))=1_{(\lambda,1)}$，即如果代理认为条件概率大于可靠性阈值 λ，那么对应样本将会被加入到课程中。

3) 代理初值选取

一般设置代理的初值为由保障泛化能力的学习机在全部样本上进行训练得到模型参数。然后通过逐渐增加使用样本的数量或者降低样本接受的可靠性程度 λ 标准, 并通过交替迭代的方法将可以得到一系列课程 Q_λ 和对应的子课程，以此对于代理进行反复训练。这样的学习过程，有助于代理达到一个更好的局部极优值。

4.2.5 半监督学习

假设所有有标注的样本的分布是 $p_{labelXY}$, 而未标注的样本上需要自适应的生成相应的课程分布 Q_λ, 它的字母表为未标注样本点与其所有对应的可能类别 $\{(x_i,y_j)|i=1,\cdots,n_{unlabelled},j=1,\cdots,|\mathcal{Y}|\}$，我们的学习代理分布 $p_{modelY|X;W}$ 由 W 参数化，注意到有标注样本的信息可以通过

$$R_{\mathcal{P}}(P_{model})=E_{(x,y)\sim p_{labelXY}}\log\frac{1}{p_{modelY|X;W}(y|x)} \tag{4-34}$$

来很好刻画，那么我们可以将对于半监督的学习问题类似于标记噪音的模型建立方法，表示成，

1) 负熵正则：$R_{\mathcal{Q}_\lambda}(Q_\lambda) = -H(Q_\lambda)$

课程设定：指定学习样本的数量

$$\begin{aligned}
&\min_{Q_\lambda, W\in\mathcal{W}} \quad E_{(x,y)\sim p_{label\,XY}} \log \frac{1}{p_{model\,Y|X;W}(y|x)} + R_{\mathcal{W}}(W) \\
&+ \eta \frac{n_{unlabelled}}{n_{labelled}} (E_{x\sim Q_{\lambda X}} D(Q_{\lambda Y|X=x} || p_{model\,Y|X=x;W}) - H(Q_\lambda)) \\
&s.t. \qquad\qquad v_{ij} \geq 0, \quad \forall i = 1, \cdots, n, \forall j = 1, \cdots, |\mathcal{Y}| \\
&\qquad \textstyle\sum_{i,j=1}^{n} v_{ij} = 1 \\
&\qquad\quad \|V\|_0 = \lambda \quad 。
\end{aligned}$$

课程设定：指定学习样本数量和等可靠性

$$\begin{aligned}
&\min_{Q_\lambda, W\in\mathcal{W}} \quad E_{(x,y)\sim p_{label\,XY}} \log \frac{1}{p_{model\,Y|X;W}(y|x)} + R_{\mathcal{W}}(W) \\
&+ \eta \frac{n_{unlabelled}}{n_{labelled}} (E_{x\sim Q_{\lambda X}} D(Q_{\lambda Y|X=x} || p_{model\,Y|X=x;W}) - H(Q_\lambda)) \\
&s.t. \qquad\qquad v_{ij} \geq 0, \quad \forall i = 1, \cdots, n, \forall j = 1, \cdots, |\mathcal{Y}| \\
&\qquad \textstyle\sum_{i,j=1}^{n} v_{ij} = 1 \\
&\qquad v_{ij} = 0 \; or \; v_{ij} = \frac{1}{\lambda} \quad \forall i = 1, \cdots, n, \forall j = 1, \cdots, |\mathcal{Y}| 。
\end{aligned}$$

2) 负 L^1 正则：$R_{\mathcal{Q}_\lambda}(Q_\lambda) = \log\lambda \|Q_\lambda\|_1$

课程设定：指定学习样本的可靠性阈值

$$\begin{aligned}
&\min_{v\in[0,1]^{n_{unlabelled}\times|\mathcal{Y}|}, W\in\mathcal{W}} \quad E_{(x,y)\sim p_{label\,XY}} \log \frac{1}{p_{model\,Y|X;W}(y|x)} + R_{\mathcal{W}}(W) \\
&+ \eta \Big(\sum_{i=1}^{n_{unlabelled}} \sum_{j=1}^{|\mathcal{Y}|} -v_{ij} \log p_{model\,Y|X;W}(y_j|x_i) + (\log\lambda)\|V\|_1 \Big) 。
\end{aligned}$$

由于半监督学习自身的无标注样本的不确定特点，使得其同样可以依据概率化定义的容易程度（条件熵）的角度来进行从易到难的自步学习过程。

课程设定：指定学习样本的难易程度阈值

$$\min_{v\in[0,1]^{n_{unlabelled}},W\in\mathcal{W}} \quad E_{(x,y)\sim p_{labelXY}}\log\frac{1}{p_{modelY|X;W}(y|x)}+R_{\mathcal{W}}(W)$$
$$+\eta(\sum_{i=1}^{n_{unlabelled}} v_i H(p_{modelY|X=x_i;W})-\lambda\|V\|_1)。$$

其中 λ 表示接受的不确定性（容易程度）阈值，范围 $[0,\log|\mathcal{Y}|]$，变化从小到大。对应求解过程（交替下降 v,W）：求解得到，

$$v_i(H(p_{modelY|X=x_i;W}))=1_{(0,\lambda)}(H(p_{modelY|X=x_i;W})), \tag{4-35}$$

即如果代理认为难易程度小于 λ 的阈值，那么对应样本将会被加入到课程中。

3) 代理初值选取

代理的初值一般由保障泛化能力的学习机在有标注样本上进行训练得到。通过逐渐改变参数 λ 和参数 η 使得在接受样本的可靠程度达到一定标准的情况，逐步增加被利用的未标注样本数量，通过交替迭代方法得到一系列的课程和子课程。这样的学习过程，有助于学习机达到一个更好的局部极优值。

除了上述方法以外，对于半监督学习的格式也可以交替的对于未标注的样本利用当前的代理进行伪标注然后利用标记噪声与外来样本的处理方法来进行处理，以此生成系列学习课程。

4.2.6 弱监督学习

如果弱监督学习的任务中主要呈现的是标记噪声，那么起可以归结为章节 4.2.4讨论的内容。如果是缺少标记内容可以归结为半监督章节 4.2.5的情形。如果是对象层次的不完美，或者受制代理的局限性无法处理，例如跟踪任务中的有漂移现象的训练图片，例如有遮挡的图片等等类型的任务，其对应对象又有着很大冗余性时，利用指定训练样本数量或者指定可靠性阈值的自步学习的方法可以很好挑出冗余样本中可靠性高的样本用于训练，以此可得到更好训练效果。

4.2.7 噪音建模回归模型

如果说离散情况的分类问题自步学习的权重函数被理解为课程（被操控的数据分布），那么回归的情况权重的物理意义则有着极大的转变。下述这种理解和之前的课

程分布的讨论有着很大差别，但是对于自步学习先验作用的理解可能有着一些帮助。

对于有监督的回归问题，人们往往会依据任务要求和数据情况选择回归函数的损失例如 L^1 或者 L^2 损失，但这往往就指定了各个样本的噪音需要服从一个同方差的高斯分布或者拉普拉斯分布。实际情况中，数据的噪音可能更加的复杂，特别是数据来自于不同来源，不同渠道的时候，这时人们尝试使用混合高斯分布等分布假设对于噪音进行聚类[31,32]。这种情况的极端化，如果我们认为每一个数据都有着自己的噪声，这将会对应到连续情况的自步学习的格式。

1) 模型假设

假设 Y 是需要预测的属性值，X 是随机变量，模型假设

回归函数关系

$$Y = f(X;W) + \varepsilon \tag{4-36}$$

噪音分布

$$\varepsilon|v \sim \sqrt{\frac{v}{\pi}}e^{-v\varepsilon^2} \tag{4-37}$$

随机变量的独立性关系假设，见下图，

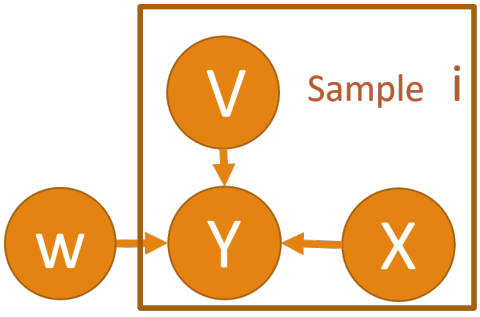

图 4-1 自步学习独立性关系图模型

我们这里假设 v 代表着样本的精度，而所有的样本的精度服从一个 $[0,1]$ 之间的均匀分布的超先验 p_V。（即样本的精度不会太高）

利用图模型的独立性假设，于是联合分布存在如下的因子分解

$$p_{VWXY} = p_{Y|X,V,W}p_V p_X p_W。 \tag{4-38}$$

通过实施对于 W,V 的最大后验估计可以得到，

$$p_{VW|XY}(v,w|x,y) \propto p_{VWXY}(v,w,x,y) \tag{4-39}$$

$$\propto p_{Y|X,W,V}(y|x,w,v)p_V(v)p_W(w)。 \tag{4-40}$$

那么，

$$\arg\sup_{v,w} p_{VW|XY}(v,w|x,y) \tag{4-41}$$

$$= \arg\inf_{v,w} -\ln p_{VWXY}(v,w,x,y) \tag{4-42}$$

$$= \arg\inf_{v,w} -\ln p_{Y|X,W,V}(y|x,w,v) - \ln p_V(v) - \ln p_W(w) \tag{4-43}$$

$$= \arg\inf_{v,w} v(y-f(X;W))^2 - \frac{1}{2}\ln\frac{v}{\pi} - \ln p_V(v) - \ln p_W(w)。 \tag{4-44}$$

同时假设随机变量的超先验由年龄参数 λ 参数化，那么自步正则项成为 $R_{SP}(v,\lambda) = -\frac{1}{2}\ln\frac{v}{\pi}$ 且 $v \in [0,1]$，而 $-\ln p_W(w)$ 编码了模型参数的正则项，于是得到了 1 个样本的自步学习噪音建模的格式。

$$\arg\inf_{v\in[0,1],w} v(y-f(x;w))^2 + R_{SP}(v,\lambda) - \ln p_W(w)。 \tag{4-45}$$

对于 n 个样本的情况，设 $X_D = (X_1,\cdots,X_n), Y_D = (Y_1,\cdots,Y_n)$ 和 $V_D = (V_1,\cdots,V_n)$。

类似的可以得到，

$$\arg\sup_{v_d,w} p_{V_DW_D|X_DY_D}(v_d,w|x_d,y_d) \tag{4-46}$$

$$= \arg\inf_{v_d,w} \sum_{i=1}^{n}(v_i(y_i-f(x_i;w)) - \frac{1}{2}\ln\frac{v_i}{\pi}) - \ln p_{V_D}(v_d) - \ln p_W(w)。 \tag{4-47}$$

因此最终可以写成

$$\arg\inf_{v,w}\sum_{i=1}^{n} v_i(y_i-f(x_i,w))^2 + R_{SP}(v,\lambda) - \ln p_W(w)。 \tag{4-48}$$

于是通过给权重赋予精度的物理意义，对于有监督的回归问题，自步学习模型可以理解为每个样本拥有自己的噪声的最大后验模型。基于这样的图模型假设，自步学习的偏序型先验（章节 3.3.1.1提到）可以很好的理解为 $p_V(v)$ 中支撑集的区域的约束。而总体性先验则可以直接对应到各个总体内部的精度参数的共享。

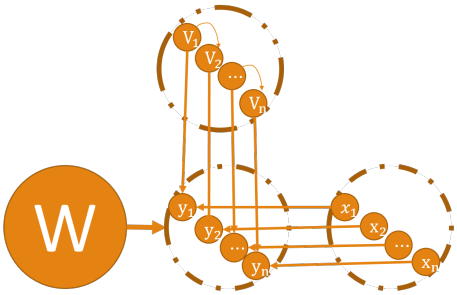

图 4-2　偏序先验的图模型

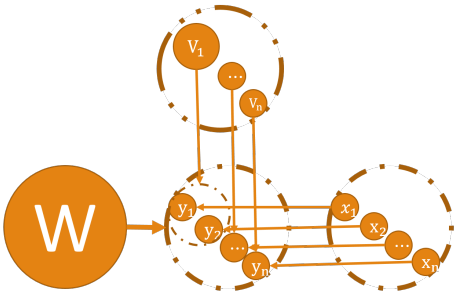

图 4-3　总体性先验的图模型

一些探寻偏序先验和总体性先验加入对目标函数影响的仿真实验结果将会在附录 附录 1.6中被呈现。经验上的结果发现，正确的先验加入可以使得目标函数更加平滑，同时局部极优点的数目会大大减少。

4.2.8　样本选择

自步学习自适应的生成课程（被操控的数据样本）的过程，本质是在依据代理当前状态做一个依据可靠性标准和先验知识的样本选择任务。通过更多先验嵌入可以使得样本的选择更加符合任务的要求，例如引入类别比例先验，例如引入更多的代理来做样本的选择。

4.2.9　课程先验的概率理解

1)　多样性先验与类别比例

通常代理的自步学习过程如果单纯依靠可靠性的标准来自适应课程可能会使得课程分布的样本类别比例与实际需要学习的样本比例产生偏差，使得学习过程偏向于学习更加可靠的类别，然而这往往使得代理在一些类别的判断上呈现出很糟糕的表现。更加均衡的学习，依据任务标准的学习是被需要的。这时候，嵌入多样性先验来指导自适应课程选择，使其符合任务的标准将会是十分重要的。从概率的角度上则任务要求的概率分布可以直接作为指导的标准。

依据指定学习样本数量模型（见章节 4.2.4.1）或者指定学习样本数量和等可靠性模型（见章节 4.2.4.1）的方法可以直接通过选取各个类别的指定数量 (比例) 的样本来实现多样性先验的嵌入。

对于指定学习样本接受可靠性阈值模型（见章节 4.2.4.2），我们可以计算出 $C_\lambda^{-1}=\sum_{i=1}^{N} v_i, Q_{\lambda Y}(y)=C_\lambda \sum_{i=1}^{s_y} v_{yi}$，其中 s_y 那些类别为 y 的样本个数。

因此当所要选择的样本的类别成分 p_{trueY} 是被指定的时候，一个自然的正则项可以自动被得到 $D(p_{trueY}||Q_{\lambda Y})$。其具有如下的变换形式

$$D(p_{trueY}||Q_{\lambda Y}) = -H(p_{trueY}) - \sum_{y\in\mathcal{Y}} p_{trueY}(y)\log\frac{1}{M}\sum_{i=1}^{s_y} v_{ci} \propto -\sum_{y\in\mathcal{Y}} p_{trueY}(y)\log C_\lambda \sum_{i=1}^{s_y} v_{ci}。 \tag{4-49}$$

然而 $\log C_\lambda$ 使得优化变得难以实行。为了缓解这个问题, 我们最小化这个问题的一个上界函数,$-\sum_{y\in\mathcal{Y}} p_{trueY}(y)\log\frac{1}{N}\sum_{i=1}^{s_y} v_{ci}$。

$$\min_{v\in[0,1]^n, W\in\mathcal{W}} \quad (\sum_{i=1}^{n} -v_i \log p_{modelY|X;W}(y_i|x_i) + (\log\lambda)\|V\|_1) + \beta D(p_{trueY}||Q_{\lambda Y})。 \tag{4-50}$$

那么上界目标函数将变成,

$$\inf_{v_i\in\{0,1\}} \sum_{i=1}^{n} v_i(\log\lambda - \log p_{classifier}(y_i|x_i)) - \beta\sum_{y\in\mathcal{Y}} p_{trueY}(y)\log\sum_{i=1}^{s_y} v_{yi}。 \tag{4-51}$$

这使得原问题可以转化为 $|\mathcal{Y}|$ 个子问题,

$$\inf_{v_{yi}\in\{0,1\}} \sum_{i=1}^{s_y} v_{yi}(\log\lambda - \log p_{classifier}(y|x_{yi})) - \beta p_{trueY}(y)\log\sum_{i=1}^{s_y} v_{yi}。 \tag{4-52}$$

让 $l_{y1},\cdots,l_{ys_y}$ 代表 $(\log\lambda - \log p_{classifier}(y|x_{yi}))\ \{i=1,\cdots,s_y\}$ 以升序排序。这里有一个最小的 m 使得

$$\beta p_{trueY}(y)\log(1+\frac{1}{m}) < l_{y(m+1)}, \tag{4-53}$$

那么最优的解 v 满足，$v_{y_i}=1$ 对于 $i\le m$ 并且 $v_{yi}=0$ 对于 $j>m$。

从正则项导出来的解可以发现，如果 $p_{trueY}(y)>0$，那么 y 这个类别当中至少有一个样本的会被选择学习。并且这种解的格式和常用的增加多样性正则项 $-\|v\|_{0.5,1}$ 和 $-\|v\|_{2,1}$ 解的格式类似，也赋予了常用多样性正则的概率理解。同时给出了一种新的多样性的正则项 $-\sum_{i=1}^{b} p_i\log\|v^{(i)}\|_{L^1}$。

2) 关联性先验和集成学习

初学者相互讨论相互学习有利于取得更好的学习效果。当多个代理来协同合作来自适应的生成课程（学习的样本分布），就是对于关联性先验的利用，或者说一种自

步学习的集成方法。这从应用经验上取得了令人震惊的好效果[33]。自步集成模型，从贝叶斯的角度，可以相当于假设模型是代理预测的条件分布的混合，

$$p_{mix}(y|x) = \sum_{i=1}^{K} \pi_i p_{agent_i}(y|x)。 \tag{4-54}$$

那么接受标准则将由多个代理的可靠性的综合，$\sum_{i=1}^{K} \pi_i p_{agent_i}(y|x)$，来确定。

求解各个代理的成分

自步协同训练模型当中各代理所对应的关联系数是需要手工依据验证数据集合来的调整确定，而通过利用在当前确定课程上使用 EM 算法来估计 π_i 是一个更简便（自适应）的方法。设随机变量 $z_{ik} \in \{0,1\}$ 描述样本生成分布的情况, $z_{ik}=1$ 表示第 i 样本来自第 k 个成分，且 $\sum_{k=1}^{K} z_{ik} = 1$。那么下面给出对应的 EM 算法格式，

E-step:

$$E(z_{ik}) = \gamma_{ik} = \frac{\pi_k p_{agent_k}(y_i|x_i)}{\sum_{j=1}^{K} \pi_j p_{agent_j}(y_i|x_i)}。 \tag{4-55}$$

M-step: 考虑课程给定情况下的半监督模型（见模型4.2.5.2）

$$\begin{aligned} & E_Z \sum_{i=1}^{n_{labelled}} \log p_{mixY,Z|X;W,\Pi}(y_i, z_i|x_i) + \eta \sum_{i=1}^{n_{unlabelled}} v_i \log p_{mixY,Z|X;W,\Pi}(y_i, z_i|x_i) \\ = & \sum_{i=1}^{n_{labelled}} \sum_{k=1}^{K} \gamma_{ik} \log p_{agent_k}(y_i|x_i) + \eta \sum_{i=n_{labelled+1}}^{n_{unlabelled}+n_{labelled}} \sum_{k=1}^{K} \gamma_{ik} v_i \log p_{agent_k}(y_i|x_i)。 \end{aligned} \tag{4-56}$$

更新 Π

$$N_k = \sum_{i}^{n_{labelled}} \gamma_{ik} + \eta \sum_{i=n_{labelled+1}}^{n_{unlabelled}+n_{labelled}} \gamma_{ik} v_i, \tag{4-57}$$

$$\pi_k = \frac{N_k}{\sum_{k=1}^{K} N_k}。 \tag{4-58}$$

更新代理的条件分布: 关于各个代理的条件分布最大化

$$\sum_{i=1}^{n_{labelled}} \sum_{k=1}^{K} \gamma_{ik} \log p_{agent_k}(y_i|x_i) + \eta \sum_{i=n_{labelled+1}}^{n_{unlabelled}+n_{labelled}} \sum_{k=1}^{K} \gamma_{ik} v_i \log p_{agent_k}(y_i|x_i)。 \tag{4-59}$$

如果说关联型先验是利用代理的数量和不同角度来提升代理的学习效果，那么直接请一个经验丰富的代理可能也是一个很不错的方法。这引出了迁移学习的先验。

3)　迁移学习先验

如果代理在其他数据集上或者任务上已经得到了很好的训练，例如在 imageNet 数据集上训练的网络，那么代理本身就具有之前任务和数据集上的经验。通过依据对于现阶段任务的适应与调整，相比于没有人任何经验的新训练的代理，利用迁移学习的代理将会具有很强自步学习能力。从应用的经验上，在半监督的数据上，要达到和其他方法相比同样的精度，其所需的监督信息将会更少。

5 自步学习的总结与展望

自步学习这一起源于人和动物自主进行课程学习的方法论，核心在于充分利用了代理已有的直觉 $p_{model}(y|x)$，学习的样本和先验知识。自步学习最显著的作用在于，其减少了所需要监督信息，并且降低了糟糕的监督信息对于代理的影响。自步学习同时也附带了一种稳健的样本选择准则，这种准则也是学习过程中自适应的生成课程重要标准。

自步学习在半监督学习、弱监督学习和稳健学习等学习主题上取得值得关注的成果，一方面印证了自步学习对于数据具有一定不确定因素的学习主题可以有很好的适用性，另一方面表明自步学习在这些任务上所选取的难易程度标准包含了对于任务的本质的刻画。同时自步学习和迁移学习 (Transfer learning) 的配合使得其极为有益于成为极少样本学习的重要方法理论。

这篇自步学习的综述主要包含了自步学习发展史，基本方法论，有助于学习过程的各种课程先验，自步学习的凹共轭性和自步学习标准统一化的概率理解。自步学习的进一步发展，可能将会是自步学习理论的革新，并伴随着更加广泛和各种学习主题、学习理论的结合。

5.1 自步学习理论的展望

5.1.1 探索课程

自步学习在学习过程中，各阶段的学习课程 (操控的数据分布) 完全由代理的条件分布确定可能会使得学习过程当中的探索更加匮乏。Liang 等人[13]，在学习过程中将求得的权重视作采样的分布函数，将随机性引入学习过程。如果学习过程中的课程能以一定概率在几种可能的候选学习分布中进行采样，也许能够使得学习过程中能够存在更多的探索，学习的结果可能也会更加的稳健。

5.1.2 训练老师

Fan 等人[34]，利用深度强化学习的方法去训练一个自动的自适应的数据筛选器，用以作为老师。如果说自步学习自适应的课程是让代理做自己的老师，那么基于很多类型的代理，很多固定的数据集，训练一个通用的老师，以使得代理的学习的速度增快，并且能达到更好学习效果，可能是一个十分有意义的探索方向。

5.2 封闭环境下的自步学习

5.2.1 偏标记学习

偏标记学习（Partial label learning），又称为歧义学习，也属于数据具有不确定因素的主题。偏标记学习，设定每一个样本都伴随着一个候选标记集合，其中只有一个标记是真实标记，代理的任务是通过对于这样带有歧义的数据进行学习，以获得类别预测的能力。

从偏标记学习的设定中可以发现每个样本其实都有一定的不确定性，而代理需要能够通过对数据的逐步学习推断出数据的真实标记。

对于偏标记学习，一个比较有意思的样本难易程度的标准则可以表示为 $H(Y_{predict}|X, candidate\ label)$，（即已知样本和候选标记集代理预测标记的熵）通过优先对于代理容易预测的样本进行学习训练，可以预想到一些困难的样本的在代理看来的歧义将会逐渐变小（难度变小）。因此自步学习的方法论可能将会十分有利于偏标记学习问题。

5.2.2 噪音标记更正

自步学习和一些传统的学习方法在面对存在标记噪音的学习问题时，一般会通过更改损失函数，或者筛选样本的方式来解决。但是如果学习数据并不存在外来样本的情况，通过对于样本正确样本的推断并对样本标记进行更正，有可能可以使得学习取得更好效果[35]。

5.3 开放环境下的自步学习

在开放环境下，大量接触的数据将与任务无关，一个具有一定经验的代理，这个时候可以根据自步学习方法论的难易程度标准和可靠性标准，在无监督的情况下对于样本进行筛选和学习。

这十分类似于，好奇的人们会依据自身的经验，在一定的自我标准下，对于没有监督信息的样例，进行筛选，挑选自己熟悉的，能够有一定把握的样例进行学习。

Pathak 等人[36]，提出在强化学习（Reinforcement learning）的框架下，加入好奇心驱动的探索机制以使得代理能够在执行任务前就能够拥有一定的经验储备。在开放环境下，好奇心机制和自步学习方法论可能能形成很好结合。

5.4 自步学习与学习方法理论的结合

自步学习和一些其他方法论结合将会具有新的活力。

5.4.1 自步学习与主动学习

Lin 等人[15]，将主动学习（Active learning）和自步学习相结合，类似于学生先自己审视一下学习内容，将自己最有把握的内容直接学习，并向老师寻求对于把最没有把握的内容的指导。这种学习模式极其有可能成为代理基本的学习格式，并且拥有广泛的应用空间和经济价值。

5.4.2 自步学习与迁移学习

自步学习的核心在于代理的直觉。通过迁移学习，可以将代理关于类似任务的丰富经验和很好的表示（Representation），迁移到新的待处理任务上。一个拥有丰富经验，提好很好特征的代理将会具有很强的自步学习的能力，这将可能会大量减少训练所需要的监督信息量。

5.4.3 自步学习与预测学习

预测学习（Prediction learning）是无监督学习的范畴的一种衍生, 涵盖面包括对抗生成模型[37]，条件生成模型[38] 等无监督生成模型，自步学习可能将会有助于自适应选择简单的样本用于预测和生成的训练, 也可以帮助条件生成模型减少监督信息的需求。

参考文献

[1] Bengio Y, Louradour J, Collobert R, et al. Curriculum learning[C] // International Conference on Machine Learning. 2009 : 41 – 48.

[2] Pan S J, Yang Q. A survey on transfer learning[J]. IEEE Transactions on knowledge and data engineering, 2010, 22(10) : 1345 – 1359.

[3] Kumar M P, Packer B, Koller D. Self-paced learning for latent variable models[C] // Advances in Neural Information Processing Systems. 2010 : 1189 – 1197.

[4] Kumar M P, Turki H, Preston D, et al. Learning specific-class segmentation from diverse data[C] // IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November. 2011 : 1800 – 1807.

[5] Zhang D, Meng D, Li C, et al. A Self-Paced Multiple-Instance Learning Framework for Co-Saliency Detection[C] // IEEE International Conference on Computer Vision. 2015 : 594 – 602.

[6] Supancic J S, Ramanan D. Self-Paced Learning for Long-Term Tracking[C] // Computer Vision and Pattern Recognition. 2013 : 2379 – 2386.

[7] Huang W, Gu J, Ma X, et al. Self-paced model learning for robust visual tracking[J]. Journal of Electronic Imaging, 2017, 26(1) : 013016 – 013016.

[8] Zhang D, Meng D, Zhao L, et al. Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning[J]. arXiv preprint arXiv:1703.01290, 2017.

[9] Sangineto E, Nabi M, Culibrk D, et al. Self Paced Deep Learning for Weakly Supervised Object Detection[J]. arXiv preprint arXiv:1605.07651, 2016.

[10] Pi T, Li X, Zhang Z, et al. Self-paced boost learning for classification[C] // Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. 2016 : 1932 – 1938.

[11] Gu N, Fan M, Meng D. Robust Semi-Supervised Classification for Noisy Labels Based on Self-Paced Learning[J]. IEEE Signal Processing Letters, 2016, 23(12) : 1806 – 1810.

[12] Jiang L, Meng D, Yu S, et al. Self-Paced Learning with Diversity[J]. Advances in Neural Information Processing Systems, 2014 : 2078 – 2086.

[13] Liang J, Jiang L, Meng D, et al. Exploiting Multi-modal Curriculum in Noisy Web Data for Large-scale Concept Learning[J]. arXiv preprint arXiv:1607.04780, 2016.

[14] Yue Z, Meng D, He J, et al. Semi-supervised learning through adaptive Laplacian graph trimming[J]. Image and Vision Computing, 2016.

[15] Lin L, Wang K, Meng D, et al. Active Self-Paced Learning for Cost-Effective and Progressive Face Identification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[16] Zhao Q, Meng D, Jiang L, et al. Self-Paced Learning for Matrix Factorization.[C] // AAAI. 2015 : 3196 – 3202.

[17] Avramova V. Curriculum Learning with Deep Convolutional Neural Networks[D]. Stockholm : Royal Institute of Technology, 2015.

[18] Li C, Wei F, Yan J, et al. Self-Paced Multi-Task Learning[J]. arXiv preprint arXiv:1604.01474, 2016.

[19] Li C, Wei F, Yan J, et al. A Self-Paced Regularization Framework for Multi-Label Learning[J]. arXiv preprint arXiv:1603.06708, 2016.

[20] Zikeba M, Tomczak J M, Swikatek J. Self-paced Learning for Imbalanced Data[C] // Asian Conference on Intelligent Information and Database Systems. 2016 : 564 – 573.

[21] Meng D, Zhao Q. What Objective Does Self-paced Learning Indeed Optimize?[J]. Computer Science, 2015.

[22] Yu F, Zhang M-L. Maximum margin partial label learning[C] // Proceedings of the 7th Asian Conference on Machine Learning. 2015 : 96 – 111.

[23] Jiang L, Meng D, Mitamura T, et al. Easy Samples First: Self-paced Reranking for Zero-Example Multimedia Search[C] // ACM International Conference on Multimedia. 2014 : 547 – 556.

[24] Jiang L, Meng D, Zhao Q, et al. Self-Paced Curriculum Learning.[C] // AAAI : Vol 2. 2015 : 6.

[25] Jiang L, Yu S-I, Meng D, et al. Bridging the ultimate semantic gap: A semantic search engine for internet videos[C] // Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. 2015 : 27 – 34.

[26] Suzumura S, Ogawa K, Sugiyama M, et al. Outlier Path: A Homotopy Algorithm for Robust SVM.[C] // ICML. 2014 : 1098 – 1106.

[27] Fan Y, He R, Liang J, et al. Self-Paced Learning: an Implicit Regularization Perspective[J]. arXiv preprint arXiv:1606.00128, 2016.

[28] Ma Z, Liu S, Meng D. On Convergence Property of Implicit Self-paced Objective[J]. arXiv preprint arXiv:1703.09923, 2017.

[29] Platt J, others. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods[J]. Advances in large margin classifiers, 1999, 10(3) : 61 – 74.

[30] Cover T M, Thomas J A. Elements of information theory[M]. Hoboken, New Jersey : John Wiley & Sons, 2012.

[31] Meng D, De la torre F. Robust matrix factorization with unknown noise[C] // Proceedings of the IEEE International Conference on Computer Vision. 2013 : 1337 – 1344.

[32] Zhao Q, Meng D, Xu Z, et al. Robust Principal Component Analysis with Complex Noise.[C] // ICML. 2014 : 55 – 63.

[33] Blum A, Mitchell T. Combining labeled and unlabeled data with co-training[C] // Proceedings of the eleventh annual conference on Computational learning theory. 1998 : 92 – 100.

[34] Fan Y, Tian F, Qin T, et al. Learning What Data to Learn[J]. arXiv preprint arXiv:1702.08635, 2017.

[35] Gidaris S, Komodakis N. Detect, Replace, Refine: Deep Structured Prediction For Pixel Wise Labeling[J]. arXiv preprint arXiv:1612.04770, 2016.

[36] Pathak D, Agrawal P, Efros A A, et al. Curiosity-driven Exploration by Self-supervised Prediction[J]. arXiv preprint arXiv:1705.05363, 2017.

[37] Goodfellow I, Pouget-abadie J, Mirza M, et al. Generative adversarial nets[C] // Advances in neural information processing systems. 2014 : 2672 – 2680.

[38] Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.

[39] Matthews L, Ishikawa T, Baker S. The template update problem[J]. IEEE transactions on pattern analysis and machine intelligence, 2004, 26(6) : 810 – 815.

[40] Rockafellar R T. Convex Analysis[M]. Princeton, New Jersey : Princeton University Press, American Elsevier Publishing Co., 1970.

附录 1　自步学习应用领域成果汇总

附录 1.1　弱监督学习中的应用

附录 1.1.0.1　图像分割

通过自步学习的方法论，Kumar 等人[4]2011 年提出了一个隐变量支持向量机的框架，其中她们的隐变量建模可以指代对于学习具体类别的分割所有缺失了人工标记信息。其中主要包括三种类别的标记：图像分割中利用通用背景或者前景类别标记，图像对象的包围框标记，图片的类别标记。她们的方法利用多样的数据，在公开的数据集（VOC2009 分割数据集，有逐像素前景类别分割；SBD 有逐像素的对于背景类别的分割；VOC2010 检测数据集，有前景类别对应实例的包围框；Imagenet，有前景图片级别的类别标注）上获得了不错的成绩。

附录 1.1.0.2　共同显著性检测

共同显著性检测是同时的从一组图片当中提取共同的显著的目标。Zhang 等人[5]2015 年对于共同显著性检测提出了多实例学习的格式，并将自步学习与其结合。代理意在自步地确定是否每一个超像素区域都属于共同显著的区域、他们的模型 SP-MIL 在参照数据集 iCoseg 和 MSRC 上，不管是平均精度和 F 值的评价标准上都超越了现有的最新水平的方法。并且在目标共同分割的任务上，在一个大规模的包含了千张网络带噪的图片的基准数据集上，SP-MIL 在 Jaccard 相似度的评价标准上也超过了最新水平的方法。

附录 1.1.0.3　追踪定位

追踪问题是自步学习一个较早的应用。在目标长时间追踪过程中自适应学习目标的外观的一个最大麻烦就是图像可能会出现漂移[39]，就是图像框中可能并没有完全包含对象。利用自步学习的方法，Supancic 等人[6]2013 年使得代理能够很谨慎的选在选择跟踪的框架并且最小化漂移以此来学习一个对于长时间跟踪的更好的外观检测模型。其最后的在线系统，能够达到 $\frac{1}{20}^{th}$ 实时，并且在基准数据集上显著的以 91%vs76% 的准确率超越了当时给予了很多不公平优势的最新水平的跟踪算法 (TLD)。

利用类似的想法，Huang 等人[7]2017 年在一个近期的跟踪基准数据集 (OTB2013) 其包括 50 个视频序列, 比较了 22 种最新水平的跟踪方法，他们的自步学习的跟踪方法在精度上达到了 0.785 的得分超过了得分第二高 (0.740) 的 KCF 跟踪方法 4.5%。同时他们的 AUC 评价得分为 0.540，超过 KCF 方法 (0.514)2.6%

附录 1.1.0.4　目标检测

自步学习 2017 年也正在计算机视觉当中的一个具有挑战性的弱监督的目标检测领域发挥着影响。弱监督目标检测的主要困难就是要利用所给的很少的弱的图像级别的标注信息来同时推断出目标具体的位置并且加以对目标检测器进行训练。Zhang 等人[8] 通过自步课程学习将显著性检测和弱监督目标检测建立器联系使得算法能够逐渐的从多类目标当中从简单到困难的获得忠实的信息。他们的方法打破了 6 个最新水平的弱监督检测的方法 (PR,CC,MDD,LLO,VPC,MfMIL) 取得了 Pascal VOC2007 数据集上 29.96 的平均得分。

对于利用自步学习减少使用的监督信息的探索也更加深入。Dong 等人①，研究了利用大量无标注图像和每类 3 到 4 个有标注的图像的目标检测任务。通过利用自步集成学习方法，在了 Pascal VOC2007 上，对比那些最新水平但利用了很大部分标注信息的弱监督学习方法，他们的方法取得了十分具有竞争力的结果 (平均精度为:42.8%(最高得分方法)vs42.8%(自步学习方法)，正确定位：64.6%(2^{th})vs 65.5%(自步学习方法，最高得分方法))。

对于弱监督目标检测的自步深度学习的研究还包括 [9]。

附录 1.2　稳健学习中的应用

要从带有大噪音的样本中提取出可靠地监督信息，机器学习算法的稳健性扮演着重要的作用。最新涌现出来利用自步学习的稳健性的研究兴趣对于稳健的学习产生了很大影响，包括 boosting 分类[10]，带有噪声的利用流形正则的半监督分类[11]，大规模的噪音网络数据概念学习[13]。

① 该文章尚未公布互联网版本

附录 1.2.0.5　稳健 Boosting 学习

Pi 等人[10], 将自步学习和 boosting 算法相结合，提出了 SPBL 模型，来共同提升模型算法的有效性和稳健性。通过最大分类边界的 boosting 优化和自步学习的样本选择，SPBL 能够捕捉到类间本质的决定性特征同时保证学习样本的可靠性。他们在若干个真实数据集上的实验结果展现了 SPBL 算法的优越性、

附录 1.2.0.6　网络标识视频数据目标检测[13]

基于自步学习和偏序课程以及随机丢弃样本的方法，作者提出一种新颖的 WELL 方法来做网络标记的数据 (带噪声的弱标注的网上视频数据)。在 FCVID(复旦哥伦比亚视频数据集，最大的人工标记视频数据集) 上，他们的 WELL 方法利用 2000 小时的有噪声标记的视频取得的精度比在利用 500 小时人工标记的数据更高。

附录 1.3　半监督学习中的应用

附录 1.3.0.7　协同训练

协同训练是半监督学习当中的一种方法，它假设样本具有两个不同的视角。这两个视角被假定是具有互补性的，利用不同视角代理，迭代的对于可靠性高的未数据进行标注和进一步训练。Ma 等人①，将协同训练和自步学习相结合，使得学习过程错误伪标注的数据可以有机会被重新召回更正，同时更多先验信息也被利用于协同训练之中。他们的方法在常用的协同训练 UCI 数据集上超过了最新水平的方法。

附录 1.3.0.8　自适应的拉普拉斯图割

基于图的半监督学习方法严重的依赖于编码了输入样本相似程度的拉普拉斯加权图。粗糙建立拉普拉斯加权图可能会导致短路现象 (本来距离很远的流形上点被连接起来了) 和数据流形结构的不充足的表示。Yue 等人[14] 提出了一种自适应的拉普拉斯图调整 (ALGT) 方法，以实现自动的调整去截断簇间的不恰当的短路边并同时实现对于簇内连接的强化，从而可以自适应的依据数据拟合一个恰当的拉普拉斯加权图。这个方法的优越性通过大量的 UCI 数据集和合成数据上的的实验结果所支持。

① 该文章被 ICML2017 收录但互联网版本尚未公开

附录 1.3.0.9　主动学习

Lin 等人[15] 将自步学习和主动学习相结合，其中自步学习用于确定无标注样本的可靠性程度，将可靠性的样本直接用于模型的训练同时将可靠性低的样本交给主动的用户咨询。这使得需要的有标注的样本数量显著下降但是同时模型依旧可以取得很具有竞争力的表现。这种方法对比其他的最新水平的方法极大的减少了用用户的投入。通过将自步学习和主动学习相结合不但有效的提升了分类器的精度同时也提升模型对于噪音样本的稳健性。

附录 1.4　无监督学习中的应用

附录 1.4.0.10　多媒体事件检测

Jiang 等人[12] 提出了自步重排序 (SPaR) 模型将自步学习引入了多媒体事件检测领域。SPaR 通过从易到难的自步的利用学习样本模式，取得了多媒体事件检测的算法的最新水平。并且 SPaR 在 TRECVID 多媒体事件搜索任务中成为了取得了大幅领先的最好成果。

附录 1.4.0.11　矩阵分解

矩阵分解有着广泛的应用但这类问题普遍是非凸的并且现有的方法很容易陷入一个不好的局部极优点。Zhao 等人[16] 将自步学习与加权矩阵分解相结合，将传统的二值分解推广到实数的加权格式上。自步学习矩阵分解的方法在大量的生成数据和运动背景分离数据上取得了不错的效果。

附录 1.4.0.12　多视角聚类

从多个视角当中发掘信息可以提升聚类算法的精度，但是现有多视角聚类很容易陷入一个糟糕的局部极优点，特别是存在一些数据缺失和大噪音数据的时候。为了缓解这个问题，Xu 等人[16] 提出针对聚类的多视角自步学习模型，使得学习的过称不光能从简单的样本到难样本进行学习，也可以从简单的视角到难的视角进行学习。他们从理论的角度说明利用多视角来聚类和对于定义样本和视角的难度有助于理想的聚类，并且在玩具数据和真实数据上他们取得了不错的实验结果。

附录 1.5　监督学习中的应用

很多极度非凸全监督学习问题，如何达到一个更好的局部极优是一个优化的难题。受到自步学习从易到难的学习过程的影响，很多研究的兴趣聚焦于能否改变对样本或者任务或者标记学习的顺序来实现一个更好的路径算法得到一个更好的局部极优解。Avramova 等人[17], 在大型有标注图像数据集中利用反自步学习的从难到易样本学习过程来训练深度卷积神经网络，取得相较于传统训练方法更好的表现.Li 等人[18,19]，将自步学习与多标记和多任务学习相结合使得学习的过程不仅可以依据样本从易到难的学习，同时也依据标记和任务进行从易到难的学习。

附录 1.6　迁移学习和样本筛选中的应用

迁移学习是利用一种学习上的经验迁移到另一种学习任务上。自步学习自适应的操控所学习数据分布能力，为重构一个和目标任务更接近的所需学习数据分布提供了很好的帮助和格式。Zikeba 等人[20] 将自步学习引入到高度不均衡的数据集任务学习中，使得代理可以自适应的调整所学习的数据分布，是其在学习的过程中更加均衡。他们的方法在高度不均衡的基准数据集上取得了不错的效果。

利用迁移学习得到的代理可用拥有更多的经验和更好的特征表示，也在一定程度上可能使其拥有更强的自步学习的能力。

附录 2　证明部分

性质 附录 2.1 (次微分的对偶性) *对于任意闭的恰当的凹函数 g 和任何向量 v, 下面关于 l 的四个条件是等价的:*

- $l \in \partial g(v)$
- $\langle z, l\rangle - g(z)$ *关于* z *在* $z = v$ *达到最小值*
- $g(v) + g^*(l) = \langle z, l\rangle$
- $v \in \partial g^*(l)$
- $\langle v, z\rangle - g(z)$ *关于* z *在* $z = l$ *达到最小值*

这个性质完全和原始的凸函数拥有的性质对称，参考书籍[40] 213−226,307−308。

定理 附录 2.1 (本质严格凸的本质光滑的对偶性[40] 251−254) *一个闭的恰当的凸函数是本质严格凸的当且仅当它的共轭是本质严格光滑的。*

推论 附录 2.1 (严格凸意味着可微性) *如果 f 是定义域有界的一个闭的严格凸函数, 那么 f^* 是一个闭的全空间上的可微的函数。*

证明 附录 2.1 *f 定义域有界 $\Rightarrow$ f 是联合有限 $\Rightarrow$[40] 116−117 f^* 在全空间有定义。f 是严格凸的 $\Rightarrow$[40] 251−254 基本严格凸。依据定理 附录 2.1, f^* 是全空间上本质光滑的，这意味着 f^* 在全空间可微*[40] 251−254。

定理 4.1 [模型等价性] 如果 v 是一维的, 且 $R_{SP}(v, \lambda)$ 关于 v 严格凸，下半连续，同时 $dom_v R_{SP}(v, \lambda) \subset [0, 1]$ 且 $0, 1 \in cl(dom_v R_{SP}(v, \lambda))$ 那么

$$F_\lambda(l) = \int_0^l v(\lambda, j)\, dj + C(\lambda) \tag{附录 2-1}$$

其中 $C(\lambda)$ 是 λ 的函数。

证明 附录 2.2 *根据性质 附录 2.1 和推论 附录 2.1,* $v(\lambda, l) = \nabla F_\lambda(l)$*。因此,*

$$F_\lambda(l) = \int_0^l v(\lambda, j)\, dj + C(\lambda) \tag{附录 2-2}$$

其中 $C(\lambda)$ *是关于* λ *的函数。*

附录 3　自步学习贝叶斯网的实验

附录 3.1　实验

为了研究自步学习的误差建模模型中，总体性来源性先验与偏序先验加入对于隐藏目标函数的影响，我们设计了如下试验。

信号形式假设

$$Y = 10sinc(X - T) + \varepsilon, \tag{附录 3-1}$$

其中 T 为模型参数, 假设利用两个测量仪器 (对应精度 $variance_1 = 5, variance_2 = 1$) 依次对于 Y 在一系列 X 上进行测量，测量误差服从

$$\varepsilon|variance \sim \mathcal{N}(0, variance^2)。 \tag{附录 3-2}$$

其中取 $T = 3$ 进行两次实验，信号图形和测量结果如下。

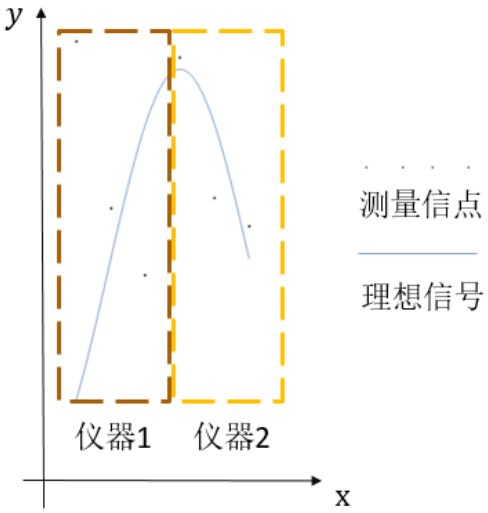

图 附录 3-1　第一次实验：两仪器测量结果图

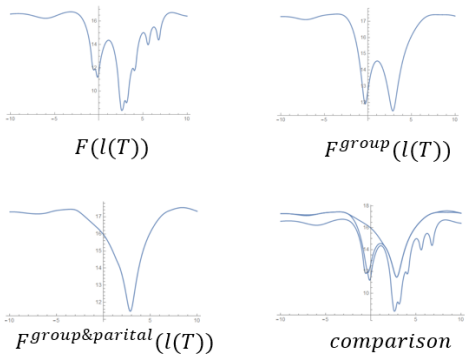

图 附录 3-2　第一次实验：目标函数对照图

分别计算出对应自步学习的隐藏目标函数 $F(l(T))$（公式见 4-48, 3.3.1.1， 4-17）关于模型参数的图像，其中第一幅是不加入任何先验知识的函数图像，第二幅是加入总体性先验的后新的目标函数的图像，第三幅是加入总体先验并加入偏序先验后目标函数的图像，第四幅是总的对比图。

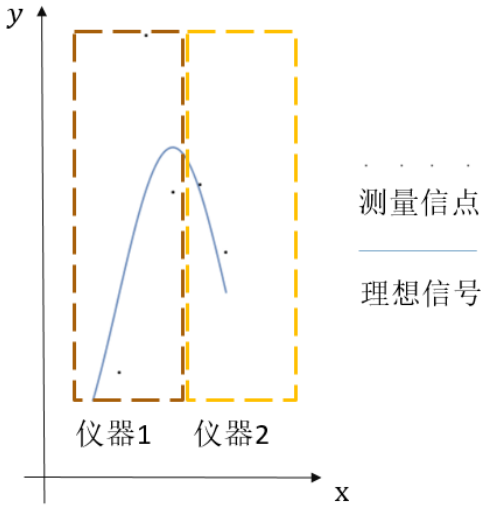

图 附录 3-3　第二次实验：两仪器测量结果图

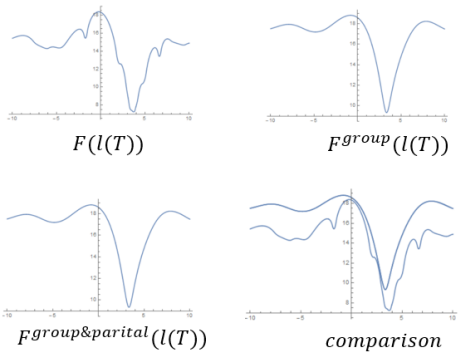

图 附录 3-4　第二次实验：目标函数对照图

发现：从实验结果可以看出总体先验和偏序先验的加入，使得对应优化的目标函数变得更加光滑，同时局部极优点的数量呈现显著的下降。

附录 3.2　Mathematica 代码

```
A = 10;
n1 = 3;
n2 = 5;
varl1 = 5;
varl2 = 1;
t = (n1 + n2 - 1)/2;
F[L_] := 1/(2) + 1/2 Log[2 L] /; (L > 1/(2))
F[L_] := L /; (L <= 1/(2))
(*Plot3D[F[L1]+F[L2],{L1,-2,10},{L2,-2,+10}]
ContourPlot[F[L1]+F[L2],{L1,-2,10},{L2,-2,+10}]*)
Data1 = Table[{j,
    A Sinc[j - RandomVariate[NormalDistribution[t, 0.01], 1]] +
     RandomVariate[NormalDistribution[0, varl1], 1]}, {j, 0, n1 - 1,
    1}];
Data2 = Table[{j,
    A Sinc[j - RandomVariate[NormalDistribution[t, 0.01], 1]] +
     RandomVariate[NormalDistribution[0, varl2], 1]}, {j, n1,
    n1 + n2 - 1, 1}];

sequence1 = Flatten[Data1];
sequence2 = Flatten[Data2];
sequence = Partition[Flatten[Append[sequence1, sequence2]], 2];
P1 = Plot[A Sinc[x - t], {x, 0, n1 + n2 - 1}];
L1 = Graphics[Point[sequence]];
Show[L1, P1]未加入先验的
(*F(1)*)
L[x_, y_] := (y - A Sinc[x - g])^2;
```

```
FL[x_, y_] := F[L[x, y]];
P21 = Plot[Table[1, n1 + n2].Apply[FL, sequence, {1}], {g, −10, 10},
  PlotRange −> All]加入前
(*样本误差方差相等先验，加入后样本精度相等先验n1n2*)
F1[L_] := n1/(2) + n1/2 Log[2 L/n1] /; (L > n1/(2))
F1[L_] := L /; (L <= n1/(2))
F2[L_] := n2/2 + n2/2 Log[2 L/n2] /; (L > n2/2)
F2[L_] := L /; (L <= n2/2)
P1110 = Plot[
  F1[Table[1, n1].Apply[L, Partition[sequence1, 2], {1}]] +
   F2[Table[1, n2].Apply[L, Partition[sequence2, 2], {1}]], {g, −10,
   10}, PlotRange −> All]
(*Plot3D[F1[g]+F2[g1],{g,−2,10},{g1,−2,100}]*)
F12[x_, y_] := F1[x] + F2[y] /; (n2 x − n1 y > 0)
F12[x_, y_] :=
 F1[(x + y)/(1 + n2/n1)] +
   F2[(x + y)/(1 + n1/n2)] /; (n2 x − n1 y <= 0)
Plot3D[F12[x, y] − F1[x] − F2[y], {x, −10, 10}, {y, −10, 10}]
P11102 = Plot[
  F12[Table[1, n1].Apply[L, Partition[sequence1, 2], {1}],
   Table[1, n2].Apply[L, Partition[sequence2, 2], {1}]], {g, −10, 10},
   PlotRange −> All]
Show[P21, P1110, P11102, PlotRange −> All]
Plot[F12[Table[1, n1].Apply[L, Partition[sequence1, 2], {1}],
   Table[1, n2].Apply[L, Partition[sequence2, 2], {1}]] − (F1[
     Table[1, n1].Apply[L, Partition[sequence1, 2], {1}]] +
    F2[Table[1, n2].Apply[L, Partition[sequence2, 2], {1}]]), {g, −10,
   10}]
ParametricPlot[{{Table[1, 11].Apply[L, Partition[sequence1, 2], {1}],
   Table[1, 10].Apply[L, Partition[sequence2, 2], {1}]}, {100 g,
   100 (g − (n1 − n2)/2)}}, {g, −10, 10}]
```

附录 4　外文翻译

Automated Curriculum Learning for Neural Networks

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, Koray Kavukcuoglu
{gravesa, bellemare, jmenick, munos, korayk}@google.com
Google DeepMind, London UK

Abstract

We introduce a method for automatically selecting the path, or syllabus, that a neural network follows through a curriculum so as to maximise learning efficiency. A measure of the amount that the network learns from each data sample is provided as a reward signal to a nonstationary multi-armed bandit algorithm, which then determines a stochastic syllabus. We consider a range of signals derived from two distinct indicators of learning progress: rate of increase in prediction accuracy, and rate of increase in network complexity. Experimental results for LSTM networks on three curricula demonstrate that our approach can significantly accelerate learning, in some cases halving the time required to attain a satisfactory performance level.

1. Introduction

Over two decades ago, in *The importance of starting small*, Elman put forward the idea that a curriculum of progressively harder tasks could significantly accelerate a neural network's training (Elman, 1993). However curriculum learning has only recently become prevalent in the field (e.g., Bengio et al., 2009), due in part to the greater complexity of problems now being considered. In particular, recent work on learning programs with neural networks has relied on curricula to scale up to longer or more complicated programs (Sutskever and Zaremba, 2014; Reed and de Freitas, 2015; Graves et al., 2016). We expect this trend to continue as the scope of neural networks widens.

One reason for the slow adoption of curriculum learning is that its effectiveness is highly sensitive to the mode of progression through the tasks. One popular approach is to define a hand-chosen performance threshold for advancement to the next task, along with a fixed probability of returning to earlier tasks, to prevent forgetting (Sutskever and Zaremba, 2014). However, as well as introducing hard-to-tune parameters, this poses problems for curricula where appropriate thresholds may be unknown or variable across tasks. More fundamentally, it presupposes that the tasks can be ordered by difficulty, when in reality they may vary along multiple axes of difficulty, or have no predefined order at all.

We propose to instead treat the decision about which task to study next as a stochastic policy, continuously adapted to optimise some notion of what Oudeyer et al. (2007) termed *learning progress*. Doing so brings us into contact with the intrinsic motivation literature (Barto, 2013), where various indicators of learning progress have been used as reward signals to encourage exploration, including compression progress (Schmidhuber, 1991), information acquisition (Storck et al., 1995), Bayesian surprise (Itti and Baldi, 2009), prediction gain (Bellemare et al., 2016) and variational information maximisation (Houthooft et al., 2016). We focus on variants of prediction gain, and also introduce a novel class of progress signals which we refer to as complexity gain. Derived from minimum description length principles, complexity gain equates acquisition of knowledge with an increase in effective information encoded in the network weights.

Given a progress signal that can be evaluated for each training example, we use a multi-armed bandit algorithm to find a stochastic policy over the tasks that maximises overall progress. The bandit is nonstationary because the behaviour of the network, and hence the optimal policy, evolves during training. We take inspiration from a previous work that modelled an adaptive student with a multi-armed bandit in the context of developmental learning (Lopes and Oudeyer, 2012). Another related area is the field of active learning, where similar gain signals have been used to guide decisions about which data point to label next (Settles, 2010). Lastly, there are parallels with recent work on using Bayesian optimisation to find the best order in which to train a word embedding network on a language corpus (Tsvetkov, 2016); however this differs from our work in that the ordering was entirely determined *be-*

fore each training run, rather than adaptively altered in response to the model's progress.

2. Background

We consider supervised or unsupervised learning problems where target sequences $\mathbf{b}^1, \mathbf{b}^2, \ldots$ are conditionally modelled given their respective input sequences $\mathbf{a}^1, \mathbf{a}^2, \ldots$. For convenience we suppose that the targets are drawn from a finite set $\mathcal{B}$, noting our framework extends to continuous targets, with densities taking the place of probabilities. As is typical for neural networks, sequences may be grouped together in batches $(\mathbf{b}^{1:B}, \mathbf{a}^{1:B})$ to accelerate training. The conditional probability output by the model is

$$p(\mathbf{b}^{1:B} \,|\, \mathbf{a}^{1:B}) = \prod_{i,j} p(\mathbf{b}^i_j \,|\, \mathbf{b}^i_{1:j-1}, \mathbf{a}^i_{1:j-1}).$$

From here onwards, we consider each batch as a single example $\mathbf{x}$ from $\mathcal{X} := (\mathcal{A} \times \mathcal{B})^N$, and write $p(\mathbf{x}) := p(\mathbf{b}^{1:B} \,|\, \mathbf{a}^{1:B})$ for its probability. Under this notation, a *task* is a distribution D over sequences from $\mathcal{X}$. A *curriculum* is an ensemble of tasks $D_1, \ldots, D_N$, and a *sample* is an example drawn from one of the tasks of the curriculum. Finally, a *syllabus* is a time-varying sequence of distributions over tasks.

We consider a neural network to be a parametric probabilistic model p_θ over $\mathcal{X}$, whose parameters are denoted θ. The expected loss of the network on the k^{th} task is

$$\mathcal{L}_k(\theta) := \mathop{\mathbb{E}}_{\mathbf{x} \sim D_k} L(\mathbf{x}, \theta),$$

where $L(\mathbf{x}, \theta) := -\log p_\theta(\mathbf{x})$ is the sample loss on $\mathbf{x}$. Whenever unambiguous, we will simply denote the expected and sample losses by $\mathcal{L}_k$ and $L(\mathbf{x})$ respectively.

2.1. Curriculum Learning

We consider two related settings. In the *multiple tasks* setting, The goal is to perform as well as possible on all tasks in the ensemble $\{D_k\}$; this is captured by the objective function

$$\mathcal{L}_{\text{MT}} := \frac{1}{N} \sum_{k=1}^{N} \mathcal{L}_k.$$

In the *target task* setting, we are only interested in minimizing the loss on the final task D_N. The other tasks then act as a series of stepping stones to the real problem. The objective function in this setting is simply $\mathcal{L}_{\text{TT}} := \mathcal{L}_N$.

2.2. Adversarial Multi-Armed Bandits

We view a curriculum containing N tasks as an N-armed bandit (Bubeck and Cesa-Bianchi, 2012), and a syllabus as an adaptive policy which seeks to maximize payoffs from this bandit. In the bandit setting, an agent selects a sequence of arms (actions) $a_1 \ldots a_T$ over T rounds of play ($a_t \in \{1, \ldots, N\}$). After each round, the selected arm yields a payoff r_t; the payoffs for the other arms are not observed.

The classic algorithm for adversarial bandits is Exp3 (Auer et al., 2002), which uses multiplicative weight updates to guarantee low regret with respect to the best arm. On round t, the agent selects an arm stochastically according to a policy π_t. This policy is defined by a set of weights $w_{t,i}$:

$$\pi_t^{\text{EXP3}}(i) := \frac{e^{w_{t,i}}}{\sum_{j=1}^{N} e^{w_{t,j}}}.$$

The weights are the sum of importance-sampled rewards:

$$w_{t,i} := \eta \sum_{s<t} \tilde{r}_{s,i} \qquad \tilde{r}_{s,i} := \frac{r_s \mathbb{I}_{[a_s=i]}}{\pi_s(i)}.$$

Exp3 acts so as to minimize regret with respect to the single best arm evaluated over the whole history. However, a common occurrence is for an arm to be optimal for a portion of the history, then another arm, and so on; the best strategy is then piecewise stationary. This is generally the case in our setting, as the expected reward for each task changes as the model learns. The Fixed Share method (Herbster and Warmuth, 1998) addresses this issue by using an ϵ-greedy strategy and mixing in the weights additively. In the bandit setting, this is known as the Exp3.S algorithm (also by Auer et al. (2002)):

$$\begin{aligned} \pi_t^{\text{EXP3.P}}(i) &:= (1-\epsilon)\pi_t^{\text{EXP3}}(i) + \frac{\epsilon}{N} \qquad (1) \\ w^{\text{S}}_{t,i} &:= \log\Big[(1-\alpha_t)\exp\Big\{w^{\text{S}}_{t-1,i} + \eta\tilde{r}^{\beta}_{t-1,i}\Big\} \\ &\quad + \frac{\alpha_t}{N-1}\sum_{j\neq i}\exp\Big\{w^{\text{S}}_{t-1,j} + \eta\tilde{r}^{\beta}_{t-1,j}\Big\}\Big] \\ w^{\text{S}}_{1,i} &:= 0 \qquad \alpha_t := t^{-1} \qquad \tilde{r}^{\beta}_{s,i} := \frac{r_s\mathbb{I}_{[a_s=i]} + \beta}{\pi_s(i)}. \end{aligned}$$

2.3. Reward Scaling

The appropriate step size η depends on the magnitudes of the rewards, which may not be known *a priori*. The problem is particularly acute in our setting, where the magnitude depends on how learning progress is measured, and varies over time as the model learns. To address this issue, we adaptively rescale all rewards to lie in the interval $[-1, 1]$ using the following procedure: Let $\mathcal{R}_t$ be the history of unscaled rewards up to time t, i.e. $\mathcal{R}_t = \{\hat{r}_i\}_{i=1}^{t-1}$. Let q_t^{lo} and q_t^{hi} be quantiles of $\mathcal{R}_t$, which we choose here to be the 20th and 80th percentiles respectively. The scaled reward r_t is obtained by clipping $\hat{r}_t$ to the interval $[q_t^{\text{lo}}, q^{\text{hi}_t}]$

and then linearly mapping the result to lie in $[-1, 1]$:

$$r_t = \begin{cases} -1 & \text{if } \hat{r}_t < q_t^{\text{lo}} \\ 1 & \text{if } \hat{r}_t > q_t^{\text{hi}} \\ \frac{2(\hat{r}_t - q_t^{\text{lo}})}{q_t^{\text{hi}} - q_t^{\text{lo}}} - 1 & \text{otherwise.} \end{cases} \quad (2)$$

Rather than keeping the entire history of rewards, we use reservoir sampling to maintain a representative sample, and compute approximate quantiles from this sample. These quantiles can be obtained in $\Theta(log|\mathcal{R}_t|)$ time.

3. Learning Progress Signals

Our goal is to use the policy output by Exp3.S as a syllabus for training our models. Ideally we would like the policy to maximize the rate at which we minimize the loss, and the reward should reflect this rate – what Oudeyer et al. (2007) calls *learning progress*. However, it usually is computationally undesirable or even impossible to measure the effect of a training sample on the target objective, and we therefore turn to surrogate measures of progress. Broadly, these measures are either 1) loss-driven, in the sense that they equate reward with a decrease in some loss; or 2) complexity-driven, when they equate reward with an increase in model complexity.

Training proceeds as follows: at each time t, we first sample a task index $k \sim \pi_t$. We then generate a sample from this task, i.e. $\mathbf{x} \sim D_k$. Note that each $\mathbf{x}$ is in general a batch of training sequences, and that in order to reduce noise in the gain signal we draw the whole batch from a single task. We compute the chosen measure of learning progress ν then divide by the time $\tau(\mathbf{x})$ required to process the sample (since it is the *rate* of progress we are concerned with, and processing time may vary from task to task) to get the raw reward $\hat{r} = \nu/\tau(\mathbf{x})$ For the purposes of this work, $\tau(\mathbf{x})$ was simply the length of the longest input sequence in $\mathbf{x}$; for other tasks or architectures a more complex calculation may be required. We then rescale $\hat{r}$ into a reward $r_t \in [-1, 1]$, and provide it to Exp3.S. The procedure is summarized as Algorithm 1.

3.1. Loss-driven Progress

We consider five loss-driven progress signals, all which compare the predictions made by the model before and after training on some sample $\mathbf{x}$. The first two signals we present are instantaneous in the sense that they only depend on $\mathbf{x}$. Such signals are appealing because they are typically cheaper to evaluate, and are agnostic about the overall goal of the curriculum. The remaining three signals more directly measure the effect of training on the desired objective, but require an additional sample $\mathbf{x}'$. In what follows we denote the model parameters before and after training on $\mathbf{x}$ by θ and θ' respectively.

Algorithm 1 Intrinsically Motivated Curriculum Learning

Initially: $w_i = 0$ for $i \in [N]$

for $t = 1 \dots T$ **do**
  $\pi(k) := (1 - \epsilon)\frac{e^{w_k}}{\sum_i e^{w_i}} + \frac{\epsilon}{N}$
  Draw task index k from π
  Draw training sample $\mathbf{x}$ from D_k
  Train network p_θ on $\mathbf{x}$
  Compute learning progress ν (Sections 3.1 & 3.2)
  Map $\hat{r} = \nu/\tau(\mathbf{x})$ to $r \in [-1, 1]$ (Section 2.3)
  Update w_i with reward r using Exp3.S (1)
end for

Prediction gain (PG). Prediction gain is defined as the instantaneous change in loss for a sample $\mathbf{x}$, before and after training on $\mathbf{x}$:

$$\nu_{PG} := L(\mathbf{x}, \theta) - L(\mathbf{x}, \theta').$$

When p_θ is a Bayesian mixture model, prediction gain upper bounds the model's information gain (Bellemare et al., 2016), and is therefore closely related to the Bayesian precept that learning is a change in posterior.

Gradient prediction gain (GPG). Computing prediction gain requires an additional forward pass. When p_θ is differentiable, an alternative is to consider the first-order Taylor series approximation to prediction gain:

$$L(\mathbf{x}, \theta') \approx L(\mathbf{x}, \theta) + [\nabla L(\mathbf{x}, \theta)]^\top \Delta_\theta,$$

where Δ_θ is the descent step. Taking this step to be the negative gradient $-\nabla_\theta L(\mathbf{x}, \theta)$ we obtain the gradient prediction gain

$$\nu_{GPG} := \|\nabla L(\mathbf{x}, \theta)\|_2^2.$$

This measures the magnitude of the gradient vector, which has been used an indicator of data salience in the active learning literature (Settles et al., 2008). We will show below that gradient prediction gain is a biased estimate true expected learning progress, and in particular favours tasks whose loss has higher variance.

Self prediction gain (SPG). Prediction gain is a biased estimate of the change in $\mathcal{L}_k(\theta)$, the expected loss on task k. Having trained on $\mathbf{x}$, we naturally expect the sample loss $L(\mathbf{x}, \theta)$ to decrease, even though the loss at other points may increase. Self prediction gain addresses this issue by sampling a second time from the same task and estimating progress on the new sample:

$$\nu_{SPG} := L(\mathbf{x}', \theta) - L(\mathbf{x}', \theta') \qquad \mathbf{x}' \sim D_k.$$

Target prediction gain (TPG). We can take the self-prediction gain idea further and evaluate directly on the loss of interest, which has has also been considered in active learning (Roy and Mccallum, 2001). In the target task setting, this becomes

$$\nu_{TPG} := L(\mathbf{x}', \theta) - L(\mathbf{x}', \theta') \qquad \mathbf{x}' \sim D_N.$$

Although this might seem like the most accurate measure so far, it tends to suffer from high variance, and also runs counter to the premise that, early in training, the model cannot improve on the difficult target task and should instead train on a task that it can master.

Mean prediction gain (MPG). Mean prediction gain is the analogue of target prediction gain in the multiple tasks setting, where it is natural to evaluate our progress across all tasks. We write

$$\nu_{MPG} := L(\mathbf{x}', \theta) - L(\mathbf{x}', \theta') \qquad \mathbf{x}' \sim D_k, k \sim U_N,$$

where U_N denotes the uniform distribution over $\{1, \ldots, N\}$. Mean prediction gain has additional variance from sampling an evaluation task $k \sim U_N$.

3.2. Complexity-driven Progress

So far we have considered gains that gauge the network's learning progress directly, by observing the rate of change in its predictive ability. We now turn to a novel set of gains that instead measure the rate at which the network's complexity increases. These gains are underpinned by the Minimum Description Length (MDL) principle (Rissanen, 1986; Grünwald, 2007): in order to best generalise from a particular dataset, one should minimise the number of bits required to describe the model parameters plus the number of bits required for the model to describe the data.

According to the MDL principle, increasing the model complexity by a certain amount is only worthwhile if it compresses the data by a greater amount. We would therefore expect the complexity to increase most in response to the training examples from which the network is best able to generalise. These examples are exactly what we seek when attempting to maximise learning progress.

MDL training for neural networks (Hinton and Van Camp, 1993) can be practically realised with stochastic variational inference (Graves, 2011; Kingma et al., 2015; Blundell et al., 2015). In this framework a variational posterior $P_\phi(\theta)$ over the network weights is maintained during training, with a single weight sample drawn for each training example. The parameters ϕ of the posterior are optimised, rather than θ itself. The total loss is the expected log-loss of the training dataset[1] (which in our case is the complete curriculum), plus the KL-divergence between the posterior and some fixed (Blundell et al., 2015) or adaptive (Graves, 2011) prior $Q_\psi(\theta)$:

$$L_{VI}(\phi, \psi) = \underbrace{KL(P_\phi \,\|\, Q_\psi)}_{\text{model complexity}} + \underbrace{\sum\nolimits_k \sum_{\mathbf{x} \in D_k} \mathop{\mathbb{E}}_{\theta \sim P_\phi} L(\mathbf{x}, \theta)}_{\text{data cost}}.$$

Since we are using stochastic gradient descent we need to determine the per-sample loss for both the model complexity and the data. Defining $S := \sum_k |D_k|$ as the total number of samples in the curriculum we obtain

$$L_{VI}(\mathbf{x}, \phi, \psi) := \frac{1}{S} KL(P_\phi \,\|\, Q_\psi) + \mathop{\mathbb{E}}_{\theta \sim P_\phi} L(\mathbf{x}, \theta), \quad (3)$$

with $L_{VI}(\phi, \psi) = \sum_k \sum_{\mathbf{x} \sim D_k} L_{VI}(\mathbf{x}, \phi, \psi)$. Some of the curricula we consider are algorithmically generated, meaning that the number of samples in each task is undefined. The treatment suggested by the MDL principle is to divide the complexity cost by the total number of samples generated so far. However we simplified matters by setting S to a large constant that roughly matches the number of samples we expect to see during training.

We used a diagonal Gaussian for both P and Q, allowing us to determine the complexity cost analytically:

$$KL(P_\phi \,\|\, Q_\psi) = \frac{(\mu_\phi - \mu_\psi)^2 + \sigma_\phi^2 - \sigma_\psi^2}{2\sigma_\psi^2} + \ln\left(\frac{\sigma_\psi}{\sigma_\phi}\right),$$

where μ_ϕ, σ_ϕ^2 and μ_ψ, σ_ψ^2 are the mean and variance vectors for P_ϕ and Q_ψ respectively. We adapted ψ with gradient descent along with ϕ, and the gradient of $\mathbb{E}_{\theta \sim P_\phi} L(\mathbf{x}, \theta)$ with respect to ϕ was estimated using the reparameterisation trick[2] (Kingma and Welling, 2013) with a single Monte-Carlo sample. The SoftPlus function $y = \ln(1 + e^x)$ was used to ensure that the variances were positive (Blundell et al., 2015).

Variational complexity gain (VCG). The increase of model complexity induced by a training example can be estimated from the change in complexity following a single parameter update from ϕ to ϕ' and ψ to ψ', yielding

$$\nu_{VCG} := KL(P_{\phi'} \,\|\, Q_{\psi'}) - KL(P_\phi \,\|\, Q_\psi)$$

Gradient variational complexity gain (GVCG). As with prediction gain, we can derive a first order Taylor ap-

[1] MDL deals with *sets* rather than *distributions*; in this context we consider each $\mathcal{D}_k$ in the curriculum to be a dataset sampled from the task distribution, rather than the distribution itself

[2] The reparameterisation trick yields a better gradient estimator for the posterior variance than that used in (Graves, 2011), which requires either calculation of the diagonal of the Hessian, or a biased approximation using the empirical Fisher. The gradient estimator for the posterior mean is the same in both cases.

proximation using the direction of gradient descent:

$$KL(P_{\phi'} \,\|\, Q_{\psi'}) \approx KL(P_\phi \,\|\, Q_\psi) - [\nabla_{\phi,\psi} KL(P_\phi \,\|\, Q_\psi)]^\top \nabla_{\psi,\phi} \mathcal{L}_{MDL}(\mathbf{x}, \phi, \psi)$$
$$\implies \nu_{VCG} \approx C - [\nabla_{\phi,\psi} KL(P_\phi \,\|\, Q_\psi)]^\top \nabla_\phi \mathop{\mathbb{E}}_{\theta \sim P_\phi} L(\mathbf{x}, \theta),$$

where C is a term that does not depend on $\mathbf{x}$ and is therefore irrelevant to the gain signal. We define the gradient variational complexity gain as

$$\nu_{GVCG} := [\nabla_{\phi,\psi} KL(P_\phi \,\|\, Q_\psi)]^\top \nabla_\phi \mathop{\mathbb{E}}_{\theta \sim P_\phi} L(\mathbf{x}, \theta),$$

which is the directional derivative of the KL along the gradient descent direction. We believe that the linear approximation is more reliable here than for prediction gain, as the model complexity has less curvature than the loss surface.

Relationship to VIME. Variational Information Maximizing Exploration (VIME) (Houthooft et al., 2016), uses a reward signal that is closely related to variational complexity gain. The difference is that while VIME measures the KL between the posterior before and after a step in parameter space, we consider the change in KL between the posterior and prior induced by the step. Therefore, while VIME looks for any change to the posterior, we focus only on changes that alter the divergence from the prior. Further research will be needed to assess the relative merits of the two signals.

L2 gain (L2G). Variational inference tends to slow down learning, making it appealing to define a complexity-based progress signal applicable to more conventionally trained networks. Many of the standard neural network regularisation terms, such as Lp-norms, can be viewed as defining an upper bound on model description length (Graves, 2011). We therefore hypothesize that the increase in regularisation cost will be indicative of the increase in model complexity. To test this hypothesis we consider training with a standard L2 regularisation term added to the loss:

$$L_{L2}(\mathbf{x}, \theta) = L(\mathbf{x}, \theta) + \frac{\alpha}{2} \|\theta\|_2^2 \tag{4}$$

where α is an empirically chosen constant. In this case the complexity gain can be defined as

$$\nu_{L2G} := \|\theta'\|_2^2 - \|\theta\|_2^2 \tag{5}$$

where we have dropped the $\alpha/2$ term as the gain will anyway be rescaled to $[-1, 1]$ before use. The corresponding first-order approximation is

$$\nu_{GL2G} := [\theta]^\top \nabla_\theta L(\mathbf{x}, \theta) \tag{6}$$

It is possible to calculate L2 gain for unregularized networks; however we found this an unreliable signal, presumably because the network has no incentive to decrease complexity when faced with uninformative data.

3.3. Prediction Gain Bias

Prediction gain, self prediction gain and gradient prediction gain are all closely related, but incur varying degrees of bias and variance. We now present a formal analysis of the biases present in each, noting that a similar treatment can be applied to our complexity gains.

We assume that the loss L is locally well-approximated by its first-order Taylor expansion:

$$L(\mathbf{x}, \theta') \approx L(\mathbf{x}, \theta) + \nabla L(\mathbf{x}, \theta)^\top \Delta\theta \tag{7}$$

where $\Delta\theta := \theta' - \theta$. For ease of exposition, we also suppose the network is trained with stochastic gradient descent (the same argument leads to similar conclusions when consider higher-order optimization methods):

$$\Delta\theta := -\alpha \nabla L(\mathbf{x}, \theta). \tag{8}$$

We define the true expected learning progress as

$$\nu := \mathop{\mathbb{E}}_{\mathbf{x}' \sim D} [\mathcal{L}(\theta) - \mathcal{L}(\theta')] = \alpha \Big\| \mathop{\mathbb{E}}_{\mathbf{x}' \sim D} \nabla L(\mathbf{x}, \theta) \Big\|^2,$$

with the identity following from (8) (recall that $\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{x}} L(\theta)$). The expected prediction gain is then

$$\nu_{\text{PG}} = \mathop{\mathbb{E}}_{\mathbf{x}' \sim D} [L(\mathbf{x}, \theta) - L(\mathbf{x}, \theta')] = \alpha \mathop{\mathbb{E}}_{\mathbf{x}' \sim D} \big\| \nabla L(\mathbf{x}, \theta) \big\|^2.$$

Defining

$$\mathbb{V}\big(\nabla L(\mathbf{x}, \theta)\big) := \mathbb{E} \big\| \nabla L(\mathbf{x}, \theta) - \mathbb{E} \nabla L(\mathbf{x}', \theta) \|^2,$$

we find that prediction gain is the sum of two terms: true expected learning progress, plus the gradient variance:

$$\nu_{\text{PG}} = \nu + \mathbb{V}\big(\nabla L(\mathbf{x}, \theta)\big).$$

We conclude that *for equal learning progress, a prediction gain-based curriculum maximizes variance*. The problem is made worse when using gradient prediction gain, which actually relies on the Taylor approximation (7). On the other hand, self prediction gain is an unbiased estimate of expected learning progress:

$$\mathop{\mathbb{E}}_{\mathbf{x}} \nu_{\text{SPG}} = \mathop{\mathbb{E}}_{\mathbf{x}, \mathbf{x}' \sim D} [L(\mathbf{x}', \theta) - L(\mathbf{x}', \theta')] = \nu.$$

Naturally, its use of two samples results in higher variance than prediction gain, suggesting a bias-variance trade off between the two estimates.

4. Experiments

To test the efficacy of our approach, we applied all the gains defined in the previous section to three task suites: synthetic language modelling on text generated by n-gram

models, repeat copy (Graves et al., 2014) and the bAbI tasks (Weston et al., 2015)

The network architecture was stacked unidirectional LSTM (Graves, 2013) for all experiments, and the training loss was cross-entropy with either categorical targets and softmax output, or Bernoulli targets and sigmoid outputs, optimised by RMSProp with momentum (Tieleman, 2012; Graves, 2013), using a momentum of 0.9 and a learning rate of 10^{-5} unless specified otherwise. The parameters for the Exp3.S algorithm were $\eta = 10^{-3}$, $\beta = 0$, $\epsilon = 0.05$. For all experiments, one set of networks was trained with variational inference (VI) to test the variational complexity gain signals, and another set was trained with normal maximum likelihood (ML) for the other signals; both sets were repeated 10 times with different random seeds to initialise the network weights. The α regularisation parameter from Eq. (4) for the networks trained with L2 gain signals was 10^{-4} for all experiments. For all plots with a time axis, time is defined as the total number of input steps processed so far. In the absence of hand-designed curricula for these tasks, our performance benchmarks are 1) a fixed uniform policy over all the tasks and 2) directly training on the target task (where applicable). All losses and error rates are measured on independent samples not used for training or reward calculation.

4.1. N-Gram Language Modelling

Our first experiment aims to illustrate and compare the behaviour induced by different gains. We trained character-level Kneser-Ney n-gram models (Kneser and Ney, 1995) on the King James Bible data from the Canterbury corpus (Arnold and Bell, 1997), with the maximum depth parameter n ranging between 0 to 10. We then used each model to generate a separate dataset of 1M characters, which we divided into disjoint sequences of 150 characters. The first 50 characters of each sequence were used as burn-in context for the next 100, which the network was trained to predict. The LSTM network had two layers of 512 cells, and the batch size was 32.

An important characteristic of this dataset is that the amount of linguistic structure increases monotonically with n. Simultaneously, the entropy – and hence, minimum achievable loss – decreases almost monotonically in n. If we believe that learning progress should be higher for interesting data than for data that is difficult to predict, we would expect the gain signals to be drawn to higher n: they should favour structure over noise. We note that in this experiment the curriculum is superfluous: the most efficient strategy for learning the 10-gram source is to directly train on it.

Fig. 1 shows that most of the complexity-based gain signals from Section 3.2 (L2G, GL2G, GVCG) progress rapidly

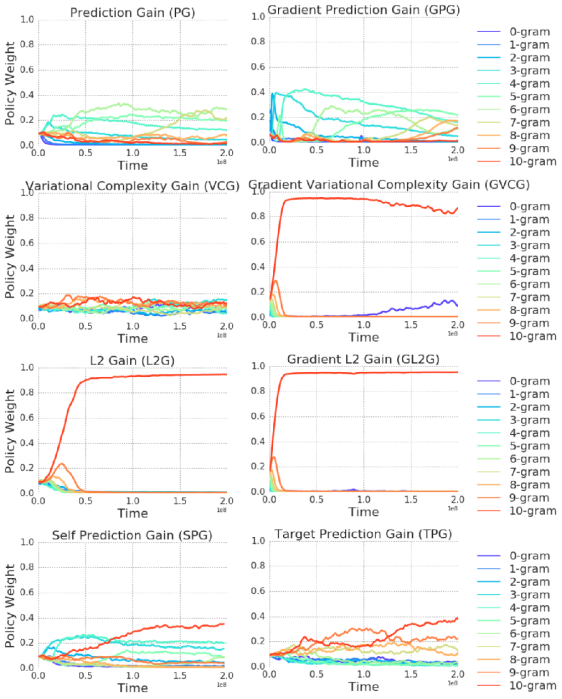

Figure 1. N-gram policies for different gain signals, truncated at 2×10^8 steps. All curves are averages over 10 runs

through the curriculum before focusing strongly on the 10-gram task (though interestingly, GVCG appears to revisit 0-gram later on in training). The clarity of the result is striking, given that sequences generated from models beyond about 6-gram are difficult to distinguish by eye. VCG follows a similar path, but with much less confidence, presumably due to the increased noise. The loss-based measures (PG, GPG, SPG, TG) also tend to move towards higher n, although more slowly and with less certainty. Unlike the complexity gains, they tend to initially favour the lower-n tasks, which may be desirable as we would expect early learning to be more efficient with simpler data.

4.2. Repeat Copy

In the repeat copy task (Graves et al., 2014) the network receives an input sequence of random bit vectors, and is then asked to output that sequence a given number of times. The task has two main dimensions of difficulty: the length of the input sequence and the required number of repeats, both of which increase the demand on the models memory. Neural Turing machines are able to learn a 'for-loop' like algorithm on simple examples that can directly generalise to much harder examples (Graves et al., 2014). For LSTM networks without access to external memory, however, sig-

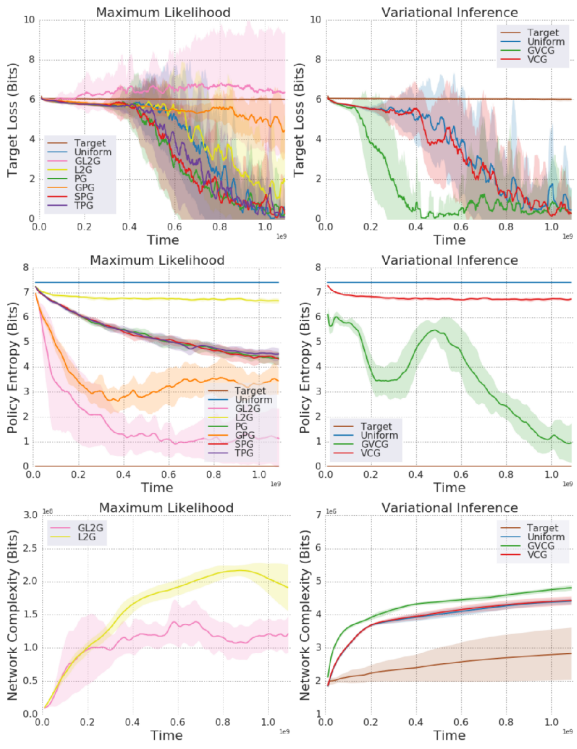

Figure 2. Target task loss (per output), policy entropy and network complexity for the repeat copy task, truncated at 1.1×10^9 steps. Curves are averages over 10 runs, shaded areas show the standard deviation. Network complexity was computed by multiplying the per-sample complexity cost by the total size of the training set.

nificant retraining is required to adapt to harder tasks.

We devised a curriculum with both the sequence length and the number of repeats varying from 1 to 13, giving 169 tasks in all, with length 13, repeats 13 defined as the target task. The LSTM network had a single layer of 512 cells, and the batch size was 32. As the data was generated online, the number of samples S in Eq. (3) (the per-sample VI loss) was undefined; we arbitrarily set it to 169M (1M per task in the curriculum).

Fig. 2 shows that GVCG solves the target task about twice as fast as uniform sampling for VI training, and that the PG, SPG and TPG gains are somewhat faster than uniform for ML training, especially in the early stages. From the entropy plots it is clear that these signals all lead to strongly non-uniform policies. The VI complexity curves also demonstrate that GVCG yields significantly higher network complexity than uniform sampling, supporting our hypothesis that increased complexity correlates with learning progress. Unlike GVCG, the VCG signal did not deviate far from a uniform policy,. L2G and particularly GPG and GL2G were much worse than uniform, suggesting that (1) the bias induced by the gradient approximation has a pernicious effect on learning and (2) that the increase in L2 norm is not a reliable measure of increased network complexity. Training directly on the target task failed to learn at all, underlining the necessity of curriculum learning for this problem.

Fig. 3 reveals a consistent strategy for the GVCG syllabuses, first focusing on short sequences with high repeats, then long sequences with low repeats, thereby decoupling the two dimensions of difficulty. At each stage the loss is substantially reduced across many tasks that the policy does not focus on. Crucially, this means that the network does not have to visit each of the 169 tasks to solve them all, and the syllabus is able to exploit this fact to more efficiently complete the curriculum.

4.3. Babi

The bAbI dataset (Weston et al., 2015) consists of 20 synthetic question-answering problems designed to probe the basic reasoning capabilities of machine learning models. Although bAbI was not specifically designed for curriculum learning, some of the tasks follow a natural ordering of complexity (e.g. 'Two Arg Relations', 'Three Arg Relations') and all are based on a consistent probabilistic grammar, leading us to hope that an efficient syllabus could be found for learning the whole set. The usual performance measure for bAbI is the number of tasks 'completed' by the model, where completion is defined as getting less than 5% of the test set questions wrong.

The data representation followed (Graves et al., 2016), with each word in the observation and target sequences represented as a 1-hot vector, along with an extra binary channel to mark answer prompts. The original datasets were small, with either 1K or 10K questions per task, so as to test generalisation from limited samples. However LSTM is known to perform poorly in this setting (Sukhbaatar et al., 2015; Graves et al., 2016), and we wished to avoid the confounding effect of overfitting on curriculum learning. We therefore used the bAbI code (Weston et al., 2015) to generate 1M stories (each containing one or more questions) for each of the 20 tasks. With so many examples, we found that training and evaluation set performance were indistinguishable, and therefore report training performance only. The LSTM network had two layer of 512 cells, the batch size was 16, and the RMSProp learning rate was 3×10^{-5}.

Fig. 4 shows that prediction gain (PG) clearly improved on uniform sampling in terms of both learning speed and number of tasks completed; for self-prediction gain (SPG) the same benefits were visible, though less pronounced. The other gains were either roughly equal to or worse than uniform. For variational inference training, GVCG was faster

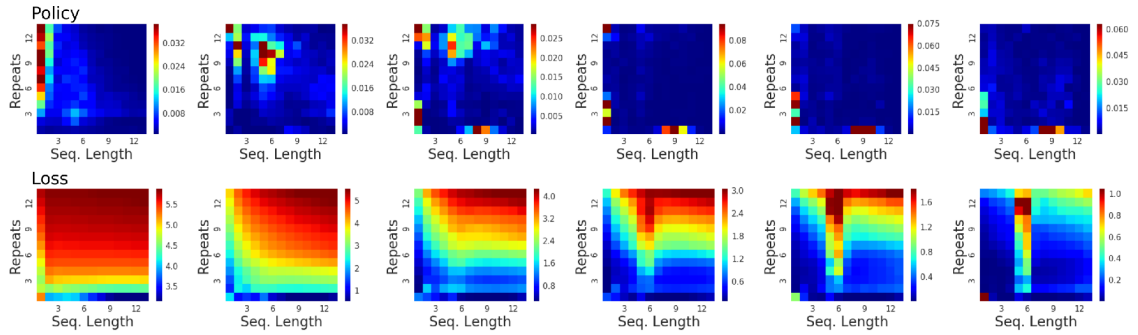

Figure 3. Average policy and loss per output over time for GVCG networks on the repeat copy task. Plots were made by dividing the first 4×10^8 steps into five equal bins, then averaging over the policies of all 10 networks over all times within each bin.

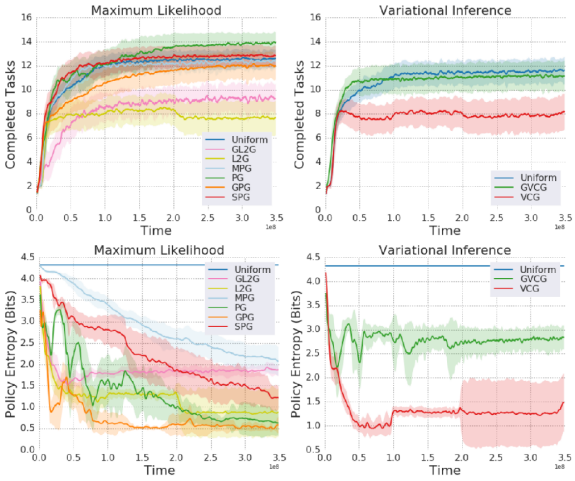

Figure 4. Completion and entropy curves for the bAbI curriculum, truncated at 3.5×10^8 steps. Curves are means over ten runs, shaded areas show standard deviation.

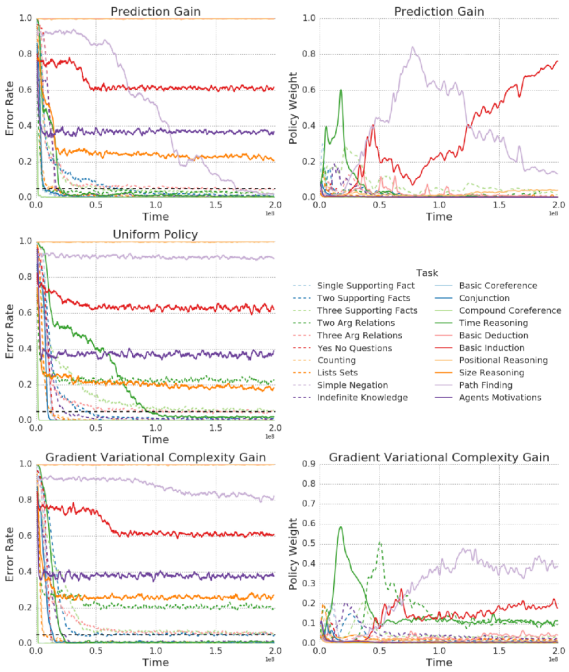

Figure 5. Per-task policy and error curves for bAbI, truncated at 2×10^8 steps. All plots are averaged over 10 runs. Black dashed lines show the 5% error threshold for task completion.

than uniform at first, then slightly worse later on, while VCG performed poorly for reasons that are unclear to us. In general, training with variational inference appeared to hamper progress on the bAbI tasks.

Fig. 5 shows how the PG and GVCG syllabuses accelerate the network's progress by selectively focusing on specific tasks until completion. For example, they both solve 'Time Reasoning' much faster than uniform sampling by concentrating on it early in training; similarly, PG focuses strongly on 'Path Finding' (one of the harder bAbI tasks) until it solves it. Also noteworthy is the way the syllabuses progress from 'Single Supporting Fact' to 'Three Supporting Facts' in order; this shows that our gain signals can discover implicit orderings, and hence opportunities for efficient transfer, in an unsorted curriculum.

5. Conclusion

Our experiments suggest that using a stochastic syllabus to maximise learning progress can lead to significant gains in curriculum learning efficiency, so long as a suitable progress signal is used. We note however that uniformly sampling from all tasks is a surprisingly strong benchmark. We speculate that this is because learning is dominated by gradients from the tasks on which the network is making

fastest progress, inducing a kind of implicit curriculum, albeit with the inefficiency of unnecessary samples. For maximum likelihood training, we found prediction gain to be the most consistent signal, while for variational inference training, gradient variational complexity gain performed best. Importantly, both are instantaneous, in the sense that they can be evaluated using only the samples used for training. As well as being more efficient, this has broader applicability to tasks where external evaluation is difficult, and suggests that learning progress is best assessed on a local, rather than global basis.

References

Arnold, R. and Bell, T. (1997). A corpus for the evaluation of lossless compression algorithms. In *Data Compression Conference, 1997. DCC'97. Proceedings*, pages 201–210. IEEE.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77.

Barto, A. G. (2013). Intrinsic motivation and reinforcement learning. In *Intrinsically Motivated Learning in Natural and Artificial Systems*, pages 17–47. Springer.

Bellemare, M. G., Srinivasan, S., Ostrovski, G., Schaul, T., Saxton, D., and Munos, R. (2016). Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*.

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48, New York, NY, USA. ACM.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1613–1622.

Bubeck, S. and Cesa-Bianchi, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning*, 5(1):1–122.

Elman, J. L. (1993). Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Graves, A. (2011). Practical variational inference for neural networks. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P. L., Pereira, F., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 24*, pages 2348–2356. Curran Associates, Inc.

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *arXiv preprint arXiv:1410.5401*.

Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwińska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476.

Grünwald, P. D. (2007). *The minimum description length principle*. The MIT Press.

Herbster, M. and Warmuth, M. K. (1998). Tracking the best expert. *Machine Learning*, 32(2):151–178.

Hinton, G. E. and Van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13. ACM.

Houthooft, R., Chen, X., Duan, Y., Schulman, J., De Turck, F., and Abbeel, P. (2016). Vime: Variational information maximizing exploration. In *Advances In Neural Information Processing Systems*, pages 1109–1117.

Itti, L. and Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision research*, 49(10):1295–1306.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 181–184, Detroit, Michigan, USA.

Lopes, M. and Oudeyer, P.-Y. (2012). The strategic student approach for life-long exploration and learning. In *IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*.

Oudeyer, P., Kaplan, F., and Hafner, V. (2007). Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286.

Reed, S. and de Freitas, N. (2015). Neural programmer-interpreters. *arXiv preprint arXiv:1511.06279*.

Rissanen, J. (1986). Stochastic complexity and modeling. *Ann. Statist.*, 14(3):1080–1100.

Roy, N. and Mccallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *In Proc. 18th International Conf. on Machine Learning*.

Schmidhuber, J. (1991). A possibility for implementing curiosity and boredom in model-building neural controllers. In *From animals to animats: proceedings of the first international conference on simulation of adaptive behavior*.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Settles, B., Craven, M., and Ray, S. (2008). Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296.

Storck, J., Hochreiter, J., and Schmidhuber, J. (1995). Reinforcement driven information acquisition in non-deterministic environments. In *Proceedings of the International Conference on Artificial Neural Networks, vol. 2*.

Sukhbaatar, S., Weston, J., Fergus, R., et al. (2015). End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Sutskever, I. and Zaremba, W. (2014). Learning to execute. *arXiv preprint arXiv:1410.4615*.

Tieleman, T., H. G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*.

Tsvetkov, Yulia, F. M. L. W. M. B. D. C. (2016). Learning the curriculum with bayesian optimization for task-specific word representation learning. *arXiv preprint arXiv:1605.03852*.

Weston, J., Bordes, A., Chopra, S., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.

外文翻译：神经网络自动课程学习

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, Koray Kavukcuoglu

摘要

我们介绍一种自动选择优化路径或者教学大纲的方法使得神经网络可以通过课程最大化学习效率。将网络从每个数据样本学习的量的度量作为奖励信号提供给非稳态多臂老虎机算法，其然后确定随机教学大纲。我们考虑从学习进度的两个不同指标得出的一系列信号：预测精度的增加率和网络复杂度的增加率。在三个课程的LSTM网络的实验结果表明我们的方法可以显着加速学习，在某些情况下可以将网络达到满意表现水平所需的训练时间缩短一半。

1 介绍

二十多年前，在从少到多的重要性文章中，Elman提出了逐渐增加难度的课程可能可以显著的加快的神经网络训练的想法(Elman,1993)。然而课程学习仅仅最近才在这个领域变得流行起来(例如，Bengio等人2009的工作)部分原因是更加复杂的问题被考虑了。特别地，最近的神经网络的学习程序依赖于课程去扩展到更加长时间和更加复杂的程序（(Sutskever和Zaremba, 2014; Reed 和de Freitas, 2015; Graves等人, 2016）。我们期待这个趋势会进一步随着神经网络的规模的庞大化而进一步延续。

课程学习很缓慢的被采用的一个原因是它的有效性对于任务之间学习过程的模式高度敏感。一个流行的方法是对于进一步学习的新任务去手工

设定一个性能的阈值，伴随着一个固定的概率调整到简单的任务以防止遗忘。然而，又因为这引入了十分困难调节的参数，对于课程来说这使得可靠的阈值是一个未知的变量或者随着任务发生改变的变量。更加基本地，这预设了任务可以依据难度来被排序，然而在现实中任务可能在不同的坐标轴上难度不一样，或者根本没有预设的顺序。

我们提出与其把针对哪一个任务被进一步学习做决定视作一个随机的政策，不如连续的去适应地优化Oudeyer等人(2007) 命名的学习进程的概念。这样做使我们联系上一些探讨本质的动机的文献（Barto，2013），其中不一样的学习进程当中的指示因子被用作奖励信号来激励探索，这包括压缩过程(Schmidhuber, 1991)，信息获取(Storck 等人, 1995)，贝叶斯惊奇(Itti和Baldi,2009)，预测量增加(Bellemare 等人, 2016)和变分信息最大化(Houthooft et al., 2016)。我们聚焦在预测量增加的一些变种方法，同时也引入一些新奇的进度信号类型，我们把它称作复杂度增加。通过最小化描述长度原则，复杂度增益等价于伴随着被编码到网络权重中的有效信息的增加所对应的知识的获取。

给出可以针对每个训练示例进行评估的进度信号，我们使用多臂老虎机算法来找到最大化整体进度的任务的随机策略。由于网络的行为，赌博尝试是不稳定的，因此在训练过程中最优策略也会发生变化。我们从以前的工作(Lopes 和Oudeyer, 2012)中得到灵感，在以发展学习为背景的情况下，我们将自适应学生以多臂老虎机任务设定建模。另一个相关领域是主动学习领域，其中已经使用类似的增益信号来指导下一个标记哪些数据点的决定(Settles, 2010)。最后，最近有关使用贝叶斯优化找到在语言语料库上训练单词嵌入网络的最佳顺序的工作(Tsvetkov, 2016)和我们的工作有些类似；然而，这与我们的工作不同，因为在每次训练之前，排序是完全确定的，而不是根据模型的进度而自适应地改变。

2 背景

我们考虑监督或无监督的学习问题，其中目标序列$\boldsymbol{b}^1, \boldsymbol{b}^2, ...$在给定各自的输入序列$\boldsymbol{a}^1, \boldsymbol{a}^2, ...$的条件下建模。为方便起见，我们假设目标是来自有限的集合$\mathcal{B}$，注意到我们的框架可以延伸到持续的目标，只需用密度函数取代概率分布。如通常的神经网络，序列可以分批分组在一起$(\boldsymbol{b}^{1:B}, \boldsymbol{a}^{1:B})$,以加速训练。模型的条件概率是

$$p(\boldsymbol{b}^{1:B}, \boldsymbol{a}^{1:B}) = \prod_{i,j} p(\boldsymbol{b}^i_j | \boldsymbol{b}^i_{1:j-1}, \boldsymbol{a}^i_{1:j-1}).$$

从这里开始，我们考虑每一批作为$\mathcal{X} := (\mathcal{A} \times \mathcal{B})^N$的一个例子$\boldsymbol{x}$, 并且记$p(\boldsymbol{x}) := p(\boldsymbol{b}^{1:B}|\boldsymbol{a}^{1:B})$为这个的概率。在这个标记下，一个任务是在$\mathcal{X}$上序列上的一本分布。一个课程是这些任务的集成$D_1, \cdots, D_N$,并且其中一个样本是从课程中的一个任务的采样得到的样例。最后，一个教案是一个任务们上的随时间变化的分布序列。

我考虑神经网为一个$\mathcal{X}$上参数化概率模型p_θ,它的参数被记为θ。第k^{th}任务上神经网络的期望损失是

$$\mathcal{L}_k(\theta) := \mathop{\mathbb{E}}_{\boldsymbol{x}\sim D_k} L(\boldsymbol{x}, \theta),$$

其中$L(\boldsymbol{x}, \theta) := -\log p_\theta(\boldsymbol{x})$是一个$\boldsymbol{x}$上的采样损失。当没有歧义的情况下，我们将会简单的分别记期望和采样损失为$\mathcal{L}_k$和$L(\boldsymbol{x})$。

2.1 课程学习

我们考虑两个关联起来的设定。在多任务的设置中，目标是在所有的任务的集成$\{D_k\}$上尽可能都表现的好；这个被有目标函数

$$\mathcal{L}_{MT} := \frac{1}{N}\sum_{k=1}^{N} \mathcal{L}_{k,}$$

在目标任务的设置中，我们仅仅对于对于最小化最后任务D_N的损失。一些任务然后扮演了一系列关于真实问题的跳板。这个最终的目标函数可以被简单定义为

$$\mathcal{L}_{TT} := \mathcal{L}_{N.}$$

2.2 对抗多臂老虎机

我们把一个包含的N个任务的课程视做一个多臂老虎机(Bubeck and Cesa-Bianchi, 2012)，我们把一个教学大纲视做一个自适应的政策来寻找这个赌博问题的最大回报。在这个赌博问题的设置中，一个代理选择一系列臂（行动）$a_1 \cdots a_T$在T次游戏中。在每一轮后，被选择的臂会产生一个回报r_t；而其他臂的回报将不会被观测到。

经典的对抗老虎机的算法是Exp3（Auer et al., 2002），他们利用了乘性的权重更新来保证对于最优的臂的遗憾程度尽可能的低。在t轮，代理依

据政策π_t随机的选择一个臂。这个政策由一个集合的权重$w_{t,i}$所定义:

$$\pi_t^{EXP3}(i) := \frac{e^{w_{t,i}}}{\sum_{j=1}^{N} e^{w_{t,j}}}$$

这些权重的是重要性采样奖励的加和:

$$w_{t,i} := \eta \sum_{s<t} \tilde{r}_{s,i}$$

$$\tilde{r}_{s,i} := \frac{r_s \mathbb{I}_{a_s=i}}{\pi_s(i)}$$

Exp3 这样做意在最小化关于由历史计算单个最优的臂的遗憾程度。然而，一个普遍的遭遇是一个臂只会在历史上一段比例上成为最优的，然后是下一个臂，然后这个过程继续；最优的策略变成逐段固定的。这是我们讨论情况的设定，对于任务的期望奖励会随着模型学习而发生改变。固定分享方法(Herbster and Warmuth, 1998)通过利用ϵ贪婪策略和加性权重的混合指出了这个问题。在赌博的设定中，这由$Exp3.S$算法(Auer et al. (2002))而出名,

$$\pi_t^{EXP3,S}(i) := (1-\epsilon)\pi_t^{EXP3}(i) + \frac{\epsilon}{N}$$

$$w_{t,i}^S := \log[(1-\alpha_t)e^{w_{t-1,i}^S + \eta\tilde{r}_{t-1,i}^{\beta}} + \frac{\alpha_t}{N-1}\sum_{j\neq i} e^{w_{t-1,j}^S + \eta\tilde{r}_{t-1,j}^{\beta}}]$$

$w_{1,i}^S := 0, \alpha_t := t^{-1}, \tilde{r}_{s,i}^{\beta} := \frac{r_s \mathbb{I}_{a_s=i}+\beta}{\pi_s(i)}$

2.3 奖励比例

适当的步长 η 取决于回报的大小，它可能不会被称为先验。在我们的设置中问题是特别严重的，其中的大小取决于学习进展被如何的测量并且随着机器的学习在它不同时间不一样。为了解决这个问题，我们自适应的重新调整所有的奖励使其落在$[-1,1]$的区间上：让$\mathcal{R}_t$未被调整的奖励函数到t时刻截止的历史，即$\mathcal{R}_t = \{\hat{r}_i\}_{i=1}^{t-1}$。让$q_t^{lo}$和$q_t^{hi}$为$\mathcal{R}_t$的分位数，我们选择其分别为$20^{th}$和$80^{th}$百分位数。被缩放的奖励$r_t$由$\hat{r}_t$在$[q_t^{lo}, q_t^{hi}]$截断所得到然后将其线性映射到$[-1,1]$:

$$r_t = \begin{cases} -1 & \text{if } \hat{r}_t < q_t^{lo} \\ 1 & \text{if } \hat{r}_t > q_t^{hi} \\ \dfrac{2(\hat{r}_t - q_t^{lo})}{q_t^{hi} - q_t^{lo}} - 1 & otherwise. \end{cases} \tag{1}$$

相比于保留所有历史的奖励，我们利用水库抽样的方法去维持又表示性的样本，并计算这个样本估计的分位数。这个分位数可以利用$\Theta(\log|\mathcal{R}_t|)$的时间来获得。

3 学习过程信号

我们的目标是去利用Exp3.S输出的政策当做一个教学大纲来训练我们的模型。理想地，我们想用政策去最大化我们能最小化损失的比例，并且奖励函数应该能够反映这个比例，Oudeyer等人把这个称作学习进程。然而，这经常是计算上不被想要的，或者甚至不可能去度量训练样本对于目标函数的影响，并且我们因此转而去替代进程的度量。广义上说，这些度量要么1）基于损失的，意思是他们讲奖励和损失的下降等价起来；2）复杂度驱动，当他们将模型复杂度和奖励等同起来。

训练的过程这样进行：在每一个时间t,我们先对任务的下标进行一个采样$k \sim \pi_t$。我们然后生成一个这个任务的采样，即$\boldsymbol{x} \sim D_k$。注意到每一个$\boldsymbol{x}$是一族训练序列，并且为了去降低获得的信号的噪声，我们抽取这个任务当中的整个完整的族。我们计算所选定的学习进程的度量v然后除以被处理样本所需要的时间$\tau(\boldsymbol{x})$（因为这是我们关心的学习进程的比例，并且处理时间可能依据任务的不同而改变）来获得原始的奖励$\hat{r} = v/\tau(\boldsymbol{x})$。由于任务的目的，$\tau(\boldsymbol{x})$仅是$\boldsymbol{x}$最长的输入序列的长度;对于其他的任务和结构一个更加复杂的计算可能会被需要。我们然后调整$\hat{r}$成为一个奖励函数$r_t \in [-1, 1]$,并将其交给Exp3.S。这个过程由算法1总结。

3.1 损失驱动的进程

我们考虑5个损失驱动的信号，当中所有的都比较了在某个样本$\boldsymbol{x}$上模型训练前后的预测。前两个信号，我们展示的是及时的信号，其只依赖于$\boldsymbol{x}$。这样的信号是十分具有吸引力的，由于他们普遍计算更加廉价，并且是对于整个全局的课程是不可知的。剩下的三个信号更加直接的度量了训练在被期望的目标函数上的效果，但需要额外的采样$\boldsymbol{x}'$。在接下来的部分我们记在$\boldsymbol{x}$上训练前后的模型参数分布为θ和θ'。

预测增加（PG）预测增益被定义为x上训练前后样本$\boldsymbol{x}$的瞬时损失变

Algorithm 1 内在动机课程学习

Initially:

$w_i = 0$ for $i \in [N]$

for $t = 1 \dots T$ **do**

$\pi(k) := (1-\epsilon)\frac{e^{w_k}}{\sum_i e^{w_i}} + \frac{\epsilon}{N}$

从π中采样得到下标k

从D_k中采样得到$\boldsymbol{x}$

从$\boldsymbol{x}$上训练模型p_θ

计算训练进程v(章节3.1&3.2)

映射$\hat{r} = v/\tau(\boldsymbol{x})$于$r \in [-1, 1]$中（章节2.3）

利用奖励r和Exp3.S更新w_i

end for

化**:**

$$v_{PG} := L(\boldsymbol{x}, \theta) - L(\boldsymbol{x}, \theta'),$$

当p_θ是一个贝叶斯混合模型，预测增量是模型信息增量的上界，并且因此更加贴近于贝叶斯的概念，学习是后验的改变。

梯度预测增益（GPG）。计算预测增益需要额外的正向传播。当p是可微分的时候，替代方案是考虑预测增益的一阶泰勒级数逼近**:**

$$L(\boldsymbol{x}, \theta') \approx L(\boldsymbol{x}, \theta) + [\nabla_\theta L(\boldsymbol{x}, \theta)]^T \Delta_\theta$$

其中Δ_θ是下降步长。将此步骤作为负梯度$-\nabla_\theta L(\boldsymbol{x}, \theta)$，得到梯度预测增益

$$v_{GPG} := \|\nabla_\theta L(\boldsymbol{x}, \theta)\|_2^2$$

它衡量了梯度矢量的大小，并已经在主动学习文献中作为数据显着性的指标（Settles 等人，2008）被使用。我们将在下面显示梯度预测增益是真正的预期学习进度一个有偏差的估计值，特别是偏好于损失方差较大的任务。

自我预测增益（SPG）。预测增益是$\mathcal{L}_k(\theta)$改变的有偏估计，即任务k的预期损失。对$\boldsymbol{x}$进行训练后，我们自然会期望样本损失$L(\boldsymbol{x}, \theta)$减小，即使其他点的损失可能会增加。自我预测增益通过从同一任务中抽取第二

次并估计新样本的进度来解决这个问题：

$$v_{SPG} := L(\boldsymbol{x}', \theta) - L(\boldsymbol{x}', \theta') \quad \boldsymbol{x}' \sim D_{k.}$$

目标预测增益（TPG）。我们可以进一步采取自我预测增益思想，直接评估感兴趣的损失，这也在主动学习中被考虑的（Roy和Mccallum，2001）。在目标任务设置中，这将成为

$$v_{TPG} := L(\boldsymbol{x}', \theta) - L(\boldsymbol{x}', \theta') \quad \boldsymbol{x}' \sim D_{N.}$$

尽管这可能看起来是迄今为止最准确的措施，但它往往会受到高度的方差的影响，而且还会与之前的假设相矛盾，在训练的前期，模型不会在困难的目标任务上得到改善并且应该转而在一个它可以驾驭的任务上训练。

平均预测增益（MPG）。平均预测增益与在多个任务设置中的目标预测增益类似，其中评估我们在所有任务中的进展是自然的。我们记

$$v_{MPG} := L(\boldsymbol{x}', \theta) - L(\boldsymbol{x}', \theta') \; \boldsymbol{x}' \sim D_k, k \sim U_{N.}$$

其中U_N表示1...N上的均匀分布。平均预测增益有额外的来自于取样评估任务$k \sim U_N$的方差。

3.2 复杂度驱动进程

到目前为止，我们已经通过观察其预测能力的变化率，考虑了直接衡量网络学习进度的收益。我们现在转向一套新的收益，而不是衡量网络复杂度增加的速度。这些收益是由最小描述长度（MDL）原则（Rissanen，1986;Grünwald，2007）支持的：为了从特定数据集中最佳地推广，应该最小化描述模型参数所需的位数加上模型描述数据所需的位数。

根据MDL原则，将模型复杂度提高一定量只有在数据压缩量较大时才是值得的。因此，我们期望复杂性最大程度地依赖于于网络最能推广的训练样例而增长。这些例子正是我们在尝试最大化学习进度时寻求的。

神经网络的MDL训练（Hinton和Van Camp，1993）可以通过随机变分推理实际实现（Graves，2011; Kingma等，2015; Blundell等，2015）。在这个框架下，对于每个训练样本，单个权重样本被采样，以在训练期间网络权重的变分后验$P_\phi(\theta)$得以被维护。后验参数ϕ优化，而不是θ本身。总损失是训练数据集的期望log损失（在我们的情况下是完整的课程），加

上后验和一些固定（Blundell et al。，2015）或自适应（Graves，2011）先验$Q_\psi(\theta)$:

$$L_{VI}(\phi,\psi) = KL(P_\phi||Q_\psi) + \sum_k \sum_{\boldsymbol{x}\in D_k} \mathop{\mathbb{E}}_{\theta\sim P_\phi} L(\boldsymbol{x},\theta).$$

由于我们使用随机梯度下降，我们需要确定模型复杂度和数据的每个样本损失。定义$S := \sum_k |D_k|$ 作为我们获得的课程中的样本总数，我们可以的得到

$$L_{VI}(\boldsymbol{x},\phi,\psi) = \frac{1}{S} KL(P_\phi||Q_\psi) + \mathop{\mathbb{E}}_{\theta\sim P_\phi} L(\boldsymbol{x},\theta),$$

那么

$$L_{VI}(\phi,\psi) = \sum_k \sum_{\boldsymbol{x}\in D_k} L_{VI}(\boldsymbol{x},\phi,\psi).$$

一些我们考虑的课程是利用算法生成的，意味着每个类别的采样数量是没有定义的。MDL原则提出的处理是将复杂度成本除以目前生成的样本总数。然而，我们通过将S设置为一个大的常数来简化事情，这个常数与我们期望在训练期间看到的样本数量大致相同。

我们使用P和Q的对角高斯，使我们能够分析地确定复杂性成本:

$$KL(P_\phi||Q_\psi) = \frac{(\mu_\phi-\mu_\psi)^2 + \sigma_\phi^2 - \sigma_\psi^2}{2\sigma_\psi^2} + \ln\frac{\sigma_\psi}{\sigma_\phi}$$

其中μ_ϕ, σ_ϕ^2和μ_ψ, σ_ψ^2分别为P_ϕ和Q_ψ的向量。我们适应ψ利用关于ϕ梯度下降法，而$\mathop{\mathbb{E}}_{\theta\sim P_\psi} L(\boldsymbol{x},\theta)$关于$\phi$梯度可以通过一次蒙特卡洛采用利用重新参数化的技巧(Kingma and Welling, 2013)得到。软性加和函数$y = \ln(1+e^x)$被使用来保证方差是正的。(Blundell 等人, 2015)

变分复杂度增益（VCG）。由训练样本引起的模型复杂度的增加可以单个参数ϕ到ϕ'和ψ到ψ'更新之后的复杂度的变化来估计，产生

$$v_{VCG} := KL(P'_\phi||Q_{\psi'}) - KL(P_\phi||Q_\psi)$$

梯度变分复杂度增益（GVCG）。与预测增益一样，我们可以使用梯度下降方向导出一阶泰勒近似:

$$KL(P'_\phi||Q_{\psi'}) \approx KL(P_\phi||Q_\psi) - [\nabla_{\phi,\psi} KL(P_\phi||Q_\psi)]^T \nabla_{\psi,\phi} \mathcal{L}_{MDL}(\boldsymbol{x},\phi,\psi)$$
$$\rightarrow v_{VCG} \approx C - [\nabla_{\phi,\psi} KL(P_\phi||Q_\psi)]^T \nabla_\phi \mathop{\mathbb{E}}_{\theta\sim P_\phi} L(\boldsymbol{x},\theta)$$

其中C是不依赖于$\boldsymbol{x}$的因此是与增益无关的信号。我们定义梯度变分复杂度增加为

$$v_{GVCG} := [\nabla_{\phi,\psi} KL(P_\phi||Q_\psi)]^T \nabla_\phi \mathop{\mathbb{E}}_{\theta\sim P_\phi} L(\boldsymbol{x},\theta),$$

这是KL沿梯度下降方向的方向导数。我们认为线性逼近在这里比预测增益更可靠，因为模型复杂度比损耗曲面具有更小的曲率。

与VIME的关系。最大化探索的变分信息（VIME）（Houthooft等，2016）使用与变分复杂度增益密切相关的奖励信号。不同之处在于，当VIME在参数空间中测量一步后后验之前与之后的KL散度，我们考虑由一步造成的后验和先验的KL散度的变化。因此，虽然VIME寻找到后方的任何变化，但我们只关注改变与之前的差异的变化。需要进一步的研究来评估两个信号的相对优点。

L2增益（L2G）。变分推理倾向于减慢学习速度，因此有助于定义适用于更传统训练有素的网络的基于复杂度的进度信号。许多标准神经网络正则化项，如L^p范数，可以被视为定义模型描述长度的上限（Graves，2011）。因此，我们假设正则化成本的增加将表明模型复杂性的增加。为了测试这个假设，我们认为训练与标准的L^2正则化项相加：

$$L_{L_2}(\boldsymbol{x},\theta) = L(\boldsymbol{x},\theta) + \frac{\alpha}{2}\|\theta\|_2^2$$

其中α是经验选择的常数。在这种情况下，复杂度增益可以被定义为

$$v_{L2G} := \|\theta'\|_2^2 - \|\theta\|_2^2$$

在那里我们放弃了$\alpha/2$项，因为增益是在使用前会重新缩放到$[-1,1]$。相应的一阶近似是

$$v_{GL2G} := [\theta]^T \nabla_\theta L(\boldsymbol{x},\theta),$$

可以计算不规则网络的L^2增益; 然而，我们发现这是一个不可靠的信号，可能是因为网络在面对无信息数据时没有激励降低复杂性。

3.3 预测增量偏差

预测增益，自我预测增益和梯度预测增益都是密切相关的，但会产生不同程度的偏差和方差。我们现在对每个偏见存在一个正式的分析，指出类似的处理可以应用于我们的复杂性增益。我们假设损失L在本地很好地接近于一阶泰勒展开：

$$L(\boldsymbol{x},\theta) \approx L(\boldsymbol{x},\theta) + \nabla L(\boldsymbol{x},\theta)^T \Delta\theta$$

其中

$$\Delta\theta = \theta' - \theta.$$

为了便于说明，我们还假定网络训练有随机梯度下降（考虑高阶优化方法时，同样的结论得出相似的结论）：

$$\Delta\theta := -\alpha \nabla L(\boldsymbol{x},\theta).$$

我们将真正的预期学习进度定义为

$$v := \mathop{\mathbb{E}}_{\boldsymbol{x}'\sim D}[L(\theta) - L(\theta')] = \alpha \| \mathop{\mathbb{E}}_{\boldsymbol{x}'\sim D} \nabla L(\boldsymbol{x}, \theta)\|^2$$

等式成立由上一个式子所保证。期望损失增加于是为

$$v_{PG} := \mathop{\mathbb{E}}_{\boldsymbol{x}'\sim D}[L(\boldsymbol{x}, \theta) - L(\boldsymbol{x}, \theta')] = \alpha \mathop{\mathbb{E}}_{\boldsymbol{x}'\sim D} \|\nabla L(\boldsymbol{x}, \theta)\|^2$$

定义

$$\mathbb{V}(\nabla L(\boldsymbol{x}, \theta)) := \mathbb{E}\|\nabla L(\boldsymbol{x}, \theta) - \mathbb{E}\nabla L(\boldsymbol{x}', \theta)\|^2$$

我们发现预测增益是两个项的总和：真正的预期学习进程，加上梯度方差：

$$v_{PG} = v + \mathbb{V}(\nabla L(\boldsymbol{x}, \theta)).$$

我们得出结论，为了平等的学习进度，基于预测增益的课程最大化方差。当使用梯度预测增益时，问题变得更糟，这实际上依赖于泰勒近似。另一方面，自我预测收益是对预期学习进度的无偏估计：

$$\mathbb{E}_{\boldsymbol{x}} v_{SPG} = \mathbb{E}_{\boldsymbol{x}, \boldsymbol{x}'\sim D}[L(\boldsymbol{x}', \theta) - L(x', \theta')] = v.$$

自然地，它使用两个样本导致比预测增益更高的方差，表明两个估计之间的偏差与方差存在权衡关系。

4 结论

我们的实验表明，只要使用合适的进度信号，使用随机教学大纲来最大限度地提高学习进度，可以大大提高课程学习效率。然而，我们注意到，从所有任务中统一抽样是一个惊人的强大基准。我们推测，这是因为尽管在一些不必要的样本的学习效率低下，但学习由网络正在取得最快进展的任务的梯度主导，引发了一种隐性的课程。对于最大似然训练，我们发现预测增益是最一致的信号，而对于变分推理训练，梯度变分复杂度增益表现最好。重要的是，两者都是瞬时的，因为它们只能使用用于训练的样本进行评估。除了更有效率之外，这对外部评估难度较大的任务也具有更广泛的适用性，并且表明学习进度最好在局部而不是全局的基础上进行评估。

致　　谢

作者感谢孟德宇老师的悉心指导、感谢马子璐同学对于自步学习优化部分内容的帮助、感谢马凡学长提供的自步学习协同训练与自步集成学习方面的支持、感谢赵谦老师和谢琦学长的建议、感谢机器学习讨论小组和 IID 小组的成员们。

作者同时感谢孟德宇老师、王尧老师、作者的父母、和郭力豪、宋豪、高宇翔等作者的室友为文章的修改提出的建议和帮助。

作者感谢参加审阅和论文答辩组的各位专家和老师，对你们所付出的辛勤劳动和给出的悉心指导表示深深的谢意！