

摘要

机器学习的研究已经发展了六十多个年头，在生活的方方面面都能看到人工智能的应用。其中，迁移学习作为一个重要的科研领域，可以将源域与目标域联系起来，使用源域的知识训练模型，并应用在目标域的分类问题上。近年来，基于领域适配的迁移学习算法在实验对比中显示出了良好的效果，与以往的传统迁移算法相比，不仅准确度有了较大的提升，而且算法的时间效率也有了不小的进步。然而，目前大多数迁移学习算法仅能适应一个源域与一个目标域的训练数据，无法适应多个不同分布的源域的样本。此外，部分迁移学习方法同步地从源域中学习数据至目标域模型，并没有根据重要程度量化学习知识的权重。在实践中，对于某数据集的数据按照某一或某些特征划分出来的数据往往分布不一致，并且这些不同分布的数据对于最终模型的重要性也不一致，知识迁移的权重也因此不平等。本文针对基于 Boosting 算法的迁移学习算法进行深入的研究，并从多个源域、相似性差异化迁移学习的角度对主流迁移学习 Boosting 算法进行优化改进，为提高多源域迁移学习的准确度与扩大多源域迁移学习的使用场景展开了相关的研究工作。本文的研究内容主要有以下两个方面：

(1) 为了实现在有监督的场景下多个源域迁移知识学习，提出了一种多源域优化权重迁移学习 (Multi-source Decision Tree Transfer Boosting, MtrBoost) 的方法。主要思想是根据不同分布的源域空间计算出到目标域的 KL 距离，利用 KL 距离的比值计算出不同分布的源域样本的学习权重比例，从而优化整体梯度函数，使学习方向朝着梯度下降最快的方向进行。相比于其它传统迁移学习算法，使用梯度下降算法能使模型较快收敛，在确保迁移学习效果的同时，也能确保学习的速度。

(2) 上述多源域迁移学习准确度较好，但是仅能适应与有监督场景下的多源域迁移学习。为了实现无监督场景下的多源域迁移学习，提出了一种多源域分布下优化权重的无监督迁移学习 (Unsupervised Multi-source Decision Tree Transfer Boosting, UMtrBoost) 方法，该方法根据各源域与目标的不同距离确定用于标注伪标签的训练数据，标注完成后，赋予不同源域不同的学习权重、赋予不同样本不同的样本权重，从而提升决策函数的分类精度。

关键词：深度决策树迁移学习，多源域迁移学习，无监督学习，KL 距离，决策树

Abstract

The research of machine learning has been developed for more than 60 years, and the application of artificial intelligence can be seen in all aspects of life. Among them, transfer learning, as an important scientific research field, can link the source domain with the target domain, use the knowledge training model of the source domain, and apply it to the classification of the target domain. In recent years, the transfer learning algorithm based on domain adaptation has shown good results in the comparison of experiments. Compared with the traditional transfer algorithm, not only the accuracy has been greatly improved, but also the time efficiency of the algorithm has been improved. However, at present, most migration learning algorithms can only adapt to the training data of one source domain and one target domain, and cannot adapt to the samples of multiple different source domains. In addition, some transfer learning methods synchronously learn data from the source domain to the target domain model, and do not quantify the weight of learning knowledge according to the importance. In practice, the data divided according to one or some characteristics of a data set are often not uniformly distributed, and the importance of these different distributed data to the final model is also inconsistent, and the weight of knowledge transfer is therefore not equal. This paper conducts in-depth research on the transfer learning algorithm based on Boosting algorithm, and optimizes and improves the mainstream transfer learning Boosting algorithm from the perspective of multiple source domains and similarity difference transfer learning, and carries out relevant research work to improve the accuracy of multi-source domain transfer learning and expand the use scenarios of multi-source domain transfer learning. The research content of this paper mainly includes the following two aspects:

(1) In order to realize the knowledge learning of multi-source domain transfer in a supervised scenario, a multi-source decision tree transfer boosting (MtrBoost) method is proposed. The main idea is to calculate the KL distance to the target domain according to the source domain space of different distributions, and use the ratio of KL distance to calculate the learning weight proportion of the source domain samples of different distributions, so as to optimize the overall gradient function, so that the learning direction is the direction of the fastest gradient decline. Compared with other traditional transfer learning algorithms, the gradient descent algorithm can make the model converge faster, and ensure the learning speed while ensuring the transfer learning effect.

(2) The above multi-source domain transfer learning accuracy is good, but it can only adapt to multi-source domain transfer learning in supervised scenarios. In order to realize multi-source domain transfer learning in unsupervised scenarios, an unsupervised multi-source decision tree transfer boosting (UMtrBoost) method with optimized weights under multi-source

domain distribution is proposed. This method determines the training data for labeling false labels according to the different distance between each source domain and the target. After labeling, different learning weights are given to different source domains. Different samples are given different weights to improve the classification accuracy of the decision function.

Keywords: Deep decision tree transfer learning, Multi-source domain transfer learning, unsupervised learning, KL divergence, Decision tree

目 录

第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.3 论文章节安排.....	3
第二章 迁移学习相关理论基础	5
2.1 引言.....	5
2.2 迁移学习的基本概念.....	7
2.2.1 迁移学习问题形式化.....	7
2.2.2 分布相似度衡量标准.....	7
2.3 迁移学习的分类.....	8
2.3.1 基于特征选择的迁移学习.....	8
2.3.2 基于模型的迁移学习.....	9
2.3.3 基于关系的迁移学习.....	10
2.3.4 基于样本的迁移学习.....	11
2.4 数据分布自适应.....	12
2.4.1 边缘分布自适应方法.....	12
2.4.2 条件分布自适应方法.....	12
2.4.3 联合分布自适应方法.....	13
2.5 本章小结.....	13
第三章 多源域分布下优化权重的迁移学习 Boosting 方法.....	15
3.1 引言.....	15
3.2 相关工作.....	15
3.2.1 单源域迁移学习算法.....	15
3.3 多源域分布下的迁移学习	17
3.3.1 问题描述.....	17
3.3.2 多源域分配权重的 Boosting 迁移学习算法描述.....	17
3.3.3 收敛性分析.....	21
3.3.4 伪代码和复杂度分析.....	23
3.4 实验和结果分析.....	25
3.4.1 数据集简介.....	25
3.4.2 实验设置.....	27
3.4.3 实验结果分析.....	30
3.5 本章小结.....	32
第四章 多源域分布下优化权重的无监督迁移学习 Boosting 方法..	35

4.1 引言.....	35
4.2 多源域分布下的迁移学习	35
4.2.1 问题描述.....	35
4.2.2 多源域分布下优化权重的无监督迁移学习 Boosting 方法	36
4.2.3 伪代码和复杂度分析.....	40
4.3 实验和结果分析.....	42
4.3.1 数据集简介.....	42
4.3.2 实验设置.....	44
4.3.3 实验结果分析.....	46
4.4 本章小结.....	48
第五章 主要结论与展望	49
5.1 主要结论.....	49
5.2 展望.....	49
参考文献.....	51

第一章 绪论

1.1 研究背景及意义

21 世纪是一个信息化的时代，现代通信技术与网络技术飞速发展。信息的获取与传播成本大大下降，信息的时效性也大幅提高。信息已经成为现代人生活必不可少的一部分，如何高效地获取数据、处理数据并在此基础上使用信息做出相应的选择已经成为现代信息技术的基石。但是，信息数据时时刻刻都在产生，全世界的网络、社交软件、智慧交通、智慧物流都在产生海量的数据，这些数据的类型也多种多样，包括但不限于图片、视频、文本等。这些数据都在给机器学习带来了新的挑战：数据的时空分布可能不一致以及数据可能缺少必要的标签。

近年来，随着机器学习在社会各行各业的应用逐步扩大，训练数据存在缺陷的情况也越来越多，因此需要有相应的办法使这些存在瑕疵的数据发挥其价值，迁移学习应运而生。在传统的机器学习分类任务中，训练数据有以下两点要求：第一，训练的数据与最终测试的数据分布必须一致；第二，训练数据的样本容量必须足够。然而，在实际的应用场景中，同时满足这两点要求的训练数据往往不多。首先，随着时间的变化，原先的数据可能与现在的数据分布不一致，这一点在时序性相关的数据上十分明显。例如，一年前的期货市场交易价格以及交易量数据，显然不能通过上一年的期货交易情况去预测现在的期货交易价格与交易量，因为这些数据是具有时效性的，对于现在而言数据可能已经过期。此外，训练数据的量可能不足且缺乏标签，人工逐一标注数据的成本可能十分高昂。因为迁移学习可以不遵从机器学习以上两个必要的条件，所以迁移学习^[1-3]可以从不同分布的数据中抽取知识逐步转移到目标模型中。迁移学习在日常生活中应用也十分常见，当两个不同的行为或者事物拥有较多的相似成分时，学会其中一种，学习另外一种也会变得相对容易，这是因为知识具有高度的相似性，从而成功的完成了迁移。与之相反，如果两个领域的相似成分较少或者没有，那么迁移知识的过程会较为困难，甚至会出现预期之外的“负迁移”^[4-5]现象。

现实情况下，获取数据的来源大多来自数字设备的自动记录，获取的数据大多为原始数据，通常不能直接使用，需要经过预处理加工。在加工预处理的过程中需要对异常数据进行剔除、对空缺值进行填充。经过了预处理的数据往往会发现不完整的现象，因此对于不完整^[6-7]的训练数据，迁移学习展现出了其极大的应用价值。

从源域的数目区分，迁移学习可以分为单源域迁移学习与多源域迁移学习两种不同的类型。单源域迁移学习只有一个目标域与一个源域，知识仅从单一的源域迁移到目标域中。多源域迁移学习拥有多个源域，知识按照不同的比例从不同的源域迁移到目标域中。从训练数据是否进行标注区分，迁移学习又可以分为有监督与无监督的场景下两种不同的情况。

在多模态^[8-9]或多视图^[10-13]场景下如何恰当的抽取知识进行学习是迁移学习首要的

目标。迁移学习的分类标准多种多样,不同的学者根据不同的学习方法或者使用场景划分了不同的类型。绝大多数的迁移算法只能从一个源域迁移知识,或者不能根据相似程度量化迁移学习知识。因此,当处于多源域迁移场景中,知识学习的效果往往不尽如意。

1.2 国内外研究现状

随着社会数字化的发展,如何对获取的数据进行相应的处理和训练,使其成为数据决策的依据成为了当前研究的重点。近些年来,迁移学习因其自生优秀的适应性特点,吸引了大批的知名学者投入精力进行研究。

在迁移学习中,Liu^[14]等人提出了一种 Butterfly 学习框架,该框架同时使用四个深度学习网络,其中两个负责所有域的适应工作,剩下的两个专门负责目标域的分类。Wang 等人^[15]给出了负迁移的定义,并提出了一种形式化的避免负迁移的方法。Scott^[16]提出了一种新的可以强化领域适配表现的度量方法。Yuta 等人^[17]提出了一种归纳迁移学习方法,通过基于 Cycle-GAN(Cycle-consistent generative adversarial networks)的无监督域适应使源域样本的注释标签应用于目标域数据样例上。Bucci 等人^[18]提出了一种以监督方式学习语义标签的模型,并使用相同图像自监督信号拓宽对数据的理解。Pablo 等人^[19]提出了一个基于模块化变分高斯过程迁移学习框架,可以在不重新访问任何数据的情况下构建集成变分高斯 GP(Gaussian processes)模型。

Boosting 方法在迁移学习中较为常见,通过多次迭代训练多个弱分类器从而集成输出为一个强分类器^[20]。Zhang 等人^[21]提出了一种新的加权技术,生成具有加权源域和目标域实例的加权合成实例。Schapire 等人^[22]提出了经典的 AdaBoost (Adaptive Boost) 算法,在每一次迭代训练过程中,分类错误的样本权重或提高。更新权重过后的样本会用于下一个基分类器的训练。Cortes 等人^[23]提出了一种新的集成学习算法 DeepBoost,它可以使用包含深度决策树或其他丰富或复杂的家族成员的假设集作为基分类器,并在不过度拟合数据的情况下获得较高的精度。Dai 等人^[24]提出了 TrAdaBoost(Adaptive boosting for transfer learning)算法,该算法对于分布不一致的训练数据做了优化,可以自动调整训练样本的权重,利用 Boosting 来区分分布不一致的训练数据。但由于假设的固定性和高复杂度,TrAdaBoost 存在过拟合的问题,在迁移学习的场景下可能拟合与目标域分布不一致的源域数据。为了弥补这一缺陷,Jiang 等人^[25]提出了 DTrBoost(Deep Decision Tree Transfer Boosting)算法,该方法通过根据 Rademacher 复杂度最小化源域和目标域的数据依赖学习边界来学习并分配给基础学习者,这保证了该算法可以在不过度拟合的情况下学习深度决策树。周等人^[26]提出一种可以在源域和目标域的特征空间中过采样的多源在线迁移学习算法,使用类别分布较为平衡的源域数据训练分类器,从而提升精度。徐等人^[27]提出了一种借助具有完整模态的辅助数据集,通过跨模态或跨数据集方向的迁移学习来帮助模态或数据集之间的数据对齐,从而实现更好的分类效果。林等人^[28]提出一种对抗域适应深度对抗重构分类网络的迁移学习模型 DARCNet (Deep Adversarial Reconstruction Classification Networks)。DARCNet 借鉴了自动编码器的思想,

在对抗域适应的基础上，增加了自动解码器的解码部分。Menapace 等人^[29]提出了一种域对齐层相结合的信息论损失，从多个源域收集训练样本的信息来构建与域无关的聚类模型。Huang 等人^[30]提出了一种有效标签传播方法，通过使用有效的域间和域内语义信息传播提高目标域中学习表示的辨别能力。Zhang 等人^[31]提出了一种基于同态加密的广泛使用的最大平均差异的安全版本，以在数据联合设置下实现有效的知识传输。Sharma 等人^[32]提出了一种基于实例亲和性的标准，用于在适应过程中进行源域到目标域的转移。Mitsuzumi 等人^[33]提出了一种自我监督的类破坏性学习方法，它可以在不使用任何域标签的情况下学习类不变表示和域对抗分类器。Le 等人^[34]提出了一种在一般域适应中通过优化传输减少标签和数据偏移的方法，可以同时减轻数据和标签偏移。

领域适配是一种传统的迁移学习方法，Long 等人^[35]提出了一种联合分布适应的迁移学习方法，可以在降维过程中共同调整边际分布和条件分布，并构建对实质性分布差异有效且稳健的新特征表示。Lu 等人^[36]提出了一种基于学习的方法来直接调整源和目标张量表示，引入了一组对齐矩阵，以将来自两个域的张量表示对齐到不变张量子空间，而无需矢量化。Cao 等人^[37]提出了一种基于结构风险最小化原则的新分布匹配机，通过提取不变的特征表示和估计共同最小化跨域分布差异的无偏实例权重来学习转移支持向量机。杨等人^[38]提出了基于二分类模型的直推式 0 阶模糊系统目标函数来训练系统参数。张等人^[39]提出了一种映射相关对齐的方法，利用深度神经网络进行端到端域自适应。Lu 等人^[40]提出了一种域适应的子结构级匹配的方法，通过聚类方法获得活动的子结构，利用活动数据的局部性信息进行准确高效的知识迁移。左等人^[41]提出一种建立在数据增强与迁移学习基础上的皮肤癌诊断方法 SC-Net(Skin Cancer-Net)，使用过采样技术补充少数样本数据，从而提高检测精度。

在多源迁移学习中，可以使用多个目标域的知识提升目标域模型的分类能力。Ge 等人^[42]提出一种监督局部加权方法 SLW (Supervised Local Weight)，将迁移的过程分为两个步骤，第一步先计算源域分类器对于目标域的判断能力，以此分配不同的权重。第二步使用目标域的训练数据训练加权的源域分类。Zhang 等人^[43]提出一种多源领域知识转移网络 MKT (Multi-source domain Knowledge Transfer)，使用浅层网络提取各个源域的低维特征完成特征对齐，之后使用多分支网络完成多个源域与目标域的高维特征对齐。

1.3 论文章节安排

本文的主要内容包括五部分，具体如下：

第一章为本文的绪论部分，重点阐述了本文的科研背景和研究意义，介绍了迁移学习算法在国内外的研究进展。

第二章为本文的理论基础，主要介绍了迁移学习相关的理论基础。

第三章对本文所提出的多源域分布下优化权重的迁移学习 Boosting 方法进行介绍，主要描述了有监督场景下多源域迁移学习的过程、算法时间复杂度分析、收敛性分析、参数设置对比分析、基线算法、实验设置、在 8 个文本数据集和 1 个图片数据集上实验

结果分析与小结。

第四章对本文所提出的多源域分布下优化权重的无监督迁移学习 **Boosting** 方法算法进行介绍，主要描述了无监督场景下多源域迁移学习的过程、算法时间复杂度分析、实验设置、基线算法、在 8 个文本数据集和 2 个图片数据集上实验结果分析与小结。

第五章是本文的结论和展望部分，汇总了全文所做的工作，指出本文研究的优势以及不足。最后展望了下一步的研究方向和未来的工作方向。

第二章 迁移学习相关理论基础

2.1 引言

机器学习是人工智能领域中广泛使用的方法，也是目前在机器智能领域效果最为显著的一种方法。机器学习的核心重点在于如何从已有的训练数据中提取相应知识用以解决目标问题。迁移学习的范畴属于机器学习，但是又与传统的机器学习有所区别。传统的机器学习训练数据与目标问题的数据分布一致，但是迁移学习需要将知识从训练数据中提取出来用于与原问题数据分布不一致的新问题上。这要求原问题与新问题必须要有相似之处，不然无法完成知识的迁移。如图 2-1 所示，学会了驾驶小汽车就会很容易学会驾驶卡车，这是因为这两件事有较高的相似度。

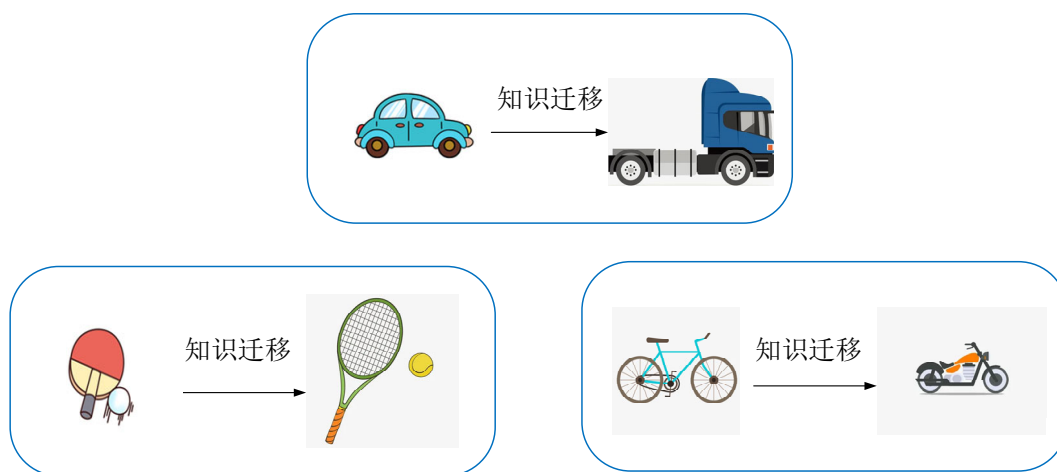
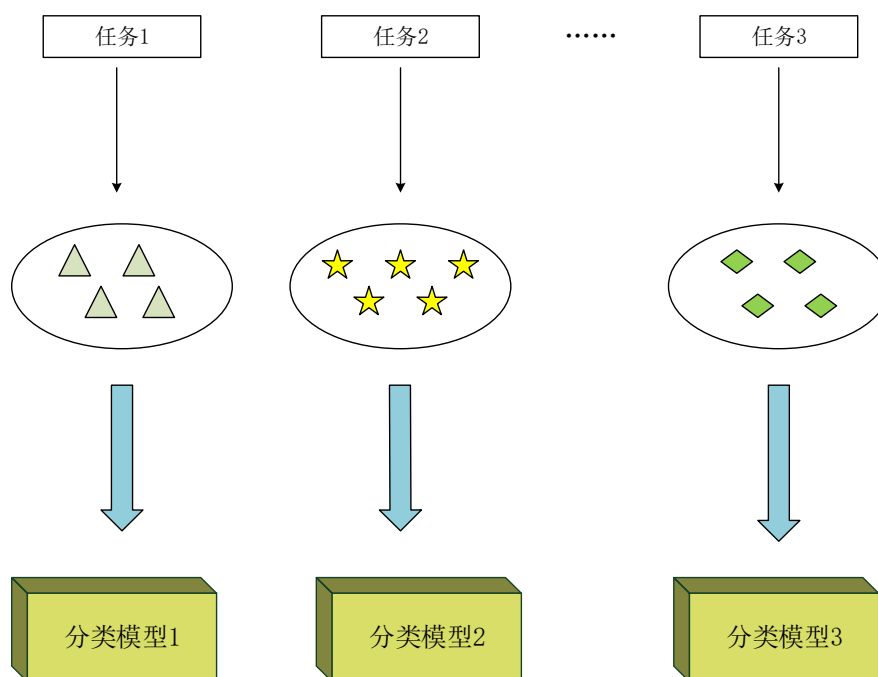


图 2-1 迁移学习示例

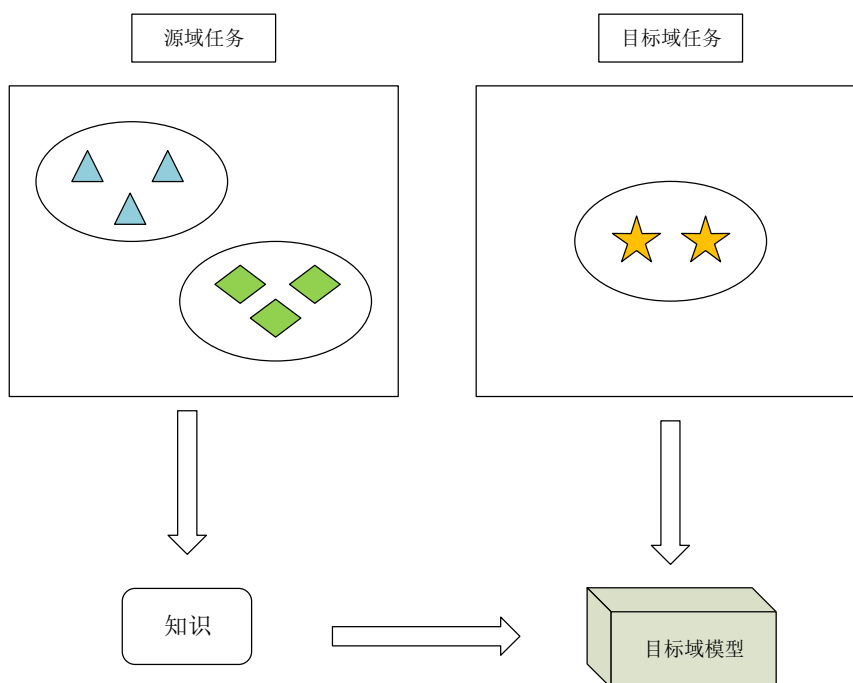
迁移学习的目的是将源域的知识迁移至目标域训练模型，从而完成分类问题。迁移学习诞生之初的目的就是解决训练数据不足、训练数据与目标域数据分布不同的问题。这与目前数据收集困难、标注困难的情况相适应，因此导致了近些年来迁移学习学术浪潮。但是迁移学习并不是没有局限性的，其迁移的过程也是其脆弱的地方，当源域与目标域之间相似的地方较多的时候，迁移学习的效果也就较好。反之，如果相似性较差，那么学习的效果将大打折扣。例如，会驾驶小汽车的人去学习驾驶卡车会十分容易，但要是使用驾驶小汽车的经验去驾驶飞机那么就十分的危险，造成这样的原因在于驾驶小汽车与驾驶飞机这两件事的相似性十分小。

迁移学习的主要方法可以分为四种^[43]：基于特征的迁移学习、基于样本的迁移学习、基于模型的迁移学习和基于关系的迁移学习。目前，研究的热点在于基于特征的迁移学习，该方法通过特征提取变化、降低特征维度等方法将源域与目标域的特征维度压缩到具有相似性的空间，这样就可以完成特征的匹配。基于样本的迁移学习的思想相对简单，需要从样本的角度出发，判断出源域中与目标域中哪些样本具有相似性，哪些样本不具备相似性。在有监督迁移学习的场景下可以完成这样的判断，但是在无监督迁移学习的

场景下很难做出准确的判断。基于模型的迁移学习和基于关系的迁移学习方法相对而言并不是十分热门，但是近些年的深度网络模型发展也使得这两种方法有了长足的进步。如图 2-2，展示了机器学习与迁移学习的区别，传统机器学习对于数据分布有严格要求，因此只能单独的训练模型，而迁移学习可以将其它源域的知识迁移到目标域中。



(a) 传统的机器学习方法



(b) 迁移学习方法

图 2-2 迁移学习与机器学习的区别

2.2 迁移学习的基本概念

2.2.1 迁移学习问题形式化

在迁移学习场景中，设 $X = X_s \cup X_d$ 为实例空间，其中 X_s 为目标域实例空间， X_d 为源域实例空间。源域的数据空间为 $X_d \times Y_d$ ，目标域的数据空间为 $X_s \times Y_s$ ，其中，源域与目标域使用相同的数据标签空间 $Y_d = Y_s = Y = \{-1, +1\}$ 。测试数据是与目标域同分布的数据，记为 Q 。设 q 是将 X 映射到 Y 的布尔函数，将训练数据 $L = \{X \times Y\}$ 分成 $L_d = \{x_i^d, q(x_i^d)\}$ ， $L_s = \{x_i^s, q(x_i^s)\}$ ， L_d 为不同源域上的训练数据空间， L_s 为目标域的训练数据空间。

因此，在给定少量带标记的目标域训练数据 L_s 和大量带标记的源域训练数据 L_d 的情况下，我们的目标是学习一个布尔函数 q 从 X 到 Y ，使测试集数据的预测误差值最小化。

2.2.2 分布相似度衡量标准

在进行迁移学习前需要对任务进行相似性判断，只有当两个任务具有足够的相似性才能进行迁移学习，否则无法确保准确度。目前主流的判断方法是使用数据分布的距离衡量相似度，样本分布的距离越小，相似度就越高，越容易完成迁移学习任务。反正，如果距离过大，那么两个任务之间的相似性也越小，越不容易完成迁移学习任务。通常情况下迁移学习的效果与样本分布的距离的大小密切相关。以下是几种常用的衡量距离的标准：

(1) 欧几里得距离

这是最直接的两点间距离估计，并且可以计算高维空间的样本距离。两个顶点之间的距离计算公式如下：

$$d = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.1)$$

(2) 曼哈顿距离

这是十九世纪赫尔曼·闵可夫斯基所提出的公式，也称为城市距离，该方法计算两个顶点之间坐标系上的绝对轴距总和。计算公式如下：

$$d = \sum_{i=1}^n |x_i - y_i| \quad (2.2)$$

(3) 闵氏距离

该距离公式是欧式距离的推广，该公式含有参数 p ，当 $p=2$ 时，该距离退化成欧几里得距离，和 p -范数在数值上相等。计算公式如下：

$$d = \sqrt[p]{\sum_{i=1}^n (x_i - y_i)^p} \quad (2.3)$$

(4) Hellinger 距离

将两个点作为概率意义进行距离计算，通常用于概率分布相似性计算，该公式是 f 散度的一种，计算方法如下：

$$H^2(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\int \left(\sqrt{\frac{dP}{d\lambda}} - \sqrt{\frac{dQ}{d\lambda}} \right)^2 d\lambda} \quad (2.4)$$

(5) 马氏距离

由印度统计学家马哈拉诺比斯提出，该距离计算样本的协方差距离，可以有效的计算两个样本集合的相似度。计算方法如下：

$$d = \sqrt{(\bar{x} - \bar{y})^T S^{-1} (\bar{x} - \bar{y})} \quad (2.5)$$

(6) Kullback-Leibler 散度

KL 散度又称为 KL 距离，使用的条件为计算距离的统计函数是凸函数且是一个闭函数。KL 距离没有上界，在数值上可以更为直观的表达数据分布情况。该公式的计算方法如下：

$$D(P \parallel Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \quad (2.6)$$

2.3 迁移学习的分类

2.3.1 基于特征选择的迁移学习

特征选择的方法考察源域与目标域的特征关系，并从中选择出目标域与源域共有的特征表示^[44-45]。该方法的最终结果会将分布于不同特征空间的源域、目标域特征汇聚到统一特征空间内。在该特征空间内，特征的分布差异十分细小，因此使用传统的机器学习方法训练目标域分类模型成为了可行方法。基于特征选择的迁移学习在业内受到了一致的好评，有大量的学者发表相关的研究成果。但是，基于特征选择的迁移学习要求源域与目标域特征有较多的相似性，如果相似性较少，那么特征选择的结果也不尽如意。该方法有大量经典的案例，例如联合聚类的分类迁移 CoCC^[46](Co-clustering Based Classification)、自由源选择自由优先级学习 FSFP^[47](Free Source selection Free Prior probability distribution)等，CoCC 算法同时对类别与特征进行聚类运算，从而完成知识与类别标签的迁移。FSFP 算法使用隐性语义分析关键词的特征，进而完成特征的选择和知识的迁移，这些方法都给后续的迁移学习研究者提供了思路。根据特征语义与维度的数目是否一致，基于特征选择的迁移学习又分为同构迁移学习与异构迁移学习方法，具体情况如图 2-3 所示。

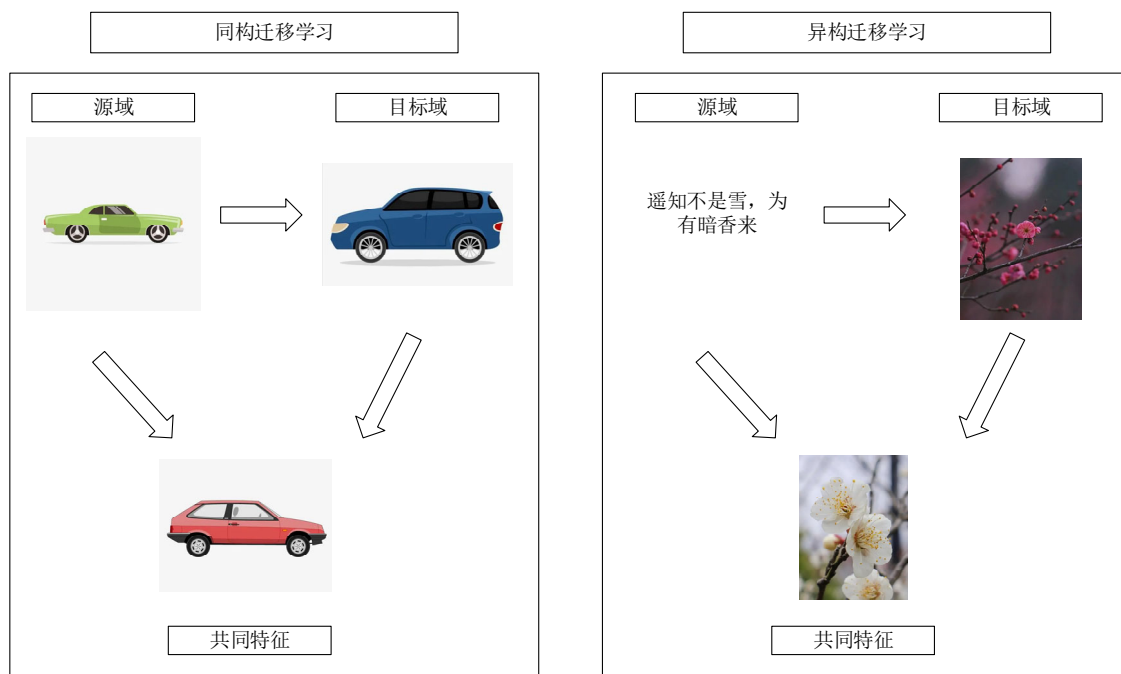


图 2-3 基于特征选择的迁移学习分类

2.3.2 基于模型的迁移学习

基于模型的迁移学习主要通过模型的参数共享实现知识的迁移，因此源域与目标域必须要有可以共享的参数。基于模型的迁移学习目前的研究成果并不是很多，有部分学者使用决策树的节点权重进行参数迁移^[48]，还有一部分学者改进支持向量机的参数^[49]，从而实现模型的迁移。随着神经网络的流行，基于网络结构的参数迁移学习也成为热点研究方向。如图 2-4 展示了基于模型的迁移学习的核心思想。

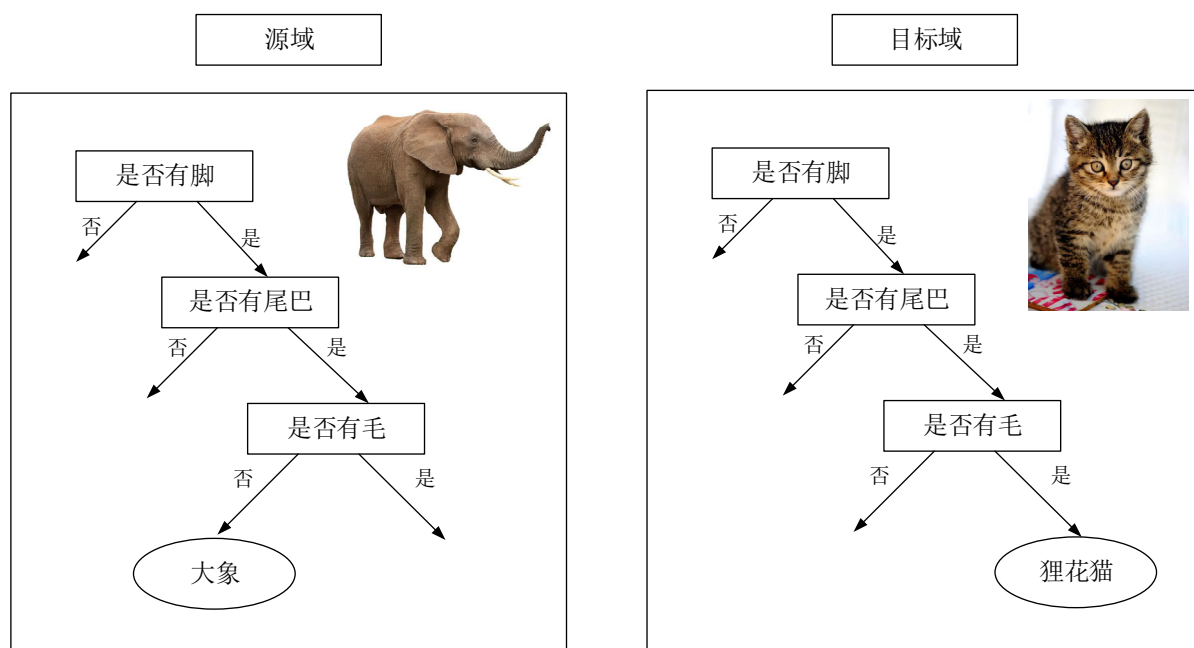


图 2-4 基于模型的迁移学习的核心思想

2.3.3 基于关系的迁移学习

基于关系的迁移学习的思路灵感来自于人类现实世界的类比推理关系，如图 2-5 所示，通过给出的巡逻保安可以联想到反病毒软件，通过邮递员可以联想到电子邮件系统。这种潜在的联系性正是基于关系的迁移学习的核心所在。但是，基于关系的迁移学习方向的研究成果有限，因为找出不同领域之间的联系相对困难。部分研究学者借助马尔科夫逻辑网络^[50-52]实现源域与目标域之间关系的配对，从而完成知识迁移，马尔科夫逻辑网络核心思想如图 2-6 所示。



图 2-5 基于关系的迁移学习方法示意图

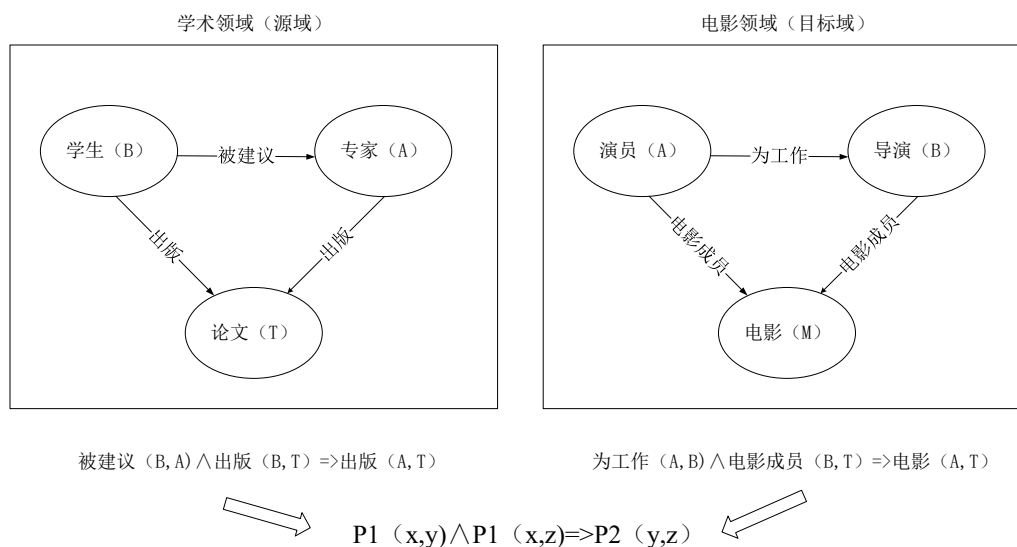


图 2-6 基马尔科夫逻辑网络核心思想示意图

2.3.4 基于样本的迁移学习

基于样本的迁移学习方法考察的是全体目标域与源域训练样本，对于来自不同源域与目标域的训练数据分配不同的学习权重，在之后的每一轮的学习中不断提升与目标域相似度高的源域样本权重，惩罚与目标域相似度较低的源域样本。如图 2-7 展示了基于样本的迁移学习核心思想。其中，笔记本电脑与平板电脑为源域，一体机为目标域，图片的大小代表该样本的权重，再多次的迭代学习后，源域样本的权重被降低，目标域的样本权重增加。



图 2-7 基于样本的迁移学习方法示意图

2.4 数据分布自适应

数据分布自适应 DA (Distribution Adaptation) 是迁移学习中较为流行的一种学习方式, 该方法直接作用与源域与目标域的训练样本数据。通过降维、变换视角等方法减少源域与目标域之间的分布距离。在经过变化之后使得源域与目标域的分布近似到可以直接进行训练。通常而言, 数据分布有两种情况: 数据边缘分布不一致、数据条件分布不一致。具体情况如图 2-8 所示。

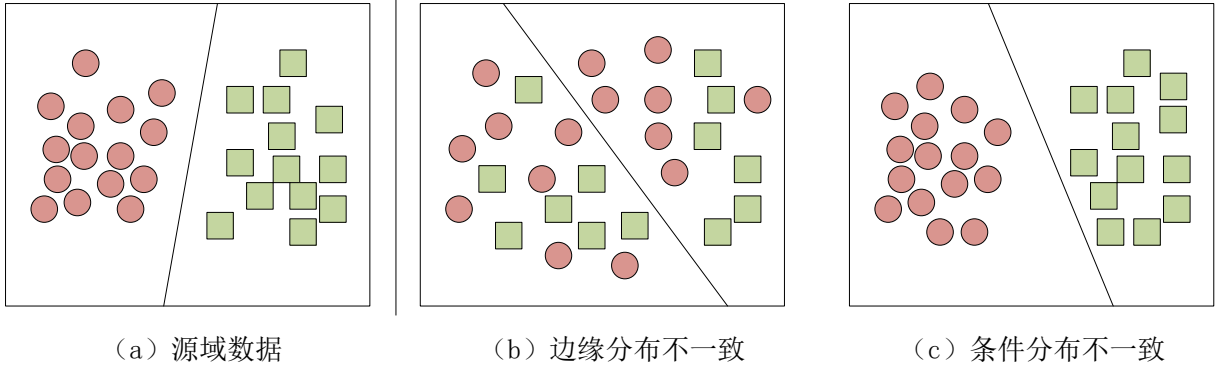


图 2-8 不同数据分布示意图

2.4.1 边缘分布自适应方法

边缘分布自适应方法 MDA (Marginal Distribution Adaptation) 可以有效地减少源域与目标域之间的概率分布差异。该方法使用 $P(\mathbf{x}_s)$ 与 $P(\mathbf{x}_t)$ 之间的差值表示距离的大小。具体的计算公式如下:

$$H(\mathcal{D}_s, \mathcal{D}_t) \approx \|P(\mathbf{x}_s) - P(\mathbf{x}_t)\| \quad (2.7)$$

典型的边缘分布自适应算法成为迁移成分分析 (Transfer Component Analysis, TCA), 该方法通过寻找一个特殊的特征映射函数, 使得源域与目标域的边缘分布与条件分布相近似。

2.4.2 条件分布自适应方法

条件分布自适应方法 CDA (Conditional Distribution Adaptation) 减少的是目标域与源域之间的条件概率分布距离, 该方法使用 $P(y_s | \mathbf{x}_s)$ 与 $P(y_t | \mathbf{x}_t)$ 之间的差值来计算, 具体计算方法如下所示:

$$H(\mathcal{D}_s, \mathcal{D}_t) \approx \|P(y_s | \mathbf{x}_s) - P(y_t | \mathbf{x}_t)\| \quad (2.8)$$

典型的条件分布自适应为层间迁移学习方法 STL (Stratified Transfer Learning), 该方法使用类内迁移的思想, 有别于现有的其它迁移学习方法只考察一个全局特征变化的方法, 更多的考察类内的特征情况, 从而实现更高精度的迁移学习。

2.4.3 联合分布自适应方法

联合分布自适应方法（Joint Distribution Adaptation）通过减少源域与目标域间的联合概率分布距离来进行适配。计算方法综合了条件分布与边缘分布两个因素。两个部分该方法使用如下公式来衡量源域与目标域之间的距离。

$$H(\mathcal{D}_s, \mathcal{D}_t) \approx \|P(\mathbf{x}_s) - P(\mathbf{x}_t) + P(y_s | \mathbf{x}_s) - P(y_t | \mathbf{x}_t)\| \quad (2.9)$$

联合分布自适应方法综合考虑了边缘分布于条件分布的情况，因此是一种较为优秀的自适应方法，在迁移学习中有着广泛的使用。

2.5 本章小结

本章内容简要介绍了迁移学习的基础理论，首先介绍了迁移学习的进展情况，其次介绍了基于特征的迁移学习、基于样本的迁移学习、基于模型的迁移学习和基于关系的迁移学习这四种主要的迁移学习分类。最后介绍了数据自适应的概念以及边缘分布自适应方法、条件分布自适应方法、联合分布自适应方法这三种主要的数据自适应方法。

第三章 多源域分布下优化权重的迁移学习 Boosting 方法

3.1 引言

迁移学习是为了解决源域与目标域数据空间分布不一致情况下，从源域中抽取有用的知识训练模型。因其可以在目标域训练数据不完备的情况下从源域迁移获取知识，从而减少了获取标注目标域数据的成本。近年来，迁移学习在各个领域都有着高度的关注与广泛的应用^[53]。当一组数据很容易过时，可能会出现迁移学习的需求。在这种情况下，在一个时间段内获得的标记数据在以后的时间段内可能不会遵循相同的分布^[54]。此外，在目标域没有标注或者标注目标域数据代价高昂的情况下，迁移学习可以节省大量的标记工作^[55]。在实践中，对于某数据集的数据按照某一或某些特征划分出来的数据往往分布不一致。并且这些不同分布的数据对于最终模型的重要性也不一致，知识迁移的权重也因此不平等。这就导致了学习的模型收敛速度的下降，同时也导致了学习过程中的振荡。并且 DtrBoost 算法对多源域迁移学习情况适应性不强，在多源域数据情况下不能很好地完成分类任务。

因此，本章内容提出了一种针对多源域不同分布数据的优化样本权重迁移学习算法。该算法使用 KL 散度^[56]（Kullback-Leibler Divergence）衡量源域与目标域的距离计算目标函数的源域权重，从而增强相似的源域目标函数权重，减少分布距离较大的源域目标函数权重，将源域分配不同的学习权重可以减小梯度方向的振荡，加快收敛速度。同时将不同的学习权重赋予不同的源域，使算法适应多源域迁移学习。之后，由目标函数导出梯度函数根据梯度下降最快的方向确定本批次的最佳梯度，并与之前的学习梯度的进行对比确定最优基分类器，并更新最优基分类器权重。完成了最优基分类器后按照源域与目标域不同的分布特点，对所有的训练样本进行权重更新。

3.2 相关工作

3.2.1 单源域迁移学习算法

在本章节简要介绍单源域迁移学习算法 DtrBoost，DeepBoost 提出了基于 Rademacher 复杂度^[57]的基分类器集的凸集学习边界。DTrBoost 在不同迭代时从源域数据和目标域数据中学习不同复杂度的假设族，并对较低复杂度的假设族分配更多的权值，以避免过拟合。

对于不同的假设集合 h_j ， $j \in [1, \dots, N]$ 。 $\{h_1, \dots, h_j, \dots, h_N\}$ 是一组不同的假设， F_w 是源域和目标域中数据梯度的损失函数， F_s 为目标域内数据梯度的损失函数。具体计算如下：

$$F_w^t = \frac{1}{n+m} \sum_{i=1}^{n+m} \Phi \left(1 - y_i \sum_{j=1}^N \alpha_{t,j} h_j(x_i, a_j) \right) + \sum_{j=1}^N (\lambda r_t + \beta) \alpha_{t,j} \quad (3.1)$$

$$F_s^t = \frac{1}{m} \sum_{i=1}^m \Phi \left(1 - y_i \sum_{j=1}^N \alpha_{t,j} h_j(x_i, a_j) \right) + \sum_{j=1}^N (\lambda r_t + \beta) \alpha_{t,j} \quad (3.2)$$

其中, $\alpha_{t,j}$ 为假设 j 在第 t 次迭代中的基分类器权重, Φ 是一个非递增的凸函数。式(3.1),

(3.2) 中的第二部分是假设 Rademacher 复杂度的正则化, 其中 λ 和 β 为参数。

$r_t = \mathfrak{R}_m(H_{d(h_t)})$ 是标准的 Rademacher 复杂度。 $d(h_t)$ 表示 h_t 所属的假设集的索引, 即

$h_t \in H_{d(h_t)}$ 。接下来就是训练的目标函数, 目的是最小化 F_w 与 F_s 在 e_j 方向上的梯度差值。

$$\min_{\alpha_{t,j}} G = \min_{\alpha_{t,j}} \left| F_w^t(\alpha_{t-1,j}, e_j) - F_s^t(\alpha_{t-1,j}, e_j) \right| \quad (3.3)$$

在第 t 次迭代中, 通过式 (3.4) 梯度增强函数选择了一个新的函数 $h_t(x_i, a_t)$ 作为当前最优的假设。

$$d_t^w(x) = \left[\frac{\partial \Phi(y, f(x))}{\partial f(x)} \right]_{f(x_i)=f_{t-1}(x_i)} \quad (3.4)$$

其中, $f_t \leftarrow f_{t-1} + \alpha_t h_t(x, a_t)$ 。 F_w^t , F_s^t 分别是 F_w 与 F_s 在 e_j 方向上的导数。然后开始迭代训练, 在每一个迭代次序内计算目标域的梯度值为:

$$\begin{aligned} d_{t,j}^s &= F_s^t(\alpha_{t-1,j}, e_j) \\ &= -\frac{1}{m} \sum_{i=1}^m y_i h_j(x_i) \Phi'(1 - y_i f_{t-1}(x_i)) + \alpha_{t-1,j} \Lambda_t^s \\ &= -\frac{1}{m} \sum_{i=1}^m y_i h_j(x_i) D_t^s(i) S_t^s + \alpha_{t-1,j} \Lambda_t^s \\ &= (2\epsilon_{t,j}^s - 1) \frac{S_t^s}{m} + \alpha_{t-1,j} \Lambda_t^s \end{aligned} \quad (3.5)$$

其中, S_t^s 是正则化后的权重分布, $D_t^s(i)$ 为目标域样本权重, $\epsilon_{t,j}^s$ 为目标域分类错误数,

Λ_t^s 为目标域惩罚项。在每次迭代时选出 $|d_{t,j}^s|$ 最大的基分类器为当前最优分类器, 并更新该分类器的权重 $\alpha_{t,j}$ 。在每次迭代的最后, 按照源域与目标域分别更新样本的权重。

最终的决策函数为:

$$f = \sum_{j=1}^N (\alpha_{T,j} - \alpha_{T/2,j}) h_j \quad (3.6)$$

上述算法只能从一个源域中学习知识转移到目标域中，在多个源域的情况下缺乏适应性。下面介绍一种新的多源迁移算法，可以从不同分布的多个源域中按分布相似程度确定知识重要性，增加相似源域样本权重，减少其余样本权重。从而减少分类的错误率。

3.3 多源域分布下的迁移学习

3.3.1 问题描述

在具有多个源域的迁移学习场景中，设 $X = X_s \cup X_{d1} \cup \dots \cup X_{dp}$ 为实例空间，其中 X_s 为目标域实例空间， X_{dp} 为 p 个分布不同的源域实例空间。源域的数据空间为 $X_{dk} \times Y_{dk}$ ，其中 $1 \leq k \leq p$ 。目标域的数据空间为 $X_s \times Y_s$ ，其中，源域与目标域使用相同的数据标签空间 $Y_{dk} = Y_s = Y = \{-1, +1\}$ 。测试数据是与目标域同分布的数据，记为 Q 。设 q 是将 X 映射到 Y 的布尔函数，将训练数据 $L = \{X \times Y\}$ 分成 $L_{dk} = \{x_i^{dk}, q(x_i^{dk})\}$ ， $L_s = \{x_i^s, q(x_i^s)\}$ ， L_{dk} 为不同源域上的训练数据空间， L_s 为目标域的训练数据空间。

因此，在给定少量带标记的目标域训练数据 L_s 和大量带标记的 p 个源域训练数据 $L_{d1}, \dots, L_{dp}$ 的情况下，我们的目标是学习一个布尔函数 q 从 X 到 Y ，使测试集数据的预测误差值最小化。

源域的训练数据来自 p 个不同的分布，如果将 p 个源域数据合并视为一个源域，在计算全局梯度的时候，不同源域上的知识重要程度是同等的。然而实际的数据相似程度并不一致，会导致相似程度较小的源域知识也同等的学习。考虑到源域的相似程度，按照相应的权重学习不同源域的知识，提高相似源域的学习权重，减少相似程度较小源域的、学习权重，从而获得更高的测试精度。

3.3.2 多源域分配权重的 Boosting 迁移学习算法描述

本节提出一种称为 Multi-source Decision Tree Transfer Boosting (MtrBoost) 的算法，该算法通过赋予不同源域不同的学习权重计算全局梯度从而提升目标域决策函数的精确度。

首先计算 KL 距离 (Kullback-Leibler Divergence) (3.7)，根据源域的 KL 距离确定两个源域的相似程度。计算距离的统计函数还有很多，例如 Jensen Shannon 距离、Hellinger 等距离。只要计算距离的统计函数是凸函数且是一个闭函数，都可以用来衡量数据分布情况^[58]。但这两种距离都具有上界，KL 距离没有上界，在数值上可以更为直

观的表达数据分布情况。通常情况下，迁移学习中较为普遍地使用 KL 距离衡量数据分布情况，因此本文也选择 KL 距离确定源域与目标域的相似程度。

$$KL(L_d \parallel L_s) = \sum L_d(x) \ln \frac{L_d(x)}{L_s(x)} \quad (3.7)$$

其中 $L_d(x)$ 是源域上的样本， $L_s(x)$ 是目标域样本。分别计算出源域 $L_{d1}, L_{d2}, \dots, L_{dp}$ 到目标域对应的 KL 距离为 $KL_{d1}, KL_{d2}, \dots, KL_{dp}$ ，之后根据式 (3.8) 计算出对应的学习权重 $\omega_1, \omega_2, \dots, \omega_p$ 。式 (3.8) 可以根据这 p 个源域 KL 距离的大小分配对应的权重，并且满足 $\omega_1 + \omega_2 + \dots + \omega_p = 1$ 的约束。

$$\omega_j = \frac{\prod_{t=1}^p (1 + e^{-KL_{dt}}) / (1 + e^{-KL_j})}{\sum_{i=1, i < j \leq p}^p (1 + e^{-KL_{di}}) (1 + e^{-KL_{dj}})} \quad (3.8)$$

其中， $1 \leq j \leq p$ ， ω_j 表示第 j 个源域的学习权重。 $H = \{h_1, \dots, h_j, \dots, h_N\}$ 是 N 个不同的假设的集合。 Q_{dk} 是第 k 个源域上的损失函数， Q_{ta} 是目标域上的损失函数。

$$Q_{dk}^t = \sum_{i=w(k)+1}^{w(k)+m_k} \frac{\psi \left(1 - y_i \sum_{t=1}^T \tau_t h_t(x_i, b_t) \right)}{m_k} + \quad (3.9)$$

$$\sum_{j=1}^N (\sigma r_t + \gamma) \tau_{t,j}$$

$$Q_{ta}^t = \frac{1}{n} \sum_1^n \psi \left(1 - y_i \sum_{t=1}^T \tau_t h_t(x_i, b_t) \right) + \quad (3.10)$$

$$\sum_{j=1}^N (\sigma r_t + \gamma) \tau_{t,j}$$

其中， n 为目标域训练集 L_s 上的实例数目， m_k 为源域训练集上第 k 个源域的实例数目， $w(k) = n + m_1 + \dots + m_{k-1}$ 为第 k 个源域样本起始位置的映射函数。 x_i 是输入的实例，且 $i \in \{1, \dots, n, \dots, n + m_1, \dots, n + m_1 + \dots + m_p\}$ ， y_i 是 x_i 真实类别。 $t \in \{1, \dots, T\}$ 是迭代训练的次数。 $h_t(x_i, b_t)$ 是在第 t 次迭代时的假设， b_t 是假设的参数，包括最佳分割特征、剪枝节点位置。 τ_t 是假设 h_t 的权重。 ψ 是非增凸函数，与 DtrBoost 一致，这里选择指数函数。式 (3.9)、(3.10) 的第二部分是正则化后的 Rademacher 复杂度， $r_t = \mathfrak{R}_m(H_{f(h_t)})$ 其中 $f(h_t)$ 是将假设 h_t 映射到该假设下标， σ 、 γ 是参数。

在得到了源域与目标域的损失函数后，在源域与目标域总体上的全局损失函数如下：

$$\begin{aligned}
 Q'_w = & \sum_{k=1}^p \frac{\omega_k}{2m_k} \sum_{i=w(k)+1}^{w(k)+m_k} \psi \left(1 - y_i \sum_{t=1}^T \tau_t h_t(x_i, b_t) \right) + \\
 & \frac{1}{2n} \sum_{i=1}^n \psi \left(1 - y_i \sum_{t=1}^T \tau_t h_t(x_i, b_t) \right) + \\
 & \sum_{j=1}^N (\sigma r_t + \gamma) \eta_{t,j}
 \end{aligned} \tag{3.11}$$

式 (3.11) 中，在源域与目标域的损失函数上增加了对应的权重，从而强调了目标域的损失。通常情况下目标域的样本重要程度大于各个源域，各源域的样本重要程度相对一致。因此本文对于源域分配一定系数后，各源域平均分配剩下的系数，使得各源域与目标域系数和为 1。对于各个不同的源域也按照 KL 距离分别赋予了权重。由此可以得到目标函数。

$$\min_{\tau_{t,j}} G = \min_{\tau_{t,j}} \left| Q'_w(\tau_{t-1,j}, \mu_j) - Q'_{ta}(\tau_{t-1,j}, \mu_j) \right| \tag{3.12}$$

式 (3.12) 中， μ_j 是第 j 个假设的方向， Q'_w 、 Q'_{ta} 分别是全局损失函数与目标域损失函数在 μ_j 方向上的导数。在每次的迭代学习中，不断的减小整体损失 Q'_w 与目标域的损失差值。同时，源域中不同分布的样本权重也会不断减小。通过惩罚项，复杂假设的惩罚项会更大，这样可以避免算法陷入过拟合的情况。

对于全部的样本，使用式 (3.11) 进行学习，在方向 μ_j 上的导数可以通过式 (3.13) 计算得到。

$$\begin{aligned}
 d_{t,j}^w &= Q'_w(\tau_{t-1,j}, \mu_j) \\
 &= - \sum_{k=1}^p \frac{\omega_k}{2m_k} \sum_{i=w(k)+1}^{w(k)+m_k} y_i h_j(x_i) \psi'(1 - y_i g_{t-1}(x_i)) \\
 &\quad - \frac{1}{2n} \sum_{i=1}^n y_i h_j(x_i) \psi'(1 - y_i g_{t-1}(x_i)) + \tau_{t-1,j} Y_t^w \\
 &= - \sum_{k=1}^p \frac{\omega_k}{2m_k} \sum_{i=w(k)+1}^{w(k)+m_k} y_i h_j(x_i) N_t^{dk}(i) v_t^{dk} \\
 &\quad - \frac{1}{2n} \sum_{i=1}^n y_i h_j(x_i) N_t^{ta}(i) v_t^{ta} + \tau_{t-1,j} \gamma_t^w \\
 &= \sum_{k=1}^p \left(2\kappa_{t,j}^{dk} - 1 \right) \frac{\omega_k v_t^{dk}}{2m_k} + \left(2\kappa_{t,j}^{ta} - 1 \right) \frac{v_t^{ta}}{2n} + \tau_{t-1,j} \gamma_t^w
 \end{aligned} \tag{3.13}$$

其中， $g_t \leftarrow g_{t-1} + \eta_t h_t(x, b_t)$ ， Q'_w 是在 μ_j 方向上的全局梯度， $Y_t^w = \sigma r_t + \gamma$ 是正则化后的惩罚项， $\kappa_{t,j}^{ta}$ 是目标域样本估计错误， $\kappa_{t,j}^{dk}$ 是不同源域的样本估计错误， $N_t^{ta}(i)$ 表示目标域样本 x_i 的权重， $N_t^{dk}(i)$ 表示第 k 个源域样本 x_i 的权重， v_t^{ta} 、 v_t^{dk} 分布是目标域样本、第

k 个源域样本在第 t 次迭代时的正则化权重。

在计算出梯度 $d_{t,j}^w$ 后，梯度增强函数将会选择出与 $-d_{t,j}^w$ 最相关的决策树分类器，选择函数如下：

$$b_j = \underset{b_j}{\operatorname{argmin}} \sum_{i=1}^{n+m_1+\dots+m_p} \left[-d_j^w(x_i) - h_j(x_i, b_j) \right]^2 \quad (3.14)$$

至此，在经过了 t 次迭代后，假设集合更新为 $H = \{h_1, \dots, h_j, \dots, h_N\}$ ，其中 $N \leq t \leq T$ 。

之后，计算目标域的梯度导数，在每次的迭代计算过程中，计算 Q_{ta}^t 在 μ_j 方向上的导数，计算步骤与计算全局样本上的梯度大小一致。

$$\begin{aligned} d_{t,j}^{ta} &= Q_{ta}^t(\tau_{t-1,j}, \mu_j) \\ &= -\frac{1}{n} \sum_{i=1}^n y_i h_j(x_i) \psi'(1 - y_i g_{t-1}(x_i)) + \tau_{t-1,j} \gamma_t^{ta} \\ &= -\frac{1}{n} \sum_{i=1}^n y_i^{ta} y_i h_j(x_i) N_t^{ta}(i) v_t^{ta} + \tau_{t-1,j} \gamma_t^{ta} \\ &= (2\kappa_{t,j}^{ta} - 1) \frac{v_t^{ta}}{n} + \tau_{t-1,j} Y_t^{ta} \end{aligned} \quad (3.15)$$

其中， Q_{ta}^t 是在 μ_j 方向上的全局梯度， $Y_t^{ta} = \sigma r_t + \gamma$ 是正则化后的惩罚项， $\kappa_{t,j}^{ta}$ 是估计错误，计算方法与 DtrBoost 一致， $N_t^{ta}(i)$ 表示目标域每个样本 x_i 的权重， v_t^{ta} 是目标域样本在第 t 次迭代时的正则化权重。每次迭代过程中，选择目标域梯度值最大的假设作为最佳学习者 h_j 。通过这种迭代方法，可以找到与目标域损失函数 Q_{ta} 梯度最接近的方向 μ_j 。

在搜索到最佳学习者后 h_j ，对 h_j 的权重进行更新。基本的更新学习者权重思路是每次迭代仅更新当前搜索到的最佳学习者 h_j ，并增加 h_j 的权重，其它学习者的权重保持不变，更新公式如下。

$$\tau_t = \tau_{t-1} + \eta_t \mu_j \quad (3.16)$$

η_t 是 DtrBoost 中提出的步长，可用线性方法计算。在迭代过程的最后阶段，对所有的样本权重进行更新，更新公式与 DtrBoost 一致，增加目标域中相同分布的样本权重，减少源域中不同分布的样本权重。目标域样本权重更新公式如下。

$$\begin{aligned} N_{t+1}^{ta}(i) &\leftarrow \frac{\psi'(1 - y_i \sum_{j=1}^N \tau_{t,j} h_j(x_i))}{v_{t+1}^{ta}}, \\ 1 \leq i \leq n \end{aligned} \quad (3.17)$$

各源域样本的更新公式如下：

$$N_{t+1}^{dk}(i) \leftarrow N_t^{dk}(i) \delta^{\left| \left(\sum_{j=1}^N \tau_{t,j} h_j(x_i) \right) - r(x_i) \right|},$$

$$n+1 \leq i \leq n + \sum_{k=1}^p m_k \quad (3.18)$$

其中， $\delta = e^{-0.5 * \log(1 + 2 \ln \sum_{k=1}^p m_k / T^{(1/2)})}$ ， $r(x_i)$ 表示 x_i 的真实类别。在经过了 T 次训练之后，模型使用集成的决策函数完成测试集样本的输出。

$$T = \sum_{j=1}^N (\tau_{T,j} - \tau_{T/2,j}) h_j \quad (3.19)$$

决策函数只使用了 $T/2$ 次迭代之后的分类器权重，在 $T/2$ 次迭代之前，源域中不同分布的数据的权重没有减少到非常小的数量。与目标域学习梯度方向相比，它将在很大程度上阻碍学习相同的梯度方向。因此，使用 $T/2$ 次迭代后的分类器权重可以提高决策函数输出的准确性。多源域优化权重的迁移学习框架如图 3-1 所示，在这里展示的是两个源域迁移情况下前 4 个迭代过程，分类器数目为 3。三角形样本为目标域样本，菱形与正方形表示两个源域的样本，样本的大小表示权重的大小。每次迭代后，都会增加错误分类的目标域数据的权重，降低错误分类的源域数据的权重。在下一轮迭代学习时，目标域的错误分类的样本权重增加会使得分类器的权重错误率增加，模型在选择最佳分类器的时候会选择对该样本分类情况更好的分类器。相反的，降低错误分类的源域数据的权重，使得模型选择最佳分类器的时候降低对于源域分类错误的情况的考察，从而变相地提高了目标域分类情况的重要性。基于这两个样本权重更新策略，使得 MtrBoost 算法可以逐步的提高目标域的分类准确性。

3.3.3 收敛性分析

下面论证本文提出的 MtrBoost 算法的收敛性，DtrBoost 算法是个单源域的迁移学习框架，而 MtrBoost 算法是一个多源域的迁移学习模型。

定理 1 $\exists T_0, t \geq T_0, |Q_w^t(\tau_{t-1,j}, \mu_j) - Q_{ta}^t(\tau_{t-1,j}, \mu_j)| \leq (1/2) |\sum_{k=1}^p (2\kappa_{t,j}^{dk} - 1)(\omega_k \exp(-(1 - \delta)\alpha) / m_k) - (2\kappa_{t,j}^{ta} - 1)v_t^{ta} / n|$ 。

与 DtrBoost 算法一致，本文设置 $T_0 = T/2$ ， $\delta = \exp(-0.5 * \log(1 + 2 \ln \sum_{k=1}^p m_k / T^{(1/2)}))$ ，为了方便简明地表述定理 1 的证明过程，本文将 $|Q_w^t(\tau_{t-1,j}, \mu_j) - Q_{ta}^t(\tau_{t-1,j}, \mu_j)|$ 简写为 $|Q_w^t - Q_{ta}^t|$ ，下面给出证明过程。

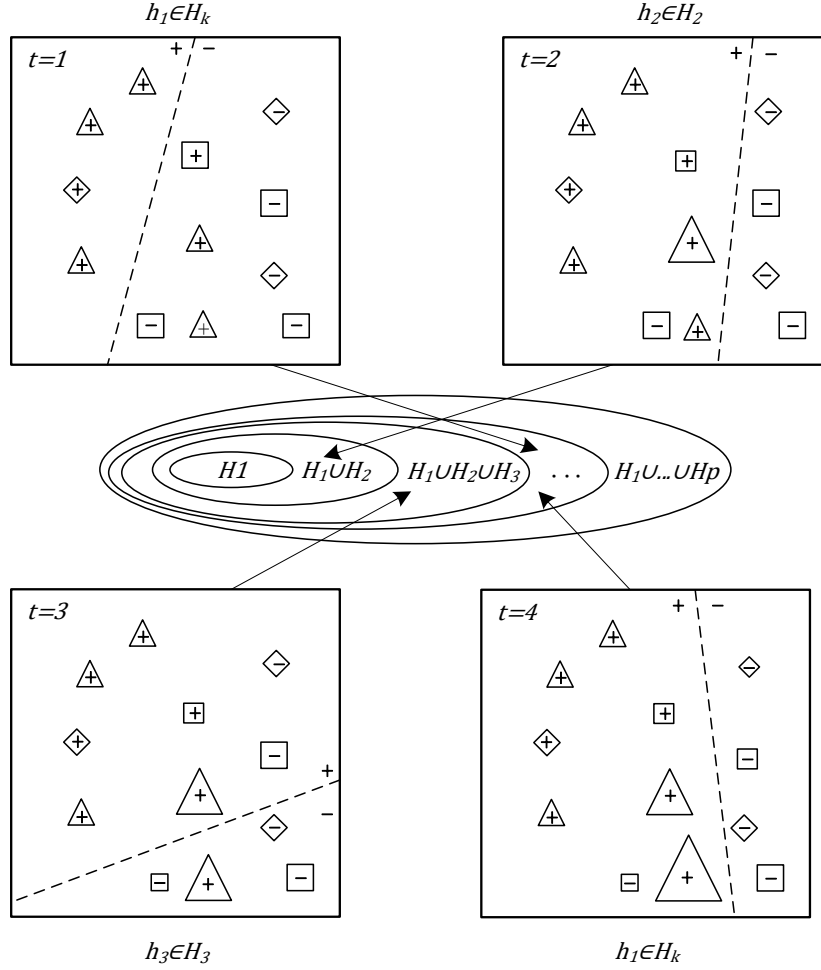


图 3-1 多源域优化权重的迁移学习框架

根据式 (3.13)、(3.15) 可以得到以下公式。

$$\begin{aligned}
 |Q_w^t - Q_{ta}^t| &= \left| \sum_{k=1}^p \frac{\omega_k}{2m_k} \sum_{i=w(k)+1}^{w(k)+m_k} y_i h_j(x_i) \psi'(1 - y_i g_{t-1}(x_i)) \right. \\
 &\quad \left. - \frac{1}{2n} \sum_{i=1}^n y_i h_j(x_i) \psi'(1 - y_i g_{t-1}(x_i)) \right| \\
 &= \frac{1}{2} \left| \sum_{k=1}^p (2\kappa_{t,j}^{dk} - 1) \frac{\omega_k v_t^{dk}}{m_k} - (2\kappa_{t,j}^{ta} - 1) \frac{v_t^{ta}}{n} \right|
 \end{aligned} \tag{3.20}$$

根据 DtrBoost 算法的引理 1, 对于 $\forall \kappa, \exists T_0, t \geq T_0, v_t^{dk} \leq \exp(-(1-\delta)\alpha)$, 其中 α 是一个在 $(t \geq T_0)$ 次迭代后收敛的小数目。可以得到下式。

$$\begin{aligned}
 |Q_w^t - Q_{ta}^t| &= \frac{1}{2} \left| \sum_{k=1}^p (2\kappa_{t,j}^{dk} - 1) \frac{\omega_k v_t^{dk}}{m_k} - (2\kappa_{t,j}^{ta} - 1) \frac{v_t^{ta}}{n} \right| \\
 &\leq \frac{1}{2} \left| \sum_{k=1}^p (2\kappa_{t,j}^{dk} - 1) \frac{\omega_k \exp(-(1-\delta)\alpha)}{m_k} - (2\kappa_{t,j}^{ta} - 1) \frac{v_t^{ta}}{n} \right|
 \end{aligned} \tag{3.21}$$

由于 $v_t^{ia} = \sum_{i=1}^n 1/(1+e^{y_i g_{t-1}(x_i)}) < n$ ，因此 v_t^{ia}/n 收敛。 ω_k 的取值范围为 $[0,1]$ ， $\kappa_{t,j}^{dk}$ ， $\kappa_{t,j}^{ia}$ 的取值范围 $[0,1/2]$ 。在单个源域上 $(2\kappa_{t,j}^{dk}-1)\omega_k \exp(-(1-\delta)\alpha)/m_k$ 收敛，所以多个源域求和后的 $\sum_{k=1}^p (2\kappa_{t,j}^{dk}-1) \cdot \omega_k \exp(-(1-\delta)\alpha)/m_k$ 也收敛。因此 $|Q_w^t - Q_{ia}^t|$ 收敛，MtrBoost 算法收敛性得证。

3.3.4 伪代码和复杂度分析

下面提供 MtrBoost 算法的算法描述和复杂度分析。

算法 1 MtrBoost 的算法描述

输入：目标域训练集 L_s ，源域训练集 $L_{d1}, L_{d2}, \dots, L_{dp}$ ，训练迭代次数 M 。

初始化： $H = \emptyset$ ， $N^{ia}(1) = \dots = N^{ia}(n) = N^{d1}(n+1) = \dots = N^{dp}(n+m_1+\dots+m_p) = 1/(n+m_1+\dots+m_p)$ 。

步骤：

Step 1: **for** $t \leftarrow 1$ **to** M **do**

Step 1.1: 根据式 (3.13) 计算全局梯度 $d_{t,j}^w$ ，并在训练集上训练分类器假设 h_t 。

Step 1.2: **for** h_t **in** H **do**

Step 1.2.1: 根据式 (3.15) 计算目标域梯度 $d_{t,j}^{ia}$

Step 1.3: 选择目标域梯度最大的分类器 $q = \operatorname{argmax}_j |d_{t,j}^{ia}|, j \in [1, N]$

Step 1.4: 更新目标域梯度最大分类器权重 $\tau_t = \tau_{t-1} + \eta_t \mu_q$ 。

Step 1.5: **for** $i \leftarrow 1$ **to** n **do**

Step 1.5.1: 根据式 (3.17) 更新每个目标域样本权重。

Step 1.6: **for** $i \leftarrow n+1$ **to** $m_1+\dots+m_p$ **do**

Step 1.6.1: 根据式 (3.18) 更新每个源域样本权重。

Step 2: 输出训练好的集成决策函数 (3.19)。

算法的过程可以分为两个部分，第一个部分是 Step 1 处的基分类器的迭代训练过程，该部分算法时间复杂度记为 $T_1(n)$ 。第二个部分是 Step 2 处的决策函数的集成输出过程，该部分算法时间复杂度记为 $T_2(n)$ 。在第一个训练过程中又可以分为分类器训练和训练样本权重更新两个小的步骤，分别位于 Step 1.1 至 Step 1.3，Step 1.4 至 Step 1.6。这两个步骤时间复杂度分别记为 $T_{1.1}(n), T_{1.2}(n)$ 。设目标域训练集 L_s 的样本数目为 n ，源域训练集 $L_{d1}, L_{d2}, \dots, L_{dp}$ 的样本数目分别为 $m_1, m_2, \dots, m_p$ ，训练批次数目为 M ，分类器集合数目为 N 。

执行 Step 1.1 步骤的时间复杂度为 $O(n + \sum_{k=1}^p m_k)$ ，又因为外层总共训练迭代 M 次，

所以总体该步骤的时间复杂度为 $O(M(n + \sum_{k=1}^p m_k))$ 。执行 Step 1.2 步骤时分类器数目具有不定性，因此遍历分类器的循环次数也不确定，分类器数目 N 范围为 $1 \leq N \leq t$ 。又因为外层总共训练迭代 M 次，执行 Step 1.2.1 步骤的时间复杂度范围为 $[O(nM), O(nM^2)]$ ，Step 1.3 处选择目标域梯度最大的分类器，需要遍历计算后的梯度值数组，在外层也训练迭代 M 次，因此时间复杂度范围是 $[O(M), O(M^2)]$ 。所以 $T_{1.1}(n)$ 的时间复杂度最小值为 $O(M(n + \sum_{k=1}^p m_k) + (n+1)M)$ ，时间复杂度最大值为 $O(M(n + \sum_{k=1}^p m_k) + (n+1)M^2)$ 。

步骤 Step 1.4 更新目标域梯度最大分类器权重的时间复杂度为 $O(1)$ ，又因为外层总共训练迭代 M 次，所以执行 Step 1.4 步骤的时间复杂度为 $O(M)$ 。Step 1.5 更新目标域样本权重，目标域样本数目为确定的 n ，每一次迭代时需要遍历每个分类器，但是分类器数目不是固定值，取值范围为 $[1, M]$ ，所以执行 Step 1.5 更新目标域样本权重的时间复杂度最小值为 $O(nM)$ ，最大值为 $O(nM^2)$ 。执行 Step 1.6 步骤，在每次迭代时也需要遍历各个分类器，计算分类器训练集上与样本真实值的误差累计和。分类器数目不是固定值，取值范围也是为 $[1, M]$ ，更新源域样本权重的时间复杂度最小值为 $O(M \sum_{k=1}^p m_k)$ ，最大值为 $O(M^2 \sum_{k=1}^p m_k)$ 。假设目标域样本数目少于各源域样本数目， $T_{1.2}(n)$ 最小值为 $O((n+1)M + M \sum_{k=1}^p m_k) \approx O(M \sum_{k=1}^p m_k)$ ， $T_{1.2}(n)$ 最大值为 $O(M + nM^2 + M^2 \sum_{k=1}^p m_k) \approx O(M^2 \sum_{k=1}^p m_k)$ 。

因最终的分类型数目不固定，第二部分执行集成决策函数的时间复杂度范围为 $O(1) \leq T_2(n) \leq O(M/2)$ 。综上，总体时间复杂度最小值为 $O(M(n + \sum_{k=1}^p m_k) + M \sum_{k=1}^p m_k + 1 + nM + M) \approx O(M \sum_{k=1}^p m_k)$ ，时间复杂度最大值为 $O(M(n + \sum_{k=1}^p m_k) + (n+1)M^2 + M^2 \sum_{k=1}^p m_k + M/2) \approx O(M^2 \sum_{k=1}^p m_k)$ 。由于 $n \ll m$ ，所以 DtrBoost 算法的时间复杂度最小值为 $O(M(n+m) + Mm + 1 + nM + M) = O(2(m+n+1)M) \approx O(mM)$ ，最大值为 $O((n+m)M + (n+1)M^2 + M^2m + M/2) = O((n+m+1/2)M + O((n+m+1/2)M + (n+m+1)M^2)) \approx O(mM^2)$ 其中 M 为迭代训练次数， m 为源域训练样本的数目。MtrBoost 算法与 DtrBoost 算法相比，时间复杂度最小值、最大值均在同一量级。EasyTL^[59]算法的时间复杂度为 $O(Cn^3)$ ，其中 C 为样本分类的类别数目， n 为训练样本的数目。MtrBoost 算法与

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/408073137106006041>