迁移学习简明手册

一点心得体会

版本号：v1.1

王晋东
中国科学院计算技术研究所
tutorial.transferlearning.xyz

2018 年 4 月初稿
2019 年 10 月最新修改

摘 要

迁移学习作为机器学习的一大分支，已经取得了长足的进步。本手册简明地介绍迁移学习的概念与基本方法，并对其中的领域自适应问题中的若干代表性方法进行讲述。最后简要探讨迁移学习未来可能的方向。

本手册编写的目的是帮助迁移学习领域的初学者快速入门并掌握基本方法，为自己的研究和应用工作打下良好基础。

本手册的编写逻辑很简单：是什么——介绍迁移学习；为什么——为什么要用迁移学习、为什么能用；怎么办——如何进行迁移 (迁移学习方法)。其中，是什么和为什么解决概念问题，这是一切的前提；怎么办是我们的重点，也占据了最多的篇幅。为了最大限度地方便初学者，我们还特别编写了一章上手实践，直接分享实现代码和心得体会。

推荐语

看了王晋东同学的“迁移学习小册子”，点三个赞！迁移学习被认为是机器学习的下一个爆点，但介绍迁移学习的文章却很有限。这个册子深入浅出，既回顾了迁移学习的发展历史，又囊括了迁移学习的最新进展。语言流畅，简明通透。应该对机器学习的入门和提高都有很大帮助！

——杨强 (迁移学习权威学者，香港科技大学教授，IJCAI president, AAAI/ACM fellow)

目录

写在前面

一直以来都有这样的愿望：无论学习什么知识，总是希望可以快速准确地找到对应的有价值资源进行学习。我相信我们每个人都梦寐以求。然而，越来越多的学科，尤其是我目前从事的计算机科学、人工智能领域，当下正在飞速地发展着。太多的新知识都难以事半功倍地找到快速入手的教程。庄子曰：“吾生也有涯，而知也无涯。以有涯随无涯，殆已。”

我只是迁移学习领域一个很普通的博士生，也同样经历了由“一问三不知”到“稍稍理解”的艰难过程。我在 2016 年初入门迁移学习之时，迁移学习这个概念还未曾像今天一样炙手可热。当时所能找到的学习资源只有两种：别人已发表的论文和已做过的演讲。这些还是不够简单、不够直观。我需要从如此众多的材料中不断归纳，才能站在博士研究的那个圈子的边缘，以便将来可以做出一点点贡献，往圆圈外突破一点点。

相信不只是我，任何一个刚刚入门的学习者都会经历此过程。

“沉舟侧畔千帆过，病树前头万木春。”

己所不欲，勿施于人。正是因为我在初学之时也经历过如此沮丧的时期，我才在 Github 上对迁移学习进行了整理归纳，在知乎网上以“王晋东不在家”为名分享自己对于迁移学习和机器学习的理解和教训、在线上线下与大家讨论相关的问题。很欣慰的是，这些免费开放的资源或多或少地，帮助到了一些初学者，使他们更快速地步入迁移学习之门。

但这些还是不太够。Github 上的资源模式已经固定，目前主要是进行日常更新，不断加入新的论文和代码。目前还是缺乏一个人人都能上手的初学者教程。也只一次，有读者提问有没有相关的入门教程，能真正从 0 到 1 帮助初学者进行入门。

最近，南京大学博士 (现任旷视科技南京研究院负责人) 魏秀参学长写了一本《解析卷积神经网络—深度学习实践手册》，给很多深度学习的初学者提供了帮助。受他的启发，我也决定将自己在迁移学习领域的一些学习心得体会整理成一本手册，免费进行分享。希望能借此方式，帮助更多的初学者。我们不谈风月，只谈干货。

我不是大佬，我也是迁移学习路上的一名小学生。迁移学习领域比我做的好的同龄人太多了。因此，不敢谈什么指导。所有的目的都仅为分享。

本手册在互联网上免费开放。随着作者理解的深入 (以及其他有意者的增补)，本手册肯定会不断修改、越来越好。因此，我打算效仿软件的开发、采取版本更新的方式进行管理。

希望未来可以有更多的有志之士加入，让我们的教程日渐丰富。

致谢

本手册编写过程中得到了许多人的帮助。在此对他们表示感谢。

感谢我的导师、中国科学院计算技术研究所的陈益强研究员。是他一直以来保持着对我的信心，相信我能做出好的研究成果，不断鼓励我，经常与我讨论以明确问题，才有了今天的我。陈老师给我提供了优良的实验环境。我一定会更加努力地科研，做出更多更好的研究成果。

感谢香港科技大学计算机系的杨强教授。杨教授作为迁移学习领域国际泰斗，经常不厌其烦地回答我一些研究上的问题。能够得到杨教授的指导，是我的幸运。希望我能在杨教授带领下，做出更踏实的研究成果。

感谢新加坡南洋理工大学的于涵老师。作为我论文的共同作者，于老师认真的写作态度、对论文的把控能力是我一直学习的榜样。于老师还经常鼓励我，希望可以和于老师有着更多合作，发表更好的文章。

感谢清华大学龙明盛助理教授。龙老师在迁移学习领域发表了众多高质量的研究成果，是我入门时学习的榜样。龙老师还经常对我的研究给予指导。希望有机会可以真正和龙老师合作。

感谢美国伊利诺伊大学芝加哥分校的 Philip S. Yu 教授对我的指导和鼓励。

感谢新加坡 A*STAR 的郝书吉老师。我博士生涯的发表的第一篇论文是和郝老师合作完成的。正是有了第一篇论文被发表，才增强了我的自信，在接下来的研究中放平心态。

感谢我的好基友、西安电子科技大学博士生段然同学和我的同病相怜，让我们可以一起吐槽读博生活。

感谢我的室友沈建飞、以及实验室同学的支持。

感谢我的知乎粉丝和所有交流过迁移学习的学者对我的支持。

最后感谢我的女友和父母对我的支持。

本手册中出现的插图，绝大多数来源于相应论文的配图。感谢这些作者做出的优秀的研究成果。希望我能早日作出可以比肩的研究。

说明

本手册的编写目的是帮助迁移学习领域的初学者快速进行入门。我们尽可能绕开那些非常理论的概念，只讲经验方法。我们还配有多方面的代码、数据、论文资料，最大限度地方便初学者。

本手册的方法部分，关注点是近年来持续走热的领域自适应 (Domain Adaptation) 问题。迁移学习还有其他众多的研究领域。由于作者研究兴趣所在和能力所限，对其他部分的研究只是粗略介绍。非常欢迎从事其他领域研究的读者提供内容。

本手册的每一章节都是自包含的，因此，初学者不必从头开始阅读每一部分。直接阅读自己需要的或者自己感兴趣的部分即可。本手册每一章节的信息如下：

第 1 章介绍了迁移学习的概念，重点解决什么是迁移学习、为什么要进行迁移学习这两个问题。

第 2 章介绍了迁移学习的研究领域。

第 3 章介绍了迁移学习的应用领域。

第 4 章是迁移学习领域的一些基本知识，包括问题定义，域和任务的表示，以及迁移学习的总体思路。特别地，我们提供了较为全面的度量准则介绍。度量准则是迁移学习领域重要的工具。

第 5 章简要介绍了迁移学习的四种基本方法，即基于样本迁移、基于特征迁移、基于模型迁移、基于关系迁移。

第 6 章到第 8 章，介绍了领域自适应的 3 大类基本的方法，分别是：数据分布自适应法、特征选择法、子空间学习法。

第 9 章重点介绍了目前主流的深度迁移学习方法。

第 10 章提供了简单的上手实践教程。

第 11 章对迁移学习进行了展望，提出了未来几个可能的研究方向。

第 12 章是对全手册的总结。

第 13 章是附录，提供了迁移学习领域相关的学习资源，以供读者参考。

由于作者水平有限，不足和错误之处，敬请不吝批评指正。

手册的相关资源：

我们在 Github 上持续维护了迁移学习的资源仓库，包括论文、代码、文档、比赛等，请读者持续关注：https://github.com/jindongwang/transferlearning，配合本手册更香哦！

网站 (内含勘误表)：http://t.cn/RmasEFe

开发维护地址：http://github.com/jindongwang/transferlearning-tutorial

作者的联系方式：

邮箱: jindongwang@outlook.com，知乎: 王晋东不在家。

微博: 秦汉日记，个人网站:http://jd92.wang。

1 迁移学习基本概念

1.1 引子

冬末春初，北京的天气渐渐暖了起来。这是一句再平常不过的气候描述。对于我们在北半球生活的人来说，这似乎是一个司空见惯的现象。北京如此，纽约如此，东京如此，巴黎也如此。然而此刻，假如我问你，阿根廷的首都布宜诺斯艾利斯，天气如何？稍稍有点地理常识的人就该知道，阿根廷位于南半球，天气恰恰相反：正是夏末秋初的时候，天气渐渐凉了起来。

我们何以根据北京的天气来推测出纽约、东京和巴黎的天气？我们又何以不能用相同的方式来推测阿根廷的天气？

答案显而易见：因为它们的地理位置不同。除去阿根廷在南半球之外，其他几个城市均位于北半球，故而天气变化相似。

我们可以利用这些地点地理位置的相似性和差异性，很容易地推测出其他地点的天气。这样一个简单的事实，就引出了我们要介绍的主题：迁移学习。

1.2 迁移学习的概念

迁移学习，顾名思义，就是要进行迁移。放到我们人工智能和机器学习的学科里讲，迁移学习是一种学习的思想和模式。

我们都对机器学习有了基本的了解。机器学习是人工智能的一大类重要方法，也是目前发展最迅速、效果最显著的方法。机器学习解决的是让机器自主地从数据中获取知识，从而应用于新的问题中。迁移学习作为机器学习的一个重要分支，侧重于将已经学习过的知识迁移应用于新的问题中。

迁移学习的核心问题是，找到新问题和原问题之间的相似性，才可以顺利地实现知识的迁移。比如在我们一开始说的天气问题中，那些北半球的天气之所以相似，是因为它们的地理位置相似；而南北半球的天气之所以有差异，也是因为地理位置有根本不同。

其实我们人类对于迁移学习这种能力，是与生俱来的。比如，我们如果已经会打乒乓球，就可以类比着学习打网球。再比如，我们如果已经会下中国象棋，就可以类比着下国际象棋。因为这些活动之间，往往有着极高的相似性。生活中常用的“举一反三”、“照猫画虎”就很好地体现了迁移学习的思想。

回到我们的问题中来。我们用更加学术更加机器学习的语言来对迁移学习下一个定义。迁移学习，是指利用数据、任务、或模型之间的相似性，将在旧领域学习过的模型，应用于新领域的一种学习过程。

迁移学习最权威的综述文章是香港科技大学杨强教授团队的 A survey on transfer learning [Pan and Yang, 2010]。

图 1简要表示了一个迁移学习过程。图 2给出了生活中常见的迁移学习的例子。

图 1: 迁移学习示意图

图 2: 迁移学习的例子

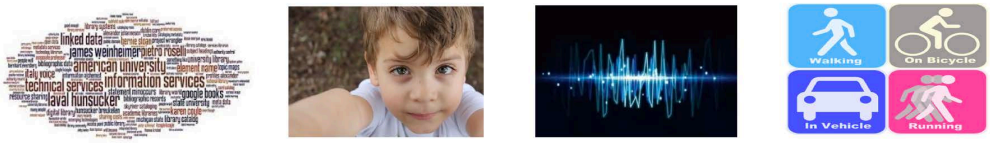
值得一提的是，新华社报道指出，迁移学习是中国领先于世界的少数几个人工智能领域之一 [xinhua, 2016]。中国的人工智能赶超的机会来了！

1.3 为什么需要迁移学习？

了解了迁移学习的概念之后，紧接着还有一个非常重要的问题：迁移学习的目的是什么？或者说，为什么要用迁移学习？

我们把原因概括为以下四个方面:

1. 大数据与少标注之间的矛盾。



文本　图片及视频　音频　行为

图 3: 多种多样的数据来源

我们正处在一个大数据时代，每天每时，社交网络、智能交通、视频监控、行业物流等，都产生着海量的图像、文本、语音等各类数据。数据的增多，使得机器学习和深度学习模型可以依赖于如此海量的数据，持续不断地训练和更新相应的模型，使得模型的性能越来越好，越来越适合特定场景的应用。然而，这些大数据带来了严重的问题：总是缺乏完善的数据标注。

众所周知，机器学习模型的训练和更新，均依赖于数据的标注。然而，尽管我们可以获取到海量的数据，这些数据往往是很初级的原始形态，很少有数据被加以正确的人工标注。数据的标注是一个耗时且昂贵的操作，目前为止，尚未有行之有效的方式来解决这一问题。这给机器学习和深度学习的模型训练和更新带来了挑战。反过来说，特定的领域，因为没有足够的标定数据用来学习，使得这些领域一直不能很好的发展。

2. 大数据与弱计算之间的矛盾。

大数据，就需要大设备、强计算能力的设备来进行存储和计算。然而，大数据的大计算能力，是” 有钱人” 才能玩得起的游戏。比如 Google，Facebook，Microsoft，这些巨无霸公司有着雄厚的计算能力去利用这些数据训练模型。例如，ResNet 需要很长的时间进行训练。Google TPU 也都是有钱人的才可以用得起的。

绝大多数普通用户是不可能具有这些强计算能力的。这就引发了大数据和弱计算之间的矛盾。在这种情况下，普通人想要利用这些海量的大数据去训练模型完成自己的任务，基本上不太可能。那么如何让普通人也能利用这些数据和模型？

3. 普适化模型与个性化需求之间的矛盾。

机器学习的目标是构建一个尽可能通用的模型，使得这个模型对于不同用户、不同设备、不同环境、不同需求，都可以很好地进行满足。这是我们的美好愿景。这就是要尽可能

图 4: 大数据与强计算能力

地提高机器学习模型的泛化能力，使之适应不同的数据情形。基于这样的愿望，我们构建了多种多样的普适化模型，来服务于现实应用。然而，这只能是我们竭尽全力想要做的，目前却始终无法彻底解决的问题。人们的个性化需求五花八门，短期内根本无法用一个通用的模型去满足。比如导航模型，可以定位及导航所有的路线。但是不同的人有不同的需求。比如有的人喜欢走高速，有的人喜欢走偏僻小路，这就是个性化需求。并且，不同的用户，通常都有不同的隐私需求。这也是构建应用需要着重考虑的。

所以目前的情况是，我们对于每一个通用的任务都构建了一个通用的模型。这个模型可以解决绝大多数的公共问题。但是具体到每个个体、每个需求，都存在其唯一性和特异性，一个普适化的通用模型根本无法满足。那么，能否将这个通用的模型加以改造和适配，使其更好地服务于人们的个性化需求？

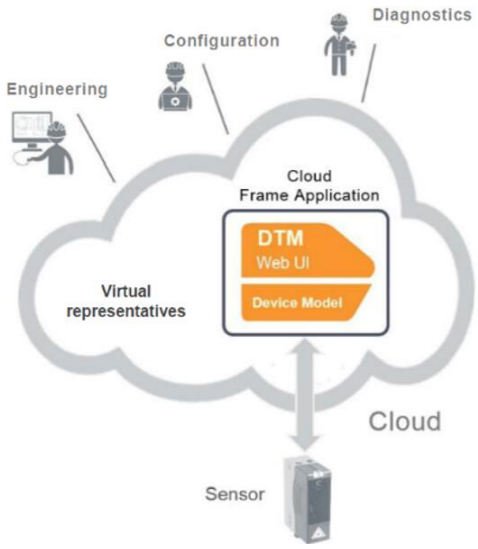


图 5: 普适化模型与个性化需求

4. 特定应用的需求。

机器学习已经被广泛应用于现实生活中。在这些应用中，也存在着一些特定的应用，它们面临着一些现实存在的问题。比如推荐系统的冷启动问题。一个新的推荐系统，没有足够的用户数据，如何进行精准的推荐？一个崭新的图片标注系统，没有足够的标签，如何进行精准的服务？现实世界中的应用驱动着我们去开发更加便捷更加高效的机器学习方法来加以解决。

上述存在的几个重要问题，使得传统的机器学习方法疲于应对。迁移学习则可以很好

图 6: 特定应用需求：冷启动

地进行解决。那么，**迁移学习是如何进行解决的呢？**

1. 大数据与少标注：迁移数据标注

单纯地凭借少量的标注数据，无法准确地训练高可用度的模型。为了解决这个问题，我们直观的想法是：多增加一些标注数据不就行了？但是不依赖于人工，如何增加标注数据？

利用迁移学习的思想，我们可以寻找一些与目标数据相近的有标注的数据，从而利用这些数据来构建模型，增加我们目标数据的标注。

2. 大数据与弱计算：模型迁移

不可能所有人都有能力利用大数据快速进行模型的训练。利用迁移学习的思想，我们可以将那些大公司在大数据上训练好的模型，迁移到我们的任务中。针对于我们的任务进行微调，从而我们也可以拥有在大数据上训练好的模型。更进一步，我们可以将这些模型针对我们的任务进行自适应更新，从而取得更好的效果。

3. 普适化模型与个性化需求：自适应学习

为了解决个性化需求的挑战，我们利用迁移学习的思想，进行自适应的学习。考虑到不同用户之间的相似性和差异性，我们对普适化模型进行灵活的调整，以便完成我们的任务。

4. 特定应用的需求：相似领域知识迁移

为了满足特定领域应用的需求，我们可以利用上述介绍过的手段，从数据和模型方法上进行迁移学习。

表1概括地描述了迁移学习的必要性。

表 1: 迁移学习的必要性

矛盾	传统机器学习	迁移学习
大数据与少标注	增加人工标注，但是昂贵且耗时	数据的迁移标注
大数据与弱计算	只能依赖强大计算能力，但是受众少	模型迁移
普适化模型与个性化需求	通用模型无法满足个性化需求	模型自适应调整
特定应用	冷启动问题无法解决	数据迁移

1.4 与已有概念的区别和联系

迁移学习并不是一个横空出世的概念，它与许多已有的概念都有些联系，但是也有着一些区别。我们在这里汇总一些与迁移学习非常接近的概念，并简述迁移学习与它们的区别和联系。

1. 迁移学习 VS 传统机器学习：

迁移学习属于机器学习的一类，但它在如下几个方面有别于传统的机器学习 (表 2)：

2. 迁移学习 VS 多任务学习：

表 2: 传统机器学习与迁移学习的区别

比较项目	传统机器学习	迁移学习
数据分布	训练和测试数据服从相同的分布	训练和测试数据服从不同的分布
数据标注	需要足够的数据标注来训练模型	不需要足够的数据标注
模型	每个任务分别建模	模型可以在不同任务之间迁移

多任务学习指多个相关的任务一起协同学习；迁移学习则强调知识由一个领域迁移到另一个领域的过程。迁移是思想，多任务是其中的一个具体形式。

3. 迁移学习 VS 终身学习：

终身学习可以认为是序列化的多任务学习，在已经学习好若干个任务之后，面对新的任务可以继续学习而不遗忘之前学习的任务。迁移学习则侧重于模型的迁移和共同学习。

4. 迁移学习 VS 领域自适应：

领域自适应问题是迁移学习的研究内容之一，它侧重于解决特征空间一致、类别空间一致，仅特征分布不一致的问题。而迁移学习也可以解决上述内容不一致的情况。

5. 迁移学习 VS 增量学习：

增量学习侧重解决数据不断到来，模型不断更新的问题。迁移学习显然和其有着不同之处。

6. 迁移学习 VS 自我学习：

自我学习指的是模型不断地从自身处进行更新，而迁移学习强调知识在不同的领域间进行迁移。

7. 迁移学习 VS 协方差漂移

协方差漂移指数据的边缘概率分布发生变化。领域自适应研究问题解决的就是协方差漂移现象。

1.5 负迁移

我们都希望迁移学习能够比较顺利地进行，我们得到的结果也是满足我们要求的，皆大欢喜。然而，事情却并不总是那么顺利。这就引入了迁移学习中的一个负面现象，也就是所谓的**负迁移**。

用我们熟悉的成语来描述：如果说成功的迁移学习是“举一反三”、“照猫画虎”，那么负迁移则是“东施效颦”。东施已经模仿西施捂着胸口皱着眉头，为什么她还是那么丑?

要理解负迁移，首先要理解什么是迁移学习。迁移学习指的是，利用数据和领域之间存在的相似性关系，把之前学习到的知识，应用于新的未知领域。迁移学习的核心问题是，找到两个领域的相似性。找到了这个相似性，就可以合理地利用，从而很好地完成迁移学习任务。比如，之前会骑自行车，要学习骑摩托车，这种相似性指的就是自行车和摩托车之间的相似性以及骑车体验的相似性。这种相似性在我们人类看来是可以接受的。

所以，如果这个相似性找的不合理，也就是说，两个领域之间不存在相似性，或者基本不相似，那么，就会大大损害迁移学习的效果。还是拿骑自行车来说，你要拿骑自行车的经验来学习开汽车，这显然是不太可能的。因为自行车和汽车之间基本不存在什么相似性。所以，这个任务基本上完不成。这时候，我们可以说出现了**负迁移 (Negative Transfer)**。

所以，为什么东施和西施做了一样的动作，反而变得更丑了？因为东施和西施之间压根就不存在相似性。

迁移学习领域权威学者、香港科技大学杨强教授发表的迁移学习的综述文章 A survey on transfer learning [Pan and Yang, 2010] 给出了负迁移的一个定义：

负迁移指的是，在源域上学习到的知识，对于目标域上的学习产生负面作用。

文章也引用了一些经典的解决负迁移问题的文献。但是普遍较老，这里就不说了。

所以，产生负迁移的原因主要有：

- 数据问题：源域和目标域压根不相似，谈何迁移？
- 方法问题：源域和目标域是相似的，但是，迁移学习方法不够好，没找到可迁移的成分。

负迁移给迁移学习的研究和应用带来了负面影响。在实际应用中，找到合理的相似性，并且选择或开发合理的迁移学习方法，能够避免负迁移现象。

最新的研究成果

随着研究的深入，已经有新的研究成果在逐渐克服负迁移的影响。杨强教授团队 2015 在数据挖掘领域顶级会议 KDD 上发表了传递迁移学习文章 Transitive transfer learning [Tan et al., 2015]，提出了传递迁移学习的思想。传统迁移学习就好比是踩着一块石头过河，传递迁移学习就好比是踩着连续的两块石头。

更进一步，杨强教授团队在 2017 年人工智能领域顶级会议 AAAI 上发表了远领域迁移学习的文章 Distant domain transfer learning [Tan et al., 2017]，可以用人脸来识别飞机！这就好比是踩着一连串石头过河。这些研究的意义在于，传统迁移学习只有两个领域足够相似才可以完成，而当两个领域不相似时，传递迁移学习却可以利用处于这两个领域之间的若干领域，将知识传递式的完成迁移。这个是很有意义的工作，可以视为解决负迁移的有效思想和方法。可以预见在未来会有更多的应用前景。

图 7对传递迁移学习给出了简明的示意。

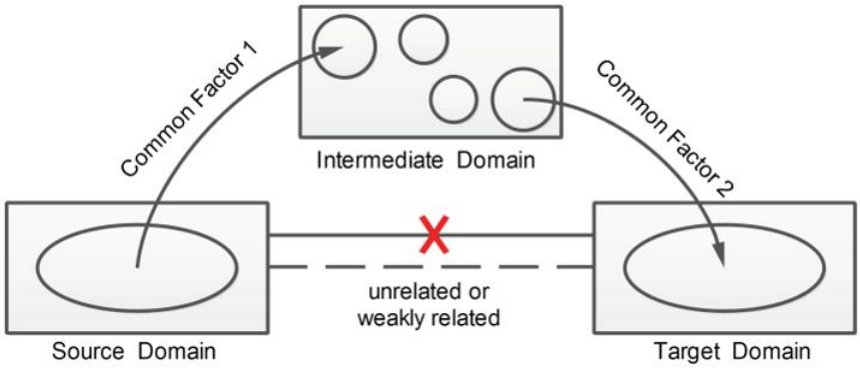


图 7: 传递式迁移学习示意图

2 迁移学习的研究领域

依据目前较流行的机器学习分类方法，机器学习主要可以分为有监督、半监督和无监督机器学习三大类。同理，迁移学习也可以进行这样的分类。需要注意的是，依据的分类准则不同，分类结果也不同。在这一点上，并没有一个统一的说法。我们在这里仅根据目前较流行的方法，对迁移学习的研究领域进行一个大致的划分。

图 8给出了迁移学习的常用分类方法总结。

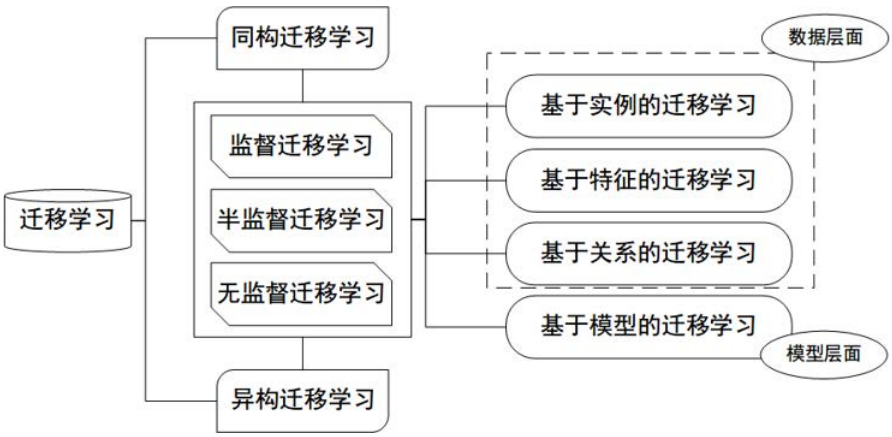


图 8: 迁移学习的研究领域与研究方法分类

大体上讲，迁移学习的分类可以按照四个准则进行：按目标域有无标签分、按学习方法分、按特征分、按离线与在线形式分。不同的分类方式对应着不同的专业名词。当然，即使是一个分类下的研究领域，也可能同时处于另一个分类下。下面我们对这些分类方法及相应的领域作简单描述。

2.1 按目标域标签分

这种分类方式最为直观。类比机器学习，按照目标领域有无标签，迁移学习可以分为以下三个大类：

1. 监督迁移学习 (Supervised Transfer Learning)
2. 半监督迁移学习 (Semi-Supervised Transfer Learning)
3. 无监督迁移学习 (Unsupervised Transfer Learning)

显然，少标签或无标签的问题 (半监督和无监督迁移学习)，是研究的热点和难点。这也是本手册重点关注的领域。

2.2 按学习方法分类

按学习方法的分类形式，最早在迁移学习领域的权威综述文章 [Pan and Yang, 2010] 给出定义。它将迁移学习方法分为以下四个大类：

1. 基于样本的迁移学习方法 (Instance based Transfer Learning)
2. 基于特征的迁移学习方法 (Feature based Transfer Learning)

3. 基于模型的迁移学习方法 (Model based Transfer Learning)

4. 基于关系的迁移学习方法 (Relation based Transfer Learning)

这是一个很直观的分类方式，按照数据、特征、模型的机器学习逻辑进行区分，再加上不属于这三者中的关系模式。

基于实例的迁移，简单来说就是通过权重重用，对源域和目标域的样例进行迁移。就是说直接对不同的样本赋予不同权重，比如说相似的样本，我就给它高权重，这样我就完成了迁移，非常简单非常非常直接。

基于特征的迁移，就是更进一步对特征进行变换。意思是说，假设源域和目标域的特征原来不在一个空间，或者说它们在原来那个空间上不相似，那我们就想办法把它们变换到一个空间里面，那这些特征不就相似了？这个思路也非常直接。这个方法是用得非常多的，一直在研究，目前是感觉是研究最热的。

基于模型的迁移，就是说构建参数共享的模型。这个主要就是在神经网络里面用的特别多，因为神经网络的结构可以直接进行迁移。比如说神经网络最经典的 finetune 就是模型参数迁移的很好的体现。

基于关系的迁移，这个方法用的比较少，这个主要就是说挖掘和利用关系进行类比迁移。比如老师上课、学生听课就可以类比为公司开会的场景。这个就是一种关系的迁移。

目前最热的就是基于特征还有模型的迁移，然后基于实例的迁移方法和他们结合起来使用。

迁移学习方法是本手册的重点。我们在后续的篇幅中介绍。

2.3 按特征分类

按照特征的属性进行分类，也是一种常用的分类方法。这在最近的迁移学习综述 [Weiss et al., 2016] 中给出。按照特征属性，迁移学习可以分为两个大类：

1. 同构迁移学习 (Homogeneous Transfer Learning)

2. 异构迁移学习 (Heterogeneous Transfer Learning)

这也是一种很直观的方式：如果特征语义和维度都相同，那么就是同构；反之，如果特征完全不相同，那么就是异构。举个例子来说，不同图片的迁移，就可以认为是同构；而图片到文本的迁移，则是异构的。

2.4 按离线与在线形式分

按照离线学习与在线学习的方式，迁移学习还可以被分为：

1. 离线迁移学习 (Offline Transfer Learning)

2. 在线迁移学习 (Online Transfer Learning)

目前，绝大多数的迁移学习方法，都采用了离线方式。即，源域和目标域均是给定的，迁移一次即可。这种方式的缺点是显而易见的：算法无法对新加入的数据进行学习，模型也无法得到更新。与之相对的，是在线的方式。即随着数据的动态加入，迁移学习算法也可以不断地更新。

3 迁移学习的应用

迁移学习是机器学习领域的一个重要分支。因此，其应用并不局限于特定的领域。凡是满足迁移学习问题情景的应用，迁移学习都可以发挥作用。这些领域包括但不限于计算机视觉、文本分类、行为识别、自然语言处理、室内定位、视频监控、舆情分析、人机交互等。图 9展示了迁移学习可能的应用领域。

下面我们选择几个研究热点，对迁移学习在这些领域的应用场景作一简单介绍。

图 9: 迁移学习的应用领域概览

3.1 计算机视觉

迁移学习已被广泛地应用于计算机视觉的研究中。特别地，在计算机视觉中，迁移学习方法被称为 Domain Adaptation。Domain adaptation 的应用场景有很多，比如图片分类、图片哈希等。

图 10展示了不同的迁移学习图片分类任务示意。同一类图片，不同的拍摄角度、不同光照、不同背景，都会造成特征分布发生改变。因此，使用迁移学习构建跨领域的鲁棒分类器是十分重要的。

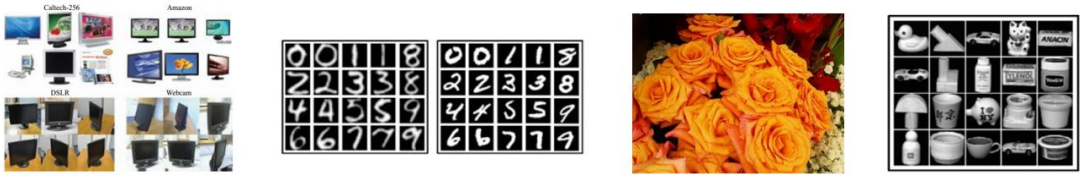


图 10: 迁移学习图片分类任务

计算机视觉三大顶会 (CVPR、ICCV、ECCV) 每年都会发表大量的文章对迁移学习在视觉领域的应用进行介绍。

3.2 文本分类

由于文本数据有其领域特殊性，因此，在一个领域上训练的分类器，不能直接拿来作用到另一个领域上。这就需要用到迁移学习。例如，在电影评论文本数据集上训练好的分类

器，不能直接用于图书评论的预测。这就需要进行迁移学习。图 11是一个由电子产品评论迁移到 DVD 评论的迁移学习任务。

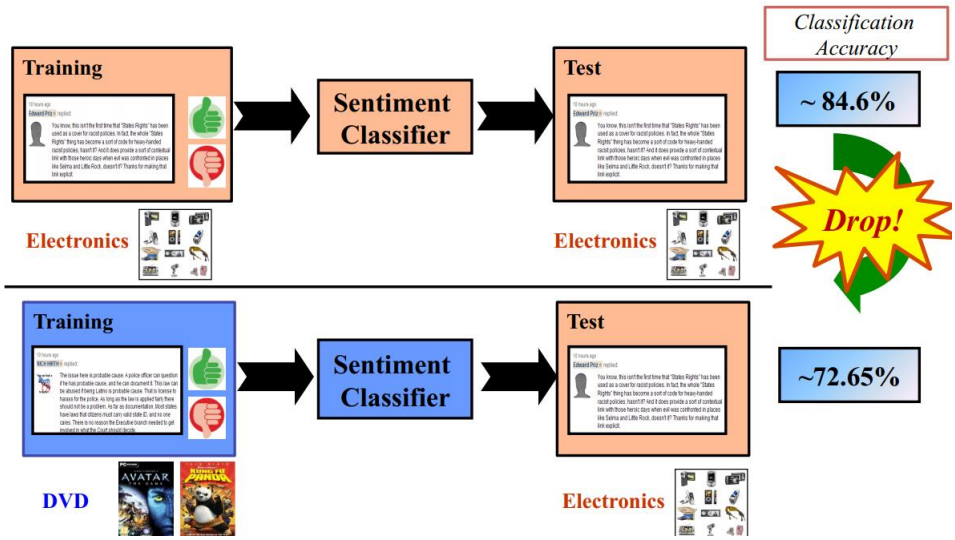


图 11: 迁移学习文本分类任务

文本和网络领域顶级会议 WWW 和 CIKM 每年有大量的文章对迁移学习在文本领域的应用作介绍。

3.3 时间序列

行为识别 (Activity Recognition) 主要通过佩戴在用户身体上的传感器，研究用户的行为。行为数据是一种时间序列数据。不同用户、不同环境、不同位置、不同设备，都会导致时间序列数据的分布发生变化。此时，也需要进行迁移学习。图 12展示了同一用户不同位置的信号差异性。在这个领域，华盛顿州立大学的 Diane Cook 等人在 2013 年发表的关于迁移学习在行为识别领域的综述文章 [Cook et al., 2013] 是很好的参考资料。

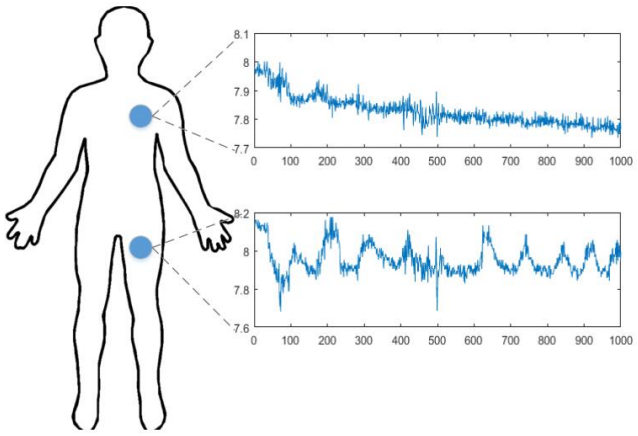


图 12: 不同位置的传感器信号差异示意图

室内定位 (Indoor Location) 与传统的室外用 GPS 定位不同，它通过 WiFi、蓝牙等设备研究人在室内的位置。不同用户、不同环境、不同时刻也会使得采集的信号分布发生变化。图 13展示了不同时间、不同设备的 WiFi 信号变化。

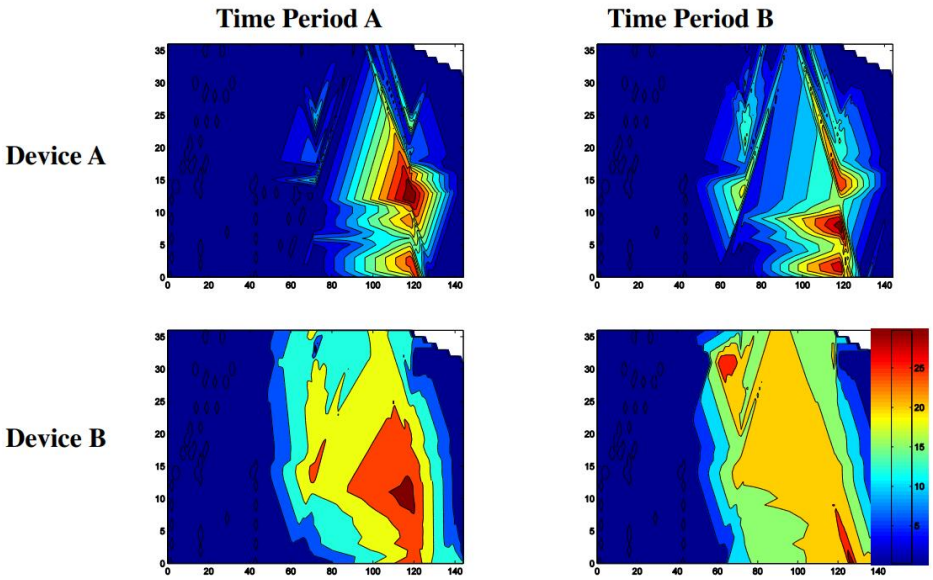


图 13: 室内定位由于时间和设备的变化导致的信号变化

3.4 医疗健康

医疗健康领域的研究正变得越来越重要。不同于其他领域，医疗领域研究的难点问题是，*无法获取足够有效的医疗数据*。在这一领域，迁移学习同样也变得越来越重要。

最近，顶级生物期刊细胞杂志报道了由张康教授领导的广州妇女儿童医疗中心和加州大学圣迭戈分校团队的重磅研究成果：基于深度学习开发出一个能诊断眼病和肺炎两大类疾病的 AI 系统 [Kermany et al., 2018]，准确性匹敌顶尖医生。这不仅是中国研究团队首次在顶级生物医学杂志发表有关医学人工智能的研究成果；也是世界范围内首次使用如此庞大的标注好的高质量数据进行**迁移学习**，并取得高度精确的诊断结果，达到匹敌甚至超越人类医生的准确性；还是全世界首次实现用 AI 精确推荐治疗手段。细胞杂志封面报道了该研究成果。

我们可以预见到的是，迁移学习对于那些不易获取标注数据的领域，将会发挥越来越重要的作用。

4 基础知识

本部分介绍迁移学习领域的一些基本知识。我们对迁移学习的问题进行简单的形式化，给出迁移学习的总体思路，并且介绍目前常用的一些度量准则。本部分中出现的所有符号和表示形式，是以后章节的基础。已有相关知识的读者可以直接跳过。

4.1 迁移学习的问题形式化

迁移学习的问题形式化，是进行一切研究的前提。在迁移学习中，有两个基本的概念：**领域 (Domain)** 和**任务 (Task)**。它们是最基础的概念。定义如下：

4.1.1 领域

领域 (Domain): 是进行学习的主体。领域主要由两部分构成：*数据和生成这些数据的概率分布*。通常我们用花体 $\mathcal{D}$ 来表示一个 domain，用大写斜体 P 来表示一个概率分布。

特别地，因为涉及到迁移，所以对应于两个基本的领域：**源领域 (Source Domain)** 和**目标领域 (Target Domain)**。这两个概念很好理解。源领域就是有知识、有大量数据标注的领域，是我们要迁移的对象；目标领域就是我们最终要赋予知识、赋予标注的对象。知识从源领域传递到目标领域，就完成了迁移。

领域上的数据，我们通常用小写粗体 $\mathbf{x}$ 来表示，它也是向量的表示形式。例如，$\mathbf{x}_i$ 就表示第 i 个样本或特征。用大写的黑体 $\mathbf{X}$ 表示一个领域的数据，这是一种矩阵形式。我们用大写花体 $\mathcal{X}$ 来表示数据的特征空间。

通常我们用小写下标 s 和 t 来分别指代两个领域。结合领域的表示方式，则：$\mathcal{D}_s$ 表示源领域，$\mathcal{D}_t$ 表示目标领域。

值得注意的是，概率分布 P 通常只是一个逻辑上的概念，即我们认为不同领域有不同的概率分布，却一般不给出（也难以给出）P 的具体形式。

4.1.2 任务

任务 (Task): 是学习的目标。任务主要由两部分组成：*标签和标签对应的函数*。通常我们用花体 $\mathcal{Y}$ 来表示一个标签空间，用 $f(\cdot)$ 来表示一个学习函数。

相应地，源领域和目标领域的类别空间就可以分别表示为 $\mathcal{Y}_s$ 和 $\mathcal{Y}_t$。我们用小写 y_s 和 y_t 分别表示源领域和目标领域的实际类别。

4.1.3 迁移学习

有了上面领域和任务的定义，我们就可以对迁移学习进行形式化。

迁移学习 (Transfer Learning): 给定一个有标记的源域 $\mathcal{D}_s = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ 和一个无标记的目标域 $\mathcal{D}_t = \{\mathbf{x}_j\}_{j=n+1}^{n+m}$。这两个领域的数据分布 $P(\mathbf{x}_s)$ 和 $\mathrm{P}(\mathbf{x}_t)$ 不同，即 $P(\mathbf{x}_s) \neq P(\mathbf{x}_t)$。迁移学习的目的就是要借助 $\mathcal{D}_s$ 的知识，来学习目标域 $\mathcal{D}_t$ 的知识 (标签)。

更进一步，结合我们前面说过的迁移学习研究领域，迁移学习的定义需要进行如下的考虑：

(1) 特征空间的异同，即 $\mathcal{X}_s$ 和 $\mathcal{X}_t$ 是否相等。

(2) 类别空间的异同：即 $\mathcal{Y}_s$ 和 $\mathcal{Y}_t$ 是否相等。

(3) 条件概率分布的异同：即 $Q_s(y_s|\mathbf{x}_s)$ 和 $Q_t(y_t|\mathbf{x}_t)$ 是否相等。

结合上述形式化，我们给出**领域自适应 (Domain Adaptation)** 这一热门研究方向的定义：

领域自适应 (Domain Adaptation): 给定一个有标记的源域 $\mathcal{D}_s = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$ 和一个无标记的目标域 $\mathcal{D}_t = \{\mathbf{x}_j\}_{j=n+1}^{n+m}$，假定它们的特征空间相同，即 $\mathcal{X}_s = \mathcal{X}_t$，并且它们的类别空间也相同，即 $\mathcal{Y}_s = \mathcal{Y}_t$ 以及条件概率分布也相同，即 $Q_s(y_s|\mathbf{x}_s) = Q_t(y_t|\mathbf{x}_t)$。但是这两个域的边缘分布不同，即 $P_s(\mathbf{x}_s) \neq P_t(\mathbf{x}_t)$。迁移学习的目标就是，利用有标记的数据 $\mathcal{D}_s$ 去学习一个分类器 $f : \mathbf{x}_t \mapsto \mathbf{y}_t$ 来预测目标域 $\mathcal{D}_t$ 的标签 $\mathbf{y}_t \in \mathcal{Y}_t$.

在实际的研究和应用中，读者可以针对自己的不同任务，结合上述表述，灵活地给出相关的形式化定义。

符号小结

我们已经基本介绍了迁移学习中常用的符号。表 3是一个符号表：

表 3: 迁移学习形式化表示常用符号

符号	含义
下标 s / t	指示源域 / 目标域
$\mathcal{D}_s$ / $\mathcal{D}_t$	源域数据 / 目标域数据
$\mathbf{x}$ / $\mathbf{X}$ / $\mathcal{X}$	向量 / 矩阵 / 特征空间
$\mathbf{y}$ / $\mathcal{Y}$	类别向量 / 类别空间
(n, m) [或 (n_1, n_2) 或 (n_s, n_t)]	(源域样本数, 目标域样本数)
$P(\mathbf{x}_s)$ / $P(\mathbf{x}_t)$	源域数据 / 目标域数据的边缘分布
$Q(\mathbf{y}_s\|\mathbf{x}_s)$ / $Q(\mathbf{y}_t\|\mathbf{x}_t)$	源域数据 / 目标域数据的条件分布
$f(\cdot)$	要学习的目标函数

4.2 总体思路

形式化之后，我们可以进行迁移学习的研究。迁移学习的总体思路可以概括为：开发算法来最大限度地利用有标注的领域的知识，来辅助目标领域的知识获取和学习。

迁移学习的核心是，找到源领域和目标领域之间的**相似性**，并加以合理利用。这种相似性非常普遍。比如，不同人的身体构造是相似的；自行车和摩托车的骑行方式是相似的；国际象棋和中国象棋是相似的；羽毛球和网球的打球方式是相似的。这种相似性也可以理解为**不变量**。以不变应万变，才能立于不败之地。

举一个杨强教授经常举的例子来说明：我们都知道在中国大陆开车时，驾驶员坐在左边，靠马路右侧行驶。这是基本的规则。然而，如果在英国、香港等地区开车，驾驶员是坐在右边，需要靠马路左侧行驶。那么，如果我们从中国大陆到了香港，应该如何快速地适应他们的开车方式呢？诀窍就是找到这里的不变量：不论在哪个地区，驾驶员都是紧靠马路中间。这就是我们这个开车问题中的不变量。

找到相似性 (不变量)，是进行迁移学习的核心。

有了这种相似性后，下一步工作就是，如何度量和利用这种相似性。度量工作的目标有两点：一是很好地度量两个领域的相似性，不仅定性地告诉我们它们是否相似，更定量地给出相似程度。二是以度量为准则，通过我们所要采用的学习手段，增大两个领域之间的相似性，从而完成迁移学习。

一句话总结： 相似性是核心，度量准则是重要手段。

4.3 度量准则

度量不仅是机器学习和统计学等学科中使用的基础手段，也是迁移学习中的重要工具。它的核心就是衡量两个数据域的差异。计算两个向量（点、矩阵）的距离和相似度是许多机器学习算法的基础，有时候一个好的距离度量就能决定算法最后的结果好坏。比如 KNN 分类算法就对距离非常敏感。本质上就是找一个变换使得源域和目标域的距离最小（相似度最大）。所以，相似度和距离度量在机器学习中非常重要。

这里给出常用的度量手段，它们都是迁移学习研究中非常常见的度量准则。对这些准则有很好的理解，可以帮助我们设计出更加好用的算法。用一个简单的式子来表示，度量就是描述源域和目标域这两个领域的距离：

$$DISTANCE(\mathcal{D}_s, \mathcal{D}_t) = \text{DistanceMeasure}(\cdot, \cdot) \tag{4.1}$$

下面我们从距离和相似度度量准则几个方面进行简要介绍。

4.3.1 常见的几种距离

1. 欧氏距离

定义在两个向量 (空间中的两个点) 上：点 $\mathbf{x}$ 和点 $\mathbf{y}$ 的欧氏距离为：

$$d_{Euclidean} = \sqrt{(\mathbf{x}-\mathbf{y})^\top(\mathbf{x}-\mathbf{y})} \tag{4.2}$$

c **2. 闵可夫斯基距离**

Minkowski distance，两个向量（点）的 p 阶距离：

$$d_{Minkowski} = (||\mathbf{x}-\mathbf{y}||^p)^{1/p} \tag{4.3}$$

当 $p=1$ 时就是曼哈顿距离，当 $p=2$ 时就是欧氏距离。

3. 马氏距离

定义在两个向量 (两个点) 上，这两个数据在同一个分布里。点 $\mathbf{x}$ 和点 $\mathbf{y}$ 的马氏距离为：

$$d_{Mahalanobis} = \sqrt{(\mathbf{x}-\mathbf{y})^\top\Sigma^{-1}(\mathbf{x}-\mathbf{y})} \tag{4.4}$$

其中，Σ 是这个分布的协方差。

当 $\Sigma = \mathbf{I}$ 时，马氏距离退化为欧氏距离。

4.3.2 相似度

1. 余弦相似度

衡量两个向量的相关性 (夹角的余弦)。向量 $\mathbf{x}, \mathbf{y}$ 的余弦相似度为：

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{|\mathbf{x}|\cdot|\mathbf{y}|} \tag{4.5}$$

2. 互信息

定义在两个概率分布 X, Y 上，$x \in X, y \in Y$。它们的互信息为：

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{4.6}$$

3. 皮尔逊相关系数

衡量两个随机变量的相关性。随机变量 X, Y 的 Pearson 相关系数为：

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{4.7}$$

理解：协方差矩阵除以标准差之积。

范围：$[-1, 1]$，绝对值越大表示（正/负）相关性越大。

4. Jaccard 相关系数

对两个集合 X, Y，判断他们的相关性，借用集合的手段：

$$J = \frac{X \cap Y}{X \cup Y} \tag{4.8}$$

理解：两个集合的交集除以并集。

扩展：Jaccard 距离 $=1 - J$。

4.3.3 KL 散度与 JS 距离

KL 散度和 JS 距离是迁移学习中被广泛应用的度量手段。

1. KL 散度

Kullback-Leibler divergence，又叫做相对熵，衡量两个概率分布 $P(x), Q(x)$ 的距离：

$$D_{KL}(P||Q) = \sum_{i=1} P(x) \log \frac{P(x)}{Q(x)} \tag{4.9}$$

这是一个非对称距离：$D_{KL}(P||Q) \neq D_{KL}(Q||P)$.

2. JS 距离

Jensen-Shannon divergence，基于 KL 散度发展而来，是对称度量：

$$JSD(P||Q) = \frac{1}{2} D_{KL}(P||M) + \frac{1}{2} D_{KL}(Q||M) \tag{4.10}$$

其中 $M = \frac{1}{2}(P + Q)$。

4.3.4 最大均值差异 MMD

最大均值差异是迁移学习中使用频率最高的度量。Maximum mean discrepancy，它度量在再生希尔伯特空间中两个分布的距离，是一种核学习方法。两个随机变量的 MMD 平方距离为

$$MMD^2(X,Y) = \left\| \sum_{i=1}^{n_1} \phi(\mathbf{x}_i) - \sum_{j=1}^{n_2} \phi(\mathbf{y}_j) \right\|_{\mathcal{H}}^2 \tag{4.11}$$

其中 $\phi(\cdot)$ 是映射，用于把原变量映射到再生核希尔伯特空间 (Reproducing Kernel Hilbert Space, RKHS) [Borgwardt et al., 2006] 中。什么是 RKHS？形式化定义太复杂，简单来说希尔伯特空间是对于函数的内积完备的，而再生核希尔伯特空间是具有再生性 $\langle K(x,\cdot), K(y,\cdot) \rangle_{\mathcal{H}} = K(x,y)$ 的希尔伯特空间。就是比欧几里得空间更高端的。将平方展开后，RKHS 空间中的内积就可以转换成核函数，所以最终 MMD 可以直接通过核函数进行计算。

理解：就是求两堆数据在 RKHS 中的均值的距离。

Multiple-kernel MMD：多核的 MMD，简称 MK-MMD。现有的 MMD 方法是基于单一核变换的，多核的 MMD 假设最优的核可以由多个核线性组合得到。多核 MMD 的提出和计算方法在文献 [Gretton et al., 2012] 中形式化给出。MK-MMD 在许多后来的方法中被大量使用，最著名的方法是 DAN [Long et al., 2015a]。我们将在后续单独介绍此工作。

4.3.5 Principal Angle

也是将两个分布映射到高维空间 (格拉斯曼流形) 中，在流形中两堆数据就可以看成两个点。Principal angle 是求这两堆数据的对应维度的夹角之和。

对于两个矩阵 $\mathbf{X}, \mathbf{Y}$，计算方法：首先正交化 (用 PCA) 两个矩阵，然后：

$$PA(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{\min(m,n)} \sin\theta_i \tag{4.12}$$

其中 m, n 分别是两个矩阵的维度，θ_i 是两个矩阵第 i 个维度的夹角，$\Theta = \{\theta_1, \theta_2, \cdots, \theta_t\}$ 是两个矩阵 SVD 后的角度：

$$\mathbf{X}^\top \mathbf{Y} = \mathbf{U}(\cos\Theta)\mathbf{V}^\top \tag{4.13}$$

4.3.6 A-distance

$\mathcal{A}$-distance 是一个很简单却很有用的度量。文献 [Ben-David et al., 2007] 介绍了此距离，它可以用来估计不同分布之间的差异性。$\mathcal{A}$-distance 被定义为建立一个线性分类器来区分两个数据领域的 hinge 损失 (也就是进行二类分类的 hinge 损失)。它的计算方式是，我们首先在源域和目标域上训练一个二分类器 h，使得这个分类器可以区分样本是来自于哪一个领域。我们用 $err(h)$ 来表示分类器的损失，则 $\mathcal{A}$-distance 定义为：

$$\mathcal{A}(\mathcal{D}_s, \mathcal{D}_t) = 2(1 - 2err(h)) \tag{4.14}$$

$\mathcal{A}$-distance 通常被用来计算两个领域数据的相似性程度，以便与实验结果进行验证对比。

4.3.7 Hilbert-Schmidt Independence Criterion

希尔伯特-施密特独立性系数，Hilbert-Schmidt Independence Criterion，用来检验两组数据的独立性：

$$HSIC(X, Y) = trace(HXHY) \tag{4.15}$$

其中 X, Y 是两堆数据的 kernel 形式。

4.3.8 Wasserstein Distance

Wasserstein Distance 是一套用来衡量两个概率分部之间距离的度量方法。该距离在一个度量空间 (M, ρ) 上定义，其中 $\rho(x, y)$ 表示集合 M 中两个实例 x 和 y 的距离函数，比如欧几里得距离。两个概率分布 $\mathbb{P}$ 和 $\mathbb{Q}$ 之间的 p-th Wasserstein distance 可以被定义为

$$W_p(\mathbb{P}, \mathbb{Q}) = \Big(\inf_{\mu \in \Gamma(\mathbb{P}, \mathbb{Q})} \int \rho(x, y)^p d\mu(x, y)\Big)^{1/p}, \tag{4.16}$$

其中 $\Gamma(\mathbb{P},\mathbb{Q})$ 是在集合 $M\times M$ 内所有的以 $\mathbb{P}$ 和 $\mathbb{Q}$ 为边缘分布的联合分布。著名的 Kantorovich-Rubinstein 定理表示当 M 是可分离的时候，第一 Wasserstein distance 可以等价地表示成一个积分概率度量 (integral probability metric) 的形式

$$W_1(\mathbb{P},\mathbb{Q}) = \sup_{\|f\|_L\leq 1} \mathbb{E}_{x\sim\mathbb{P}}[f(x)] - \mathbb{E}_{x\sim\mathbb{Q}}[f(x)], \tag{4.17}$$

其中 $\|f\|_L = \sup|f(x)-f(y)|/\rho(x,y)$ 并且 $\|f\|_L\leq 1$ 称为 1− 利普希茨条件。

4.4 迁移学习的理论保证 *

本部分的标题中带有 * 号，有一些难度，为可看可不看的内容。此部分最常见的形式是当自己提出的算法需要理论证明时，可以借鉴。

在第一章里我们介绍了两个重要的概念：迁移学习是什么，以及为什么需要迁移学习。但是，还有一个重要的问题没有得到解答：为什么可以进行迁移？也就是说，迁移学习的可行性还没有探讨。

值得注意的是，就目前的研究成果来说，迁移学习领域的理论工作非常匮乏。我们在这里仅回答一个问题：为什么数据分布不同的两个领域之间，知识可以进行迁移？或者说，到底达到什么样的误差范围，我们才认为知识可以进行迁移？

加拿大滑铁卢大学的 Ben-David 等人从 2007 年开始,连续发表了三篇文章 [Ben-David et al., 2007, Blitzer et al., 2008, Ben-David et al., 2010] 对迁移学习的理论进行探讨。在文中，作者将此称之为“Learning from different domains”。在三篇文章也成为了迁移学习理论方面的经典文章。文章主要回答的问题就是：在怎样的误差范围内，从不同领域进行学习是可行的？

学习误差：给定两个领域 $\mathcal{D}_s,\mathcal{D}_t$，$X$ 是定义在它们之上的数据，一个假设类 $\mathcal{H}$。则两个领域 $\mathcal{D}_s,\mathcal{D}_t$ 之间的 $\mathcal{H}$-divergence 被定义为

$$\hat{d}_{\mathcal{H}}(\mathcal{D}_s,\mathcal{D}_t) = 2\sup_{\eta\in\mathcal{H}}\left|\underset{\mathbf{x}\in\mathcal{D}_s}{P}[\eta(\mathbf{x})=1] - \underset{\mathbf{x}\in\mathcal{D}_t}{P}[\eta(\mathbf{x})=1]\right| \tag{4.18}$$

因此，这个 $\mathcal{H}$-divergence 依赖于假设 $\mathcal{H}$ 来判别数据是来自于 $\mathcal{D}_s$ 还是 $\mathcal{D}_t$。作者证明了，对于一个对称的 $\mathcal{H}$，我们可以通过如下的方式进行计算

$$d_{\mathcal{H}}(\mathcal{D}_s,\mathcal{D}_t) = 2\left(1-\min_{\eta\in\mathcal{H}}\left[\frac{1}{n_1}\sum_{i=1}^{n_1} I[\eta(\mathbf{x}_i)=0] + \frac{1}{n_2}\sum_{i=1}^{n_2} I[\eta(\mathbf{x}_i)=1]\right]\right) \tag{4.19}$$

其中 $I[a]$ 为指示函数：当 a 成立时其值为 1, 否则其值为 0。

在目标领域的泛化界：

假设 $\mathcal{H}$ 为一个具有 d 个 VC 维的假设类，则对于任意的 $\eta\in\mathcal{H}$，下面的不等式有 $1-\delta$ 的概率成立：

$$R_{\mathcal{D}_t}(\eta)\leq R_s(\eta)+\sqrt{\frac{4}{n}(d\log\frac{2en}{d}+\log\frac{4}{\delta})}+\hat{d}_{\mathcal{H}}(\mathcal{D}_s,\mathcal{D}_t)+4\sqrt{\frac{4}{n}(d\log\frac{2n}{d}+\log\frac{4}{\delta})}+\beta \tag{4.20}$$

其中

$$\beta\geq\inf_{\eta^\star\in\mathcal{H}}[R_{\mathcal{D}_s}(\eta^\star)+R_{\mathcal{D}_t}(\eta^\star)] \tag{4.21}$$

并且

$$R_s(\eta) = \frac{1}{n}\sum_{i=1}^{m} I[\eta(\mathbf{x}_i)\neq y_i] \tag{4.22}$$

具体的理论证明细节，请参照上述提到的三篇文章。

在自己的研究中，如果需要进行相关的证明，可以参考一些已经发表的文章的写法，例如 [Long et al., 2014a] 等。

另外，英国的 Gretton 等人也在进行一些学习理论方面的研究，有兴趣的读者可以关注他的个人主页：http://www.gatsby.ucl.ac.uk/~gretton/。

5 迁移学习的基本方法

按照迁移学习领域权威综述文章 A survey on transfer learning [Pan and Yang, 2010]，迁移学习的基本方法可以分为四种。这四种基本的方法分别是：基于样本的迁移，基于模型的迁移，基于特征的迁移，及基于关系的迁移。

本部分简要叙述各种方法的基本原理和代表性相关工作。基于特征和模型的迁移方法是我们的重点。因此，在后续的章节中，将会更加深入地讨论和分析。

5.1 基于样本迁移

基于样本的迁移学习方法 (Instance based Transfer Learning) 根据一定的权重生成规则，对数据样本进行重用，来进行迁移学习。图 14形象地表示了基于样本迁移方法的思想。源域中存在不同种类的动物，如狗、鸟、猫等，目标域只有狗这一种类别。在迁移时，为了最大限度地和目标域相似，我们可以人为地提高源域中属于狗这个类别的样本权重。

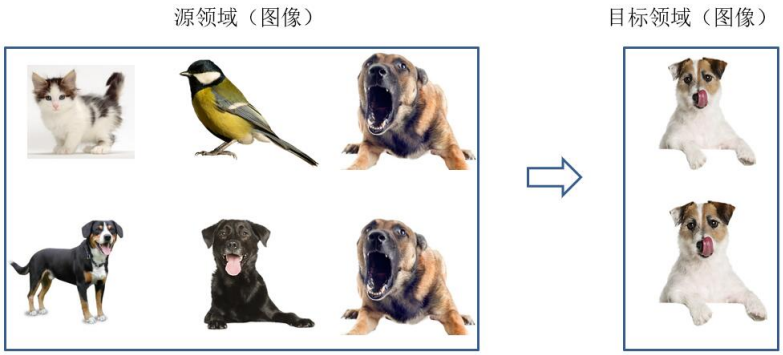


图 14: 基于样本的迁移学习方法示意图

在迁移学习中，对于源域 $\mathcal{D}_s$ 和目标域 $\mathcal{D}_t$，通常假定产生它们的概率分布是不同且未知的 ($P(\mathbf{x}_s) \neq P(\mathbf{x}_t)$)。另外，由于实例的维度和数量通常都非常大，因此，直接对 $P(\mathbf{x}_s)$ 和 $P(\mathbf{x}_t)$ 进行估计是不可行的。因而，大量的研究工作 [Khan and Heisterkamp, 2016, Zadrozny, 2004, Cortes et al., 2008, Dai et al., 2007, Tan et al., 2015, Tan et al., 2017] 着眼于对源域和目标域的分布比值进行估计 ($P(\mathbf{x}_t)/P(\mathbf{x}_s)$)。所估计得到的比值即为样本的权重。这些方法通常都假设 $\frac{P(\mathbf{x}_t)}{P(\mathbf{x}_s)} < \infty$ 并且源域和目标域的条件概率分布相同 ($P(y|\mathbf{x}_s) = P(y|\mathbf{x}_t)$)。特别地，上海交通大学 Dai 等人 [Dai et al., 2007] 提出了 TrAdaboost 方法，将 AdaBoost 的思想应用于迁移学习中，提高有利于目标分类任务的实例权重、降低不利于目标分类任务的实例权重，并基于 PAC 理论推导了模型的泛化误差上界。TrAdaBoost 方法是此方面的经典研究之一。文献 [Huang et al., 2007] 提出核均值匹配方法 (Kernel Mean Matching, KMM) 对于概率分布进行估计，目标是使得加权后的源域和目标域的概率分布尽可能相近。在最新的研究成果中，香港科技大学的 Tan 等人扩展了实例迁移学习方法的应用场景，提出了传递迁移学习方法 (Transitive Transfer Learning, TTL) [Tan et al., 2015] 和远域迁移学习 (Distant Domain Transfer Learning, DDTL) [Tan et al., 2017]，利用联合矩阵分解和深度神经网络，将迁移学习应用于多个不相似的领域之间的知识共享，取得了良好的效果。

虽然实例权重法具有较好的理论支撑、容易推导泛化误差上界，但这类方法通常只在领域间分布差异较小时有效，因此对自然语言处理、计算机视觉等任务效果并不理想。而基于

特征表示的迁移学习方法效果更好，是我们研究的重点。

5.2 基于特征迁移

基于特征的迁移方法 (Feature based Transfer Learning) 是指将通过特征变换的方式互相迁移 [Liu et al., 2011, Zheng et al., 2008, Hu and Yang, 2011]，来减少源域和目标域之间的差距；或者将源域和目标域的数据特征变换到统一特征空间中 [Pan et al., 2011, Long et al., 2014b, Duan et al., 2012]，然后利用传统的机器学习方法进行分类识别。根据特征的同构和异构性，又可以分为同构和异构迁移学习。图 15很形象地表示了两种基于特征的迁移学习方法。

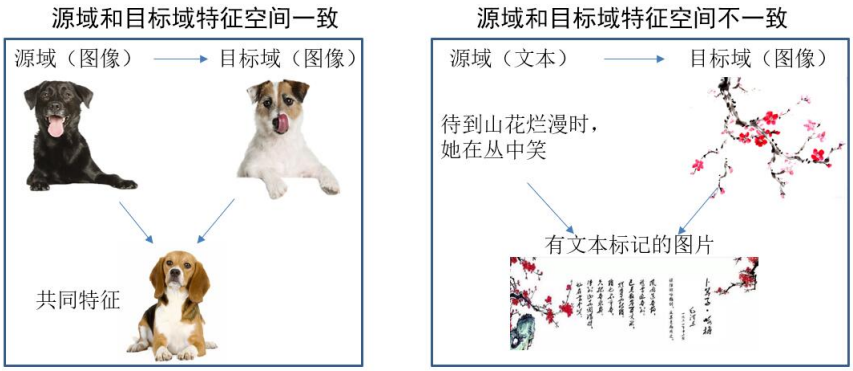


图 15: 基于特征的迁移学习方法示意图

基于特征的迁移学习方法是迁移学习领域中最热门的研究方法，这类方法通常假设源域和目标域间有一些交叉的特征。香港科技大学的 Pan 等人 [Pan et al., 2011] 提出的迁移成分分析方法 (Transfer Component Analysis, TCA) 是其中较为典型的一个方法。该方法的核心内容是以最大均值差异 (Maximum Mean Discrepancy, MMD) [Borgwardt et al., 2006] 作为度量准则，将不同数据领域中的分布差异最小化。加州大学伯克利分校的 Blitzer 等人 [Blitzer et al., 2006] 提出了一种基于结构对应的学习方法 (Structural Corresponding Learning, SCL)，该算法可以通过映射将一个空间中独有的一些特征变换到其他所有空间中的轴特征上，然后在该特征上使用机器学习的算法进行分类预测。清华大学龙明盛等人 [Long et al., 2014b] 提出在最小化分布距离的同时，加入实例选择的迁移联合匹配 (Transfer Joint Matching, TJM) 方法，将实例和特征迁移学习方法进行了有机的结合。澳大利亚卧龙岗大学的 Jing Zhang 等人 [Zhang et al., 2017a] 提出对于源域和目标域各自训练不同的变换矩阵，从而达到迁移学习的目标。

近年来，基于特征的迁移学习方法大多与神经网络进行结合 [Long et al., 2015a, Long et al., 2016, Long et al., 2017, Sener et al., 2016]，在神经网络的训练中进行学习特征和模型的迁移。由于本文的研究重点即是基于特征的迁移学习方法，因此，我们在本小节对这类方法不作过多介绍。在下一小节中，我们将从不同的研究层面，系统地介绍这类工作。

5.3 基于模型迁移

基于模型的迁移方法 (Parameter/Model based Transfer Learning) 是指从源域和目标域中找到他们之间共享的参数信息，以实现迁移的方法。这种迁移方式要求的假设条件是：源域中的数据与目标域中的数据可以共享一些模型的参数。其中的代表性工作主要

有 [Zhao et al., 2010, Zhao et al., 2011, Pan et al., 2008b, Pan et al., 2008a]。图 16形象地表示了基于模型的迁移学习方法的基本思想。

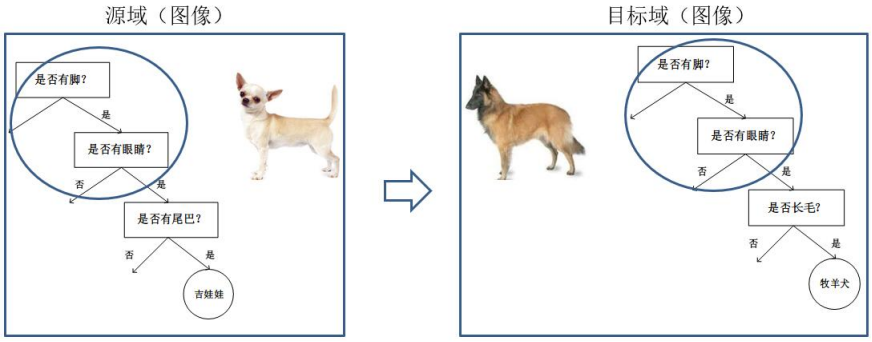


图 16: 基于模型的迁移学习方法示意图

其中，中科院计算所的 Zhao 等人 [Zhao et al., 2011] 提出了 TransEMDT 方法。该方法首先针对已有标记的数据，利用决策树构建鲁棒性的行为识别模型，然后针对无标定数据，利用 K-Means 聚类方法寻找最优化的标定参数。西安邮电大学的 Deng 等人 [Deng et al., 2014] 也用超限学习机做了类似的工作。香港科技大学的 Pan 等人 [Pan et al., 2008a] 利用 HMM，针对 Wifi 室内定位在不同设备、不同时间和不同空间下动态变化的特点，进行不同分布下的室内定位研究。另一部分研究人员对支持向量机 SVM 进行了改进研究 [Nater et al., 2011, Li et al., 2012]。这些方法假定 SVM 中的权重向量 $\mathbf{w}$ 可以分成两个部分：$\mathbf{w} = \mathbf{w_o} + \mathbf{v}$，其中 $\mathbf{w}_0$ 代表源域和目标域的共享部分，$\mathbf{v}$ 代表了对于不同领域的特定处理。在最新的研究成果中，香港科技大学的 Wei 等人 [Wei et al., 2016b] 将社交信息加入迁移学习方法的正则项中，对方法进行了改进。清华大学龙明盛等人 [Long et al., 2015a, Long et al., 2016, Long et al., 2017] 改进了深度网络结构，通过在网络中加入概率分布适配层，进一步提高了深度迁移学习网络对于大数据的泛化能力。

通过对现有工作的调研可以发现，目前绝大多数基于模型的迁移学习方法都与深度神经网络进行结合 [Long et al., 2015a, Long et al., 2016, Long et al., 2017, Tzeng et al., 2015, Long et al., 2016]。这些方法对现有的一些神经网络结构进行修改，在网络中加入领域适配层，然后联合进行训练。因此，这些方法也可以看作是基于模型、特征的方法的结合。

5.4 基于关系迁移

基于关系的迁移学习方法 (Relation Based Transfer Learning) 与上述三种方法具有截然不同的思路。这种方法比较关注源域和目标域的样本之间的关系。图 17形象地表示了不同领域之间相似的关系。

就目前来说，基于关系的迁移学习方法的相关研究工作非常少，仅有几篇连贯式的文章讨论：[Mihalkova et al., 2007, Mihalkova and Mooney, 2008, Davis and Domingos, 2009]。这些文章都借助于马尔科夫逻辑网络 (Markov Logic Net) 来挖掘不同领域之间的关系相似性。

我们将重点讨论基于特征和基于模型的迁移学习方法，这也是目前绝大多数研究工作的热点。

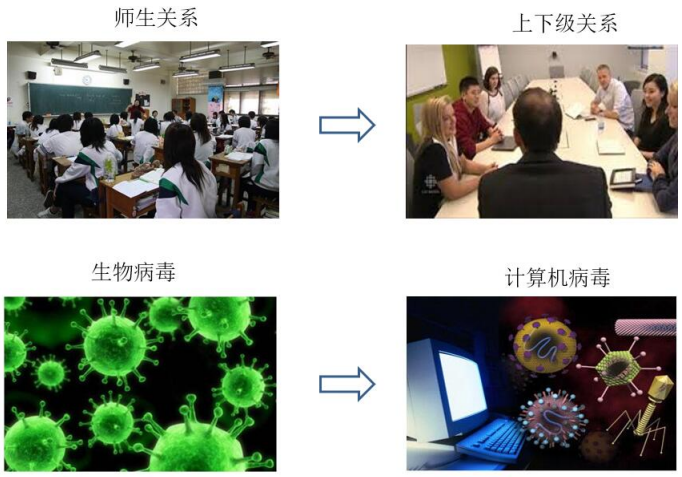


图 17: 基于关系的迁移学习方法示意图

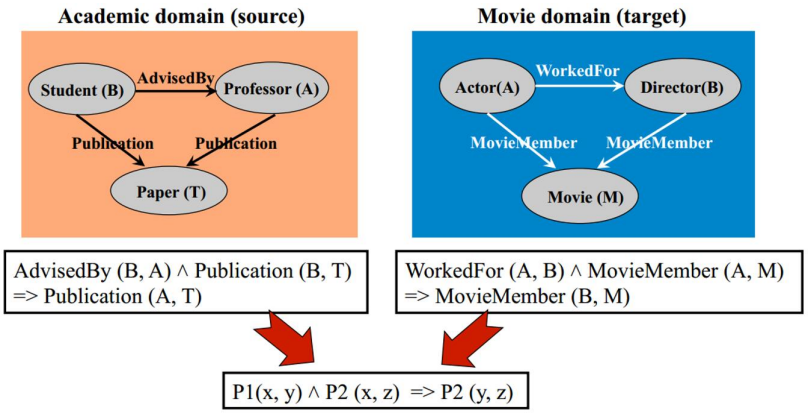
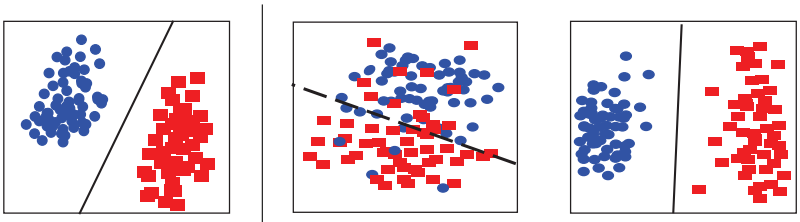


图 18: 基于马尔科夫逻辑网的关系迁移

6 第一类方法：数据分布自适应

数据分布自适应 (Distribution Adaptation) 是一类最常用的迁移学习方法。这种方法的基本思想是，由于源域和目标域的数据概率分布不同，那么最直接的方式就是通过一些变换，将不同的数据分布的距离拉近。

图 19形象地表示了几种数据分布的情况。简单来说，数据的边缘分布不同，就是数据整体不相似。数据的条件分布不同，就是数据整体相似，但是具体到每个类里，都不太相似。



(a) 源域数据　(b) 目标域数据：类型 I　(c) 目标域数据：类型 II

图 19: 不同数据分布的目标域数据

根据数据分布的性质，这类方法又可以分为边缘分布自适应、条件分布自适应、以及联合分布自适应。下面我们分别介绍每类方法的基本原理和代表性研究工作。介绍每类研究工作时，我们首先给出基本思路，然后介绍该类方法的核心，最后结合最近的相关工作介绍该类方法的扩展。

6.1 边缘分布自适应

6.1.1 基本思路

边缘分布自适应方法 (Marginal Distribution Adaptation) 的目标是减小源域和目标域的边缘概率分布的距离，从而完成迁移学习。从形式上来说，边缘分布自适应方法是用 $P(\mathbf{x}_s)$ 和 $P(\mathbf{x}_t)$ 之间的距离来近似两个领域之间的差异。即：

$$DISTANCE(\mathcal{D}_s, \mathcal{D}_t) \approx ||P(\mathbf{x}_s) - P(\mathbf{x}_t)|| \tag{6.1}$$

边缘分布自适应对应于图 19中由图 19(a)迁移到图 19(b)的情形。

6.1.2 核心方法

边缘分布自适应的方法最早由香港科技大学杨强教授团队提出 [Pan et al., 2011]，方法名称为迁移成分分析 (Transfer Component Analysis)。由于 $P(\mathbf{x}_s) \neq P(\mathbf{x}_t)$，因此，直接减小二者之间的距离是不可行的。TCA 假设存在一个特征映射 ϕ，使得映射后数据的分布 $P(\phi(\mathbf{x}_s)) \approx P(\phi(\mathbf{x}_t))$。TCA 假设如果边缘分布接近，那么两个领域的条件分布也会接近，即条件分布 $P(y_s|\phi(\mathbf{x}_s))) \approx P(y_t|\phi(\mathbf{x}_t)))$。这就是 TCA 的全部思想。因此，我们现在的目标是，找到这个合适的 ϕ。

但是世界上有无穷个这样的 ϕ，也许终我们一生也无法找到合适的那一个。庄子说过，吾生也有涯，而知也无涯，以有涯随无涯，殆已！我们肯定不能通过穷举的方法来找 ϕ 的。那么怎么办呢？

回到迁移学习的本质上来：最小化源域和目标域的距离。好了，我们能不能先假设这个 ϕ 是已知的，然后去求距离，看看能推出什么呢？

更进一步，这个距离怎么算？机器学习中有很多种形式的距离，从欧氏距离到马氏距离，从曼哈顿距离到余弦相似度，我们需要什么距离呢？TCA 利用了一个经典的也算是比较“高端”的距离叫做最大均值差异 (MMD，maximum mean discrepancy)。我们令 n_1, n_2 分别表示源域和目标域的样本个数，那么它们之间的 MMD 距离可以计算为：

$$DISTANCE(\mathbf{x}_s, \mathbf{x}_t) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(\mathbf{x}_i) - \frac{1}{n_2} \sum_{j=1}^{n_2} \phi(\mathbf{x}_j) \right\|_{\mathcal{H}} \tag{6.2}$$

MMD 是做了一件什么事呢？简单，就是求映射后源域和目标域的均值之差。

事情到这里似乎也没什么进展：我们想求的 ϕ 仍然没法求。

TCA 是怎么做的呢，这里就要感谢矩阵了！我们发现，上面这个 MMD 距离平方展开后，有二次项乘积的部分！那么，联系在 SVM 中学过的核函数，把一个难求的映射以核函数的形式来求，不就可以了？于是，TCA 引入了一个核矩阵 $\mathbf{K}$：

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{s,s} & \mathbf{K}_{s,t} \\ \mathbf{K}_{t,s} & \mathbf{K}_{t,t} \end{bmatrix} \tag{6.3}$$

以及一个 MMD 矩阵 $\mathbf{L}$，它的每个元素的计算方式为：

$$l_{ij} = \begin{cases} \frac{1}{{n_1}^2} & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s, \\ \frac{1}{{n_2}^2} & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t, \\ -\frac{1}{n_1 n_2} & \text{otherwise} \end{cases} \tag{6.4}$$

这样的好处是，直接把那个难求的距离，变换成了下面的形式：

$$\mathrm{tr}(\mathbf{KL}) - \lambda \mathrm{tr}(\mathbf{K}) \tag{6.5}$$

其中，$\mathrm{tr}(\cdot)$ 操作表示求矩阵的迹，用人话来说就是一个矩阵对角线元素的和。这样是不是感觉离目标又进了一步呢？

其实这个问题到这里就已经是可解的了，也就是说，属于计算机的部分已经做完了。只不过它是一个数学中的半定规划 (SDP，semi-definite programming) 的问题，解决起来非常耗费时间。由于 TCA 的第一作者 Sinno Jialin Pan 以前是中山大学的数学硕士，他想用更简单的方法来解决。他是怎么做的呢？

他想出了用降维的方法去构造结果。用一个更低维度的矩阵 $\mathbf{W}$：

$$\widetilde{\mathbf{K}} = (\mathbf{K}\mathbf{K}^{-1/2}\widetilde{\mathbf{W}})(\widetilde{\mathbf{W}}^{\top}\mathbf{K}^{-1/2}\mathbf{K}) = \mathbf{K}\mathbf{W}\mathbf{W}^{\top}\mathbf{K} \tag{6.6}$$

这里的 $\mathbf{W}$ 矩阵是比 $\mathbf{K}$ 更低维度的矩阵。最后的 $\mathbf{W}$ 就是问题的解答了！

好了，问题到这里，整理一下，TCA 最后的优化目标是：

$$\begin{aligned} \min_{\mathbf{W}} \quad & \mathrm{tr}(\mathbf{W}^{\top}\mathbf{K}\mathbf{L}\mathbf{K}\mathbf{W}) + \mu \mathrm{tr}(\mathbf{W}^{\top}\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{W}^{\top}\mathbf{K}\mathbf{H}\mathbf{K}\mathbf{W} = \mathbf{I}_m \end{aligned} \tag{6.7}$$

这里的 $\mathbf{H}$ 是一个中心矩阵，$\mathbf{H} = \mathbf{I}_{n_1+n_2} - 1/(n_1+n_2)\mathbf{1}\mathbf{1}^{\top}$.

这个式子下面的条件是什么意思呢？那个 min 的目标我们大概理解，就是要最小化源域和目标域的距离，加上 $\mathbf{W}$ 的约束让它不能太复杂。那么下面的条件是什么呢？下面的条件就是要实现第二个目标：维持各自的数据特征。

TCA 要维持的是什么特征呢？文章中说是 variance，但是实际是 scatter matrix，就是数据的散度。就是说，一个矩阵散度怎么计算？对于一个矩阵 $\mathbf{A}$，它的 scatter matrix 就是 $\mathbf{AHA}^\top$。这个 $\mathbf{H}$ 就是上面的中心矩阵啦。

解决上面的优化问题时，作者又求了它的拉格朗日对偶。最后得出结论，$\mathbf{W}$ 的解就是它的前 m 个特征值！简单不？数学美不美？

好了，我们现在总结一下 TCA 方法的步骤。输入是两个特征矩阵，我们首先计算 $\mathbf{L}$ 和 $\mathbf{H}$ 矩阵，然后选择一些常用的核函数进行映射 (比如线性核、高斯核) 计算 $\mathbf{K}$，接着求 $(\mathbf{KLK}+\mu\mathbf{I})^{-1}\mathbf{KHK}$ 的前 m 个特征值。仅此而已。然后，得到的就是源域和目标域的降维后的数据，我们就可以在上面用传统机器学习方法了。

为了形象地展示 TCA 方法的优势，我们借用 [Pan et al., 2011] 中提供的可视化效果，在图中展示了对于源域和目标域数据 (红色和蓝色)，分别由 PCA(主成分分析) 和 TCA 得到的分布结果。从图 20中可以很明显地看出，对于概率分布不同的两部分数据，在经过 TCA 处理后，概率分布更加接近。这说明了 TCA 在拉近数据分布距离上的优势。

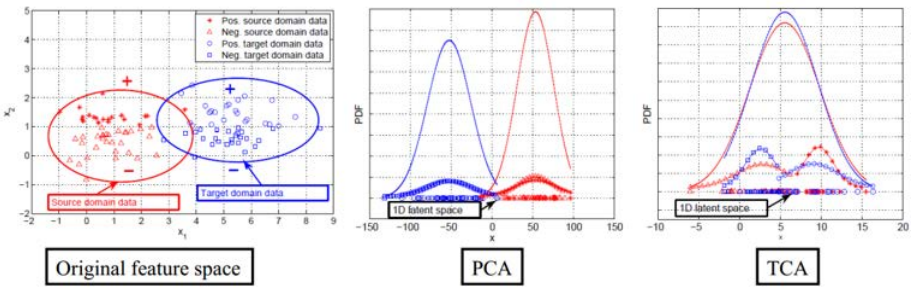


图 20: TCA 和 PCA 的效果对比

6.1.3 扩展

TCA 方法是迁移学习领域一个经典的方法，之后的许多研究工作都以 TCA 为基础。我们列举部分如下：

- ACA (Adapting Component Analysis) [Dorri and Ghodsi, 2012]: 在 TCA 中加入 HSIC
- DTMKL (Domain Transfer Multiple Kernel Learning) [Duan et al., 2012]: 在 TCA 中加入了 MK-MMD，用了新的求解方式
- TJM (Transfer Joint Matching) [Long et al., 2014b]: 在优化目标中同时进行边缘分布自适应和源域样本选择
- DDC (Deep Domain Confusion) [Tzeng et al., 2014]: 将 MMD 度量加入了深度网络特征层的 loss 中 (我们将会在深度迁移学习中介绍此工作)
- DAN (Deep Adaptation Network) [Long et al., 2015a]: 扩展了 DDC 的工作，将 MMD 换成了 MK-MMD，并且进行多层 loss 计算 (我们将会在深度迁移学习中介绍此工作)

- DME (Distribution Matching Embedding): 先计算变换矩阵，再进行特征映射 (与 TCA 顺序相反)
- CMD (Central Moment Matching) [Zellinger et al., 2017]: MMD 着眼于一阶，此工作将 MMD 推广到了多阶

6.2 条件分布自适应

条件分布自适应方法 (Conditional Distribution Adaptation) 的目标是减小源域和目标域的条件概率分布的距离，从而完成迁移学习。从形式上来说，条件分布自适应方法是用 $P(y_s|\mathbf{x}_s)$ 和 $P(y_t|\mathbf{x}_t)$ 之间的距离来近似两个领域之间的差异。即：

$$DISTANCE(\mathcal{D}_s, \mathcal{D}_t) \approx ||P(y_s|\mathbf{x}_s) - P(y_t|\mathbf{x}_t)|| \tag{6.8}$$

条件分布自适应对应于图 19中由图 19(a)迁移到图 19(c)的情形。

目前单独利用条件分布自适应的工作较少，这些工作主要可以在 [Saito et al., 2017] 中找到。最近，中科院计算所的 Wang 等人提出了 STL 方法 (Stratified Transfer Learning) [Wang et al., 2018a]。作者提出了类内迁移 (Intra-class Transfer) 的思想。指出现有的绝大多数方法都只是学习一个全局的特征变换 (Global Domain Shift)，而忽略了类内的相似性。类内迁移可以利用类内特征，实现更好的迁移效果。

STL 方法的基本思路如图 21所示。首先利用大多数投票的思想，对无标定的位置行为生成伪标签；然后在再生核希尔伯特空间中，利用类内相关性进行自适应地空间降维，使得不同情境中的行为数据之间的相关性增大；最后，通过二次标定，实现对未知标定数据的精准标定。

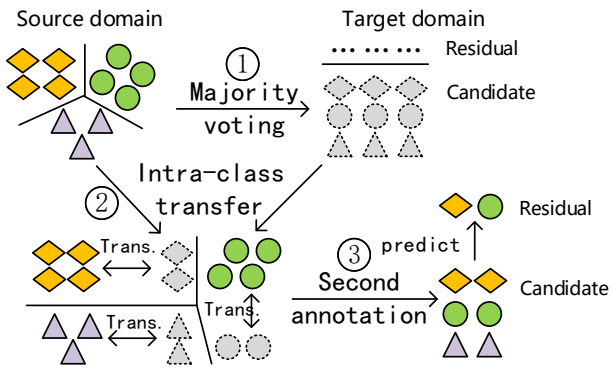


图 21: STL 方法的示意图

为了实现类内迁移，我们需要计算每一类别的 MMD 距离。由于目标域没有标记，作者使用来自大多数投票结果中的伪标记。更加准确地说，用 $c \in \{1, 2, \cdots, C\}$ 来表示类别标记，则类内迁移可以按如下方式计算:

$$D(\mathcal{D}_s, \mathcal{D}_t) = \sum_{c=1}^{C} \left\| \frac{1}{n_1^{(c)}} \sum_{\mathbf{x}_i \in \mathcal{D}_s^{(c)}} \phi(\mathbf{x}_i) - \frac{1}{n_2^{(c)}} \sum_{\mathbf{x}_j \in \mathcal{D}_t^{(c)}} \phi(\mathbf{x}_j) \right\|_{\mathcal{H}}^2 \tag{6.9}$$

其中，$\mathcal{D}_s^{(c)}$ 和 $\mathcal{D}_t^{(c)}$ 分别表示源域和目标域中属于类别 c 的样本。$n_1^{(c)} = |\mathcal{D}_s^{(c)}|$，且 $n_2^{(c)} = |\mathcal{D}_t|$。

接下来的步骤请参照 STL 方法原文进行理解。

STL 方法在大量行为识别数据中进行了跨位置行为识别的实验。实验结果表明，该方法可以很好地实现跨领域的行为识别任务，取得了当前最好的效果。

6.3 联合分布自适应

6.3.1 基本思路

联合分布自适应方法 (Joint Distribution Adaptation) 的目标是减小源域和目标域的联合概率分布的距离，从而完成迁移学习。从形式上来说，联合分布自适应方法是用 $P(\mathbf{x}_s)$ 和 $P(\mathbf{x}_t)$ 之间的距离、以及 $P(y_s|\mathbf{x}_s)$ 和 $P(y_t|\mathbf{x}_t)$ 之间的距离来近似两个领域之间的差异。即：

$$DISTANCE(\mathcal{D}_s, \mathcal{D}_t) \approx ||P(\mathbf{x}_s) - P(\mathbf{x}_t)|| + ||P(y_s|\mathbf{x}_s) - P(y_t|\mathbf{x}_t)|| \tag{6.10}$$

联合分布自适应对应于图 19中由图 19(a)迁移到图 19(b)的情形、以及图 19(a)迁移到图 19(c)的情形。

6.3.2 核心方法

联合分布适配的 JDA 方法 [Long et al., 2013] 首次发表于 2013 年的 ICCV(计算机视觉领域顶会，与 CVPR 类似)，它的作者是当时清华大学的博士生 (现为清华大学助理教授) 龙明盛。

假设是最基本的出发点。那么 JDA 这个方法的假设是什么呢？就是假设两点：1）源域和目标域边缘分布不同，2）源域和目标域条件分布不同。既然有了目标，同时适配两个分布不就可以了吗？于是作者很自然地提出了联合分布适配方法：适配联合概率。

不过这里我感觉有一些争议：边缘分布和条件分布不同，与联合分布不同并不等价。所以这里的“联合”二字实在是会引起歧义。我的理解是，同时适配两个分布，也可以叫联合，而不是概率上的“联合”。尽管作者在文章里第一个公式就写的是适配联合概率，但是这里感觉是有一些问题的。我们抛开它这个有歧义的，把“联合”理解成同时适配两个分布。

那么，JDA 方法的目标就是，寻找一个变换 $\mathbf{A}$，使得经过变换后的 $P(\mathbf{A}^\top\mathbf{x}_s)$ 和 $P(\mathbf{A}^\top\mathbf{x}_t)$ 的距离能够尽可能地接近，同时，$P(y_s|\mathbf{A}^\top\mathbf{x}_s)$ 和 $P(y_t|\mathbf{A}^\top\mathbf{x}_t)$ 的距离也要小。很自然地，这个方法也就分成了两个步骤。

边缘分布适配

首先来适配边缘分布，也就是 $P(\mathbf{A}^\top\mathbf{x}_s)$ 和 $P(\mathbf{A}^\top\mathbf{x}_t)$ 的距离能够尽可能地接近。其实这个操作就是迁移成分分析 (TCA)。我们仍然使用 MMD 距离来最小化源域和目标域的最大均值差异。MMD 距离是

$$\left\| \frac{1}{n}\sum_{i=1}^{n}\mathbf{A}^\top\mathbf{x}_i - \frac{1}{m}\sum_{j=1}^{m}\mathbf{A}^\top\mathbf{x}_j \right\|_{\mathcal{H}}^2 \tag{6.11}$$

这个式子实在不好求解。我们引入核方法，化简这个式子，它就变成了

$$D(\mathcal{D}_s, \mathcal{D}_t) = tr(\mathbf{A}^\top\mathbf{X}\mathbf{M}_0\mathbf{X}^\top\mathbf{A}) \tag{6.12}$$

其中 $\mathbf{A}$ 就是变换矩阵，我们把它加黑加粗，$\mathbf{X}$ 是源域和目标域合并起来的数据。$\mathbf{M}_0$ 是一个 MMD 矩阵：

$$(\mathbf{M}_0)_{ij} = \begin{cases} \frac{1}{n^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s \\ \frac{1}{m^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t \\ -\frac{1}{mn}, & \text{otherwise} \end{cases} \tag{6.13}$$

n, m 分别是源域和目标域样本的个数。

到此为止没有什么创新点，因为这就是一个 TCA。

条件分布适配

这是我们要做的第二个目标，适配源域和目标域的条件概率分布。也就是说，还是要找一个变换 $\mathbf{A}$，使得 $P(y_s|\mathbf{A}^\top \mathbf{x}_s)$ 和 $P(y_t|\mathbf{A}^\top \mathbf{x}_t)$ 的距离也要小。那么简单了，我们再用一遍 MMD 啊。可是问题来了：我们的目标域里，没有 y_t，没法求目标域的条件分布！

这条路看来是走不通了。也就是说，直接建模 $P(y_t|\mathbf{x}_t)$ 不行。那么，能不能有别的办法可以逼近这个条件概率？我们可以换个角度，利用类条件概率 $P(\mathbf{x}_t|y_t)$。根据贝叶斯公式 $P(y_t|\mathbf{x}_t) = p(y_t)p(\mathbf{x}_t|y_t)$，我们如果忽略 $P(\mathbf{x}_t)$，那么岂不是就可以用 $P(\mathbf{x}_t|y_t)$ 来近似 $P(y_t|\mathbf{x}_t)$？

而这样的近似也不是空穴来风。在统计学上，有一个概念叫做充分统计量，它是什么意思呢？大概意思就是说，如果样本里有太多的东西未知，样本足够好，我们就能够从中选择一些统计量，近似地代替我们要估计的分布。好了，我们为近似找到了理论依据。

实际怎么做呢？我们依然没有 y_t。采用的方法是，用 $(\mathbf{x}_s, y_s)$ 来训练一个简单的分类器 (比如 knn、逻辑斯特回归)，到 $\mathbf{x}_t$ 上直接进行预测。总能够得到一些伪标签 $\hat{y}_t$。我们根据伪标签来计算，这个问题就可解了。

类与类之间的 MMD 距离表示为

$$\sum_{c=1}^{C} \left\| \frac{1}{n_c} \sum_{\mathbf{x}_i \in \mathcal{D}_s^{(c)}} \mathbf{A}^\top \mathbf{x}_i - \frac{1}{m_c} \sum_{\mathbf{x}_i \in \mathcal{D}_t^{(c)}} \mathbf{A}^\top \mathbf{x}_i \right\|_{\mathcal{H}}^2 \tag{6.14}$$

其中，n_c, m_c 分别标识源域和目标域中来自第 c 类的样本个数。同样地我们用核方法，得到了下面的式子

$$\sum_{c=1}^{C} tr(\mathbf{A}^\top \mathbf{X} \mathbf{M}_c \mathbf{X}^\top \mathbf{A}) \tag{6.15}$$

其中 $\mathbf{M}_c$ 为

$$(\mathbf{M}_c)_{ij} = \begin{cases} \frac{1}{n_c^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_s^{(c)} \\ \frac{1}{m_c^2}, & \mathbf{x}_i, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ -\frac{1}{m_c n_c}, & \begin{cases} \mathbf{x}_i \in \mathcal{D}_s^{(c)}, \mathbf{x}_j \in \mathcal{D}_t^{(c)} \\ \mathbf{x}_i \in \mathcal{D}_t^{(c)}, \mathbf{x}_j \in \mathcal{D}_s^{(c)} \end{cases} \\ 0, & \text{otherwise} \end{cases} \tag{6.16}$$

现在我们把两个距离结合起来，得到了一个总的优化目标：

$$\min \sum_{c=0}^{C} tr(\mathbf{A}^\top \mathbf{X}\mathbf{M}_c\mathbf{X}^\top \mathbf{A}) + \lambda \|\mathbf{A}\|_F^2 \tag{6.17}$$

看到没，通过 $c = 0 \cdots C$ 就把两个距离统一起来了！其中的 $\lambda\|\mathbf{A}\|_F^2$ 是正则项，使得模型是良好定义 (Well-defined) 的。

我们还缺一个限制条件，不然这个问题无法解。限制条件是什么呢？和 TCA 一样，变换前后数据的方差要维持不变。怎么求数据的方差呢，还和 TCA 一样：$\mathbf{A}^\top\mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A} = \mathbf{I}$，其中的 $\mathbf{H}$ 也是中心矩阵，$\mathbf{I}$ 是单位矩阵。也就是说，我们又添加了一个优化目标是要 $\max \mathbf{A}^\top\mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A}$(这一个步骤等价于 PCA 了)。和原来的优化目标合并，优化目标统一为：

$$\min \frac{\sum_{c=0}^{C} tr(\mathbf{A}^\top \mathbf{X}\mathbf{M}_c\mathbf{X}^\top \mathbf{A}) + \lambda \|\mathbf{A}\|_F^2}{\mathbf{A}^\top\mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A}} \tag{6.18}$$

这个式子实在不好求解。但是，有个东西叫做 Rayleigh quotient [1]，上面两个一样的这种形式。因为 $\mathbf{A}$ 是可以进行拉伸而不改改变最终结果的，而如果下面为 0 的话，整个式子就求不出来值了。所以，我们直接就可以让下面不变，只求上面。所以我们最终的优化问题形式搞成了

$$\min \quad \sum_{c=0}^{C} tr(\mathbf{A}^\top \mathbf{X}\mathbf{M}_c\mathbf{X}^\top \mathbf{A}) + \lambda \|\mathbf{A}\|_F^2 \quad \text{s.t.} \quad \mathbf{A}^\top\mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A} = \mathbf{I} \tag{6.19}$$

怎么解？太简单了，可以用拉格朗日法。最后变成了

$$\left(\mathbf{X}\sum_{c=0}^{C}\mathbf{M}_c\mathbf{X}^\top + \lambda\mathbf{I}\right)\mathbf{A} = \mathbf{X}\mathbf{H}\mathbf{X}^\top\mathbf{A}\Phi \tag{6.20}$$

其中的 Φ 是拉格朗日乘子。别看这个东西复杂，又有要求解的 $\mathbf{A}$，又有一个新加入的 Φ 。但是它在 Matlab 里是可以直接解的 (用 eigs 函数即可)。这样我们就得到了变换 $\mathbf{A}$，问题解决了。

可是伪标签终究是伪标签啊，肯定精度不高，怎么办？有个东西叫做迭代，一次不行，我们再做一次。后一次做的时候，我们用上一轮得到的标签来作伪标签。这样的目的是得到越来越好的伪标签，而参与迁移的数据是不会变的。这样往返多次，结果就自然而然好了。

JDA 方法是十分经典的迁移学习方法。后续的相关工作通过在 JDA 的基础上加入额外的损失项，使得迁移学习的效果得到了很大提升。我们在这里简要介绍一些基于 JDA 的相关工作。

- ARTL (Adaptation Regularization) [Long et al., 2014a]: 将 JDA 嵌入一个结构风险最小化框架中，用表示定理直接学习分类器
- VDA [Tahmoresnezhad and Hashemi, 2016]: 在 JDA 的优化目标中加入了类内距和类间距的计算
- [Hsiao et al., 2016]: 在 JDA 的基础上加入结构不变性控制
- [Hou et al., 2015]：在 JDA 的基础上加入目标域的选择

[1] https://www.wikiwand.com/en/Rayleigh_quotient

- JGSA (Joint Geometrical and Statistical Alignment) [Zhang et al., 2017a]: 在 JDA 的基础上加入类内距、类间距、标签持久化
- JAN (Joint Adaptation Network) [Long et al., 2017]: 提出了联合分布度量 JMMD，在深度网络中进行联合分布的优化

6.4 动态分布自适应

平衡分布自适应 BDA

在最近的研究中，来自中科院计算所的 Wang 等人 [Wang et al., 2017] 注意到了 JDA 的不足：*边缘分布自适应和条件分布自适应并不是同等重要*。回到图 19表示的两种分布的问题上来。显然，当目标域是图 19(b)所示的情况时，边缘分布应该被优先考虑；而当目标域是图 19(c)所示的情况时，条件分布应该被优先考虑。JDA 以及后来的扩展工作均忽视了这一问题。

作者提出了 BDA 方法 (Balanced Distribution Adaptation) 来解决这一问题。该方法能够根据特定的数据领域，自适应地调整分布适配过程中边缘分布和条件分布的重要性。准确而言，BDA 通过采用一种*平衡因子* μ 来动态调整两个分布之间的距离

$$\begin{aligned} DISTANCE(\mathcal{D}_s, \mathcal{D}_t) \approx (1-\mu) DISTANCE(P(\mathbf{x}_s), P(\mathbf{x}_t)) \\ + \mu DISTANCE(P(y_s|\mathbf{x}_s), P(y_t|\mathbf{x}_t)) \end{aligned} \tag{6.21}$$

其中 $\mu \in [0,1]$ 表示平衡因子。当 $\mu \to 0$，这表示源域和目标域数据本身存在较大的差异性，因此，边缘分布适配更重要；当 $\mu \to 1$ 时，这表示源域和目标域数据集有较高的相似性，因此，条件概率分布适配更加重要。综合上面的分析可知，平衡因子可以根据实际数据分布的情况，动态地调节每个分布的重要性，并取得良好的分布适配效果。

其中的平衡因子 μ 可以通过分别计算两个领域数据的整体和局部的 $\mathcal{A}$-distance 近似给出。特别地，当 $\mu = 0$ 时，方法退化为 TCA；当 $\mu = 0.5$ 时，方法退化为 JDA。

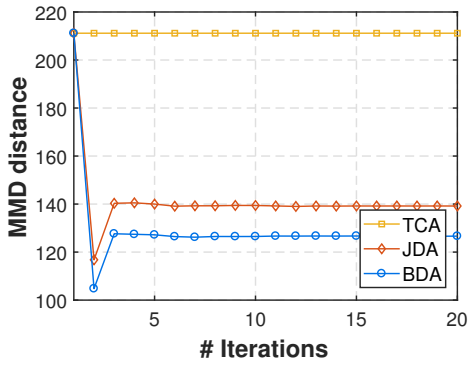
我们采用 BDA 文章中的图来具体地展示出 μ 的作用。图 22的结果清晰地显示出，平衡因子可以取得比 JDA、TCA 更小的 MMD 距离、更高的精度。

动态分布自适应

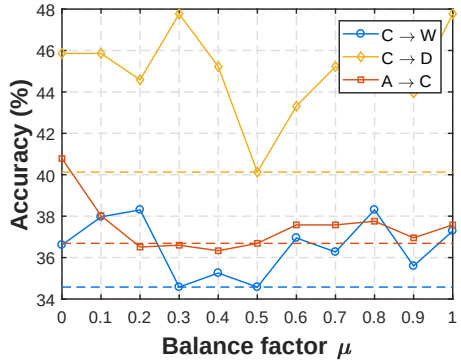
BDA 方法是首次给出边缘分布和条件分布的定量估计。然而，其并未解决平衡因子 μ 的精确计算问题。最近，作者扩展了 BDA 方法，提出了一个更具普适性的动态迁移框架 DDA(Dynamic Distribution Adaptation) [Wang et al., 2019] 来解决 μ 值的精确估计问题。

注意到，可以简单地将 μ 视为一个迁移过程中的参数，通过交叉验证 (cross-validation) 来确定其最优的取值 μ_{opt}。然而，在本章的无监督迁移学习问题定义中，目标域完全没有标记，故此方式不可行。有另外两种非直接的方式可以对 μ 值进行估计：随机猜测和最大最小平均法。随机猜测从神经网络随机调参中得到启发，指的是任意从 $[0,1]$ 区间内选择一个 μ 的值，然后进行动态迁移，其并不算是一种技术严密型的方案。如果重复此过程 t 次，记第 t 次的迁移学习结果为 r_t，则随机猜测法最终的迁移结果为 $r_{rand} = \frac{1}{t}\sum_{i=1}^{t} r_t$。最大最小平均法与随机猜测法相似，可以在 $[0,1]$ 区间内从 0 开始取 μ 的值，每次增加 0.1，得到一个集合 $[0, 0.1, \cdots, 0.9, 1.0]$，然后，与随机猜测法相似，也可以得到其最终迁移结果 $r_{maxmin} = \frac{1}{11}\sum_{i=1}^{11} r_i$。

然而，尽管上述两种估计方案有一定的可行性，它们均需要大量的重复计算，给普适计算设备带来了严峻的挑战。另外，上述结果并不具有可解释性，其正确性也无法得到保证。



(a) 不同方法的 MMD 距离比较



(b) BDA 方法中平衡因子 μ 的作用

图 22: BDA 方法的效果

作者提出的动态迁移方法是首次对 μ 值进行精确的定量估计方法。该方法利用领域的整体和局部性质来定量计算 μ(计算出的值用 $\hat{\mu}$ 来表示)。采用 $\mathcal{A}-distance$ [Ben-David et al., 2007] 作为基本的度量方式。$\mathcal{A}-distance$ 被定义为建立一个二分类器进行两个不同领域的分类得出的误差。从形式化来看，定义 $\epsilon(h)$ 作为线性分类器 h 区分两个领域 Ω_s 和 Ω_t 的误差。则，$\mathcal{A}-distance$ 可以被定义为：

$$d_A(\Omega_s, \Omega_t) = 2(1 - 2\epsilon(h)). \tag{6.22}$$

直接根据上式计算边缘分布的 $\mathcal{A}-distance$，将其用 d_M 来表示。对于条件分布之间的 $\mathcal{A}-distance$，用 d_c 来表示对应于类别 c 的条件分布距离。它可以由式 $d_c = d_A(\Omega_s^{(c)}, \Omega_t^{(c)})$ 进行计算，其中 $\Omega_s^{(c)}$ 和 $\Omega_t^{(c)}$ 分别表示来自源域和目标域的第 c 个类的样本。最终，μ 可以由下式进行计算：

$$\hat{\mu} = 1 - \frac{d_M}{d_M + \sum_{c=1}^{C} d_c}. \tag{6.23}$$

由于特征的动态和渐近变化性，此估计需要在每一轮迭代中给出。值得注意的是，这是**首次**给出边缘分布和条件分布的定量估计，对于迁移学习研究具有很大的意义。

具体而言，作者将机器学习问题规约成一个统计机器学习问题，可以用统计机器学习中的结构风险最小化的原则 (Structural Risk Minimization, SRM) [Belkin et al., 2006, Vapnik and Vapnik, 1998] 进行表示学习。在 SRM 中，分类器 f 可以被表示为：

$$f = \underset{f \in \mathcal{H}_K, (\mathbf{x}, y) \sim \Omega_l}{\arg\min} J(f(\mathbf{x}), y) + R(f), \tag{6.24}$$

其中第一项表示 f 在有标记数据上的损失，第二项为正则项，$\mathcal{H}_K$ 表示核函数 $K(\cdot,\cdot)$ 构造的希尔伯特空间 (Hilbert space)。符号 Ω_l 表示有标记的数据领域。在本章的问题中，$\Omega_l = \Omega_s$，即只有源域数据有标记。特别地，由于在迁移学习问题中，源域和目标域数据有着不同的数据分布，为了表示此分布距离，可以进一步将正则项表示成如下的形式：

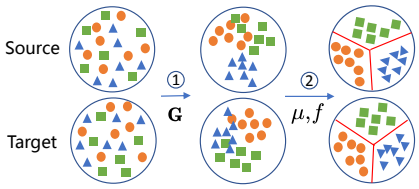
$$R(f) = \lambda \overline{D_f}(\Omega_s, \Omega_t) + R_f(\Omega_s, \Omega_t), \tag{6.25}$$

其中 $\overline{D_f}(\cdot,\cdot)$ 表示 Ω_s 和 Ω_t 的分布距离，λ 为平衡系数，$R_f(\cdot,\cdot)$ 则为其他形式的正则项。根据公式 (6.24) 中的结构风险最小化公式，如果用 $g(\cdot)$ 来表示特征学习过程，则 f 可以被表示为：

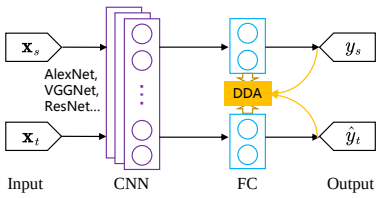
$$f = \underset{f \in \sum_{i=1}^{n} \mathcal{H}_K}{\arg\min} J(f(g(\mathbf{x}_i)), y_i) + \eta ||f||_K^2 + \lambda \overline{D_f}(\Omega_s, \Omega_t) + \rho R_f(\Omega_s, \Omega_t), \tag{6.26}$$

其中 $||f||_K^2$ 是 f 的平方标准形式。$\overline{D_f}(\cdot,\cdot)$ 这一项表示本章提出的动态迁移学习。引入拉普拉斯约束作为 f 的额外正则项 [Belkin et al., 2006]。η, λ, 和 ρ 是对应的正则项系数。

上式则为通用的一个迁移学习框架，可以适用于任何问题。为了对此框架进行学习，作者分别提出了基于流形学习的动态迁移方法 MEDA (Manifold Embedded Distribution Alignment) [Wang et al., 2018b] 和基于深度学习的动态迁移方法 DDAN (Deep Dynamic Adaptation Network) [Wang et al., 2019] 来进行学习。这两种方法分别如图 23(a)和 23(b)所示。



(a) 流形空间动态迁移 MEDA



(b) 深度网络动态迁移 DDAN

图 23: 动态分布自适应

最近，作者在 [Yu et al., 2019] 中将 DDA 的概念进一步扩展到了对抗网络中，证明了对抗网络中同样存在边缘分布和条件分布不匹配的问题。作者提出一个动态对抗适配网络 DAAN (Dynamic Adversarial Adaptation Networks) 来解决对抗网络中的动态分布适配问题，取得了当前的最好效果。图 24展示了 DAAN 的架构。

6.5 小结

综合上述三种概率分布自适应方法，我们可以得出如下的结论：

1. 精度比较：DDA > JDA > TCA > 条件分布自适应。

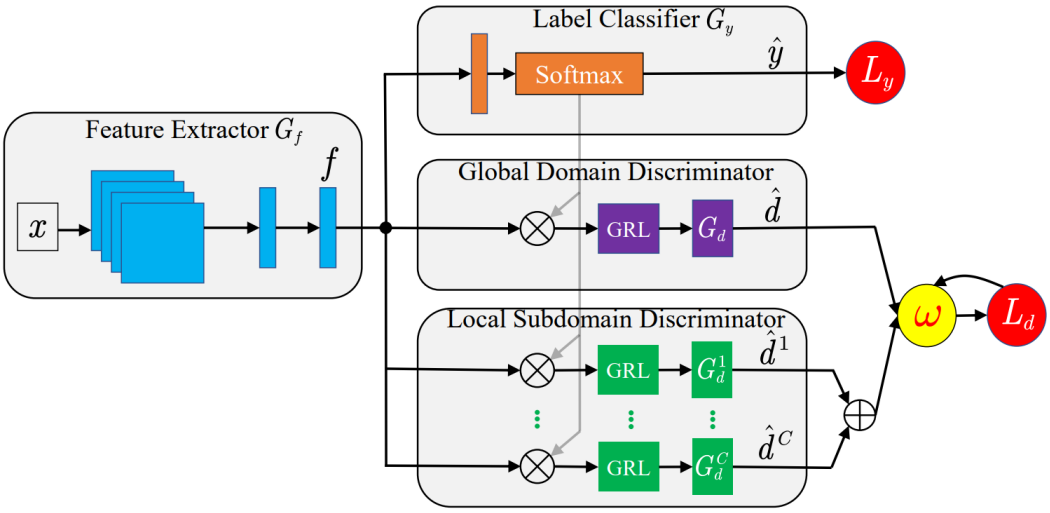


图 24: 动态对抗适配网络 DAAN 结构示意图

2. 将不同的概率分布自适应方法用于神经网络，是一个发展趋势。将概率分布适配加入深度网络中，往往会取得比非深度方法更好的结果。

7 第二类方法：特征选择

特征选择法的基本假设是：源域和目标域中均含有一部分公共的特征，在这部分公共的特征上，源领域和目标领域的数据分布是一致的。因此，此类方法的目标就是，通过机器学习方法，选择出这部分共享的特征，即可依据这些特征构建模型。

图 25形象地表示了特征选择法的主要思路。

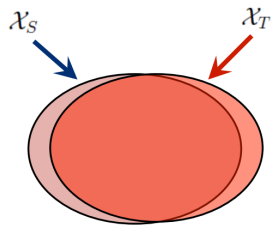


图 25: 特征选择法示意图

7.1 核心方法

这这个领域比较经典的一个方法是发表在 2006 年的 ECML-PKDD 会议上，作者提出了一个叫做 SCL 的方法 (Structural Correspondence Learning) [Blitzer et al., 2006]。这个方法的目标就是我们说的，找到两个领域公共的那些特征。作者将这些公共的特征叫做 Pivot feature。找出来这些 Pivot feature，就完成了迁移学习的任务。

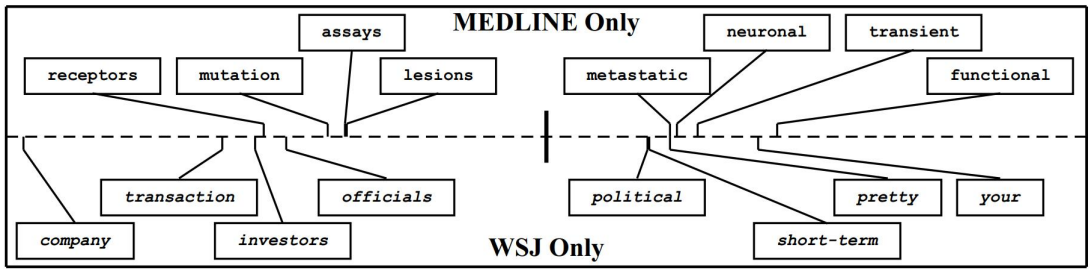


图 26: 特征选择法中的 Pivot feature 示意图

图 26形象地展示了 Pivot feature 的含义。Pivot feature 指的是在文本分类中，在不同领域中出现频次较高的那些词。

7.2 扩展

SCL 方法是特征选择方面的经典研究工作。基于 SCL，也出现了一些扩展工作。

- Joint feature selection and subspace learning [Gu et al., 2011]：特征选择 + 子空间学习
- TJM (Transfer Joint Matching) [Long et al., 2014b]: 在优化目标中同时进行边缘分布自适应和源域样本选择
- FSSL (Feature Selection and Structure Preservation) [Li et al., 2016]: 特征选择 + 信息不变性

7.3 小结

- 特征选择法从源域和目标域中选择提取共享的特征，建立统一模型
- 通常与分布自适应方法进行结合
- 通常采用稀疏表示 $||\mathbf{A}||_{2,1}$ 实现特征选择

8 第三类方法：子空间学习

子空间学习法通常假设源域和目标域数据在变换后的子空间中会有着相似的分布。我们按照特征变换的形式，将子空间学习法分为两种：基于统计特征变换的统计特征对齐方法，以及基于流形变换的流形学习方法。下面我们分别介绍这两种方法的基本思路和代表性研究成果。

8.1 统计特征对齐

统计特征对齐方法主要将数据的统计特征进行变换对齐。对齐后的数据，可以利用传统机器学习方法构建分类器进行学习。

SA 方法 (Subspace Alignment，子空间对齐) [Fernando et al., 2013] 是其中的代表性成果。SA 方法直接寻求一个线性变换 $\mathbf{M}$，将不同的数据实现变换对齐。SA 方法的优化目标如下：

$$F(\mathbf{M}) = ||\mathbf{X}_s\mathbf{M} - \mathbf{X}_t||_F^2 \tag{8.1}$$

则变换 $\mathbf{M}$ 的值为：

$$\mathbf{M}^\star = \arg\min_{\mathbf{M}}(F(\mathbf{M})) \tag{8.2}$$

可以直接获得上述优化问题的闭式解：

$$F(\mathbf{M}) = ||\mathbf{X}_s^\top\mathbf{X}_s\mathbf{M} - \mathbf{X}_s^\top\mathbf{X}_t||_F^2 = ||\mathbf{M} - \mathbf{X}_s^\top\mathbf{X}_t||_F^2 \tag{8.3}$$

SA 方法实现简单，计算过程高效，是子空间学习的代表性方法。

基于 SA 方法，Sun 等人在 2015 年提出了 SDA 方法 (Subspace Distribution Alignment) [Sun and Saenko, 2015]。该方法在 SA 的基础上，加入了概率分布自适应。图 27示意了该方法的简单流程。SDA 方法提出，除了子空间变换矩阵 $\mathbf{T}$ 之外，还应当增加一个概率分布自适应变换 $\mathbf{A}$。SDA 方法的优化目标如下：

$$\mathbf{M} = \mathbf{X}_s\mathbf{T}\mathbf{A}\mathbf{X}_t^\top \tag{8.4}$$

有别于 SA 和 SDA 方法只进行源域和目标域的一阶特征对齐，Sun 等人提出了 CORAL 方法 (CORrelation ALignment)，对两个领域进行二阶特征对齐。假设 $\mathbf{C}_s$ 和 $\mathbf{C}_t$ 分别是源领域和目标领域的协方差矩阵，则 CORAL 方法学习一个二阶特征变换 $\mathbf{A}$，使得源域和目标域的特征距离最小：

$$\min_{\mathbf{A}} ||\mathbf{A}^\top\mathbf{C}_s\mathbf{A} - \mathbf{C}_t||_F^2 \tag{8.5}$$

CORAL 方法的求解同样非常简单且高效。CORAL 方法被应用到神经网络中，提出了 DeepCORAL 方法 [Sun and Saenko, 2016]。作者将 CORAL 度量作为一个神经网络的损失进行计算。图展示了 DeepCORAL 方法的网络结构。

CORAL 损失被定义为源域和目标域的二阶统计特征距离：

$$\ell_{CORAL} = \frac{1}{4d^2}||\mathbf{C}_s - \mathbf{C}_t||_F^2 \tag{8.6}$$

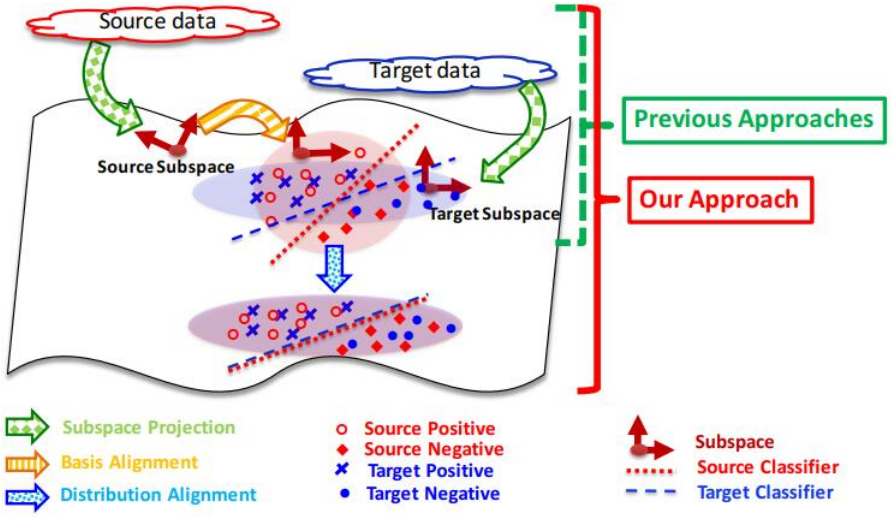


图 27: SDA 方法的示意图

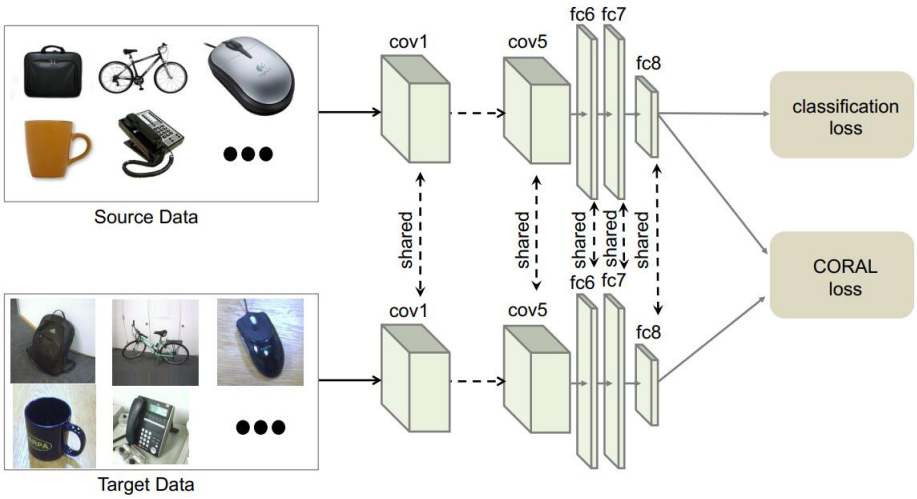


图 28: Deep-CORAL 方法示意图

8.2 流形学习

流形学习自从 2000 年在 Science 上被提出来以后，就成为了机器学习和数据挖掘领域的热门问题。它的基本假设是，现有的数据是从一个高维空间中采样出来的，所以，它具有高维空间中的低维流形结构。流形就是一种几何对象 (就是我们能想像能观测到的)。通俗点说，我们无法从原始的数据表达形式明显看出数据所具有的结构特征，那我把它想像成是处在一个高维空间，在这个高维空间里它是有个形状的。一个很好的例子就是星座。满天星星怎么描述？我们想像它们在一个更高维的宇宙空间里是有形状的，这就有了各自星座，比如织女座、猎户座。流形学习的经典方法有 Isomap、locally linear embedding、laplacian eigenmap 等。

流形空间中的距离度量：两点之间什么最短？在二维上是直线（线段），可在三维呢？地球上的两个点的最短距离可不是直线，它是把地球展开成二维平面后画的那条直线。那条线在三维的地球上就是一条曲线。这条曲线就表示了两个点之间的最短距离，我们叫它测地线。更通俗一点，两点之间，测地线最短。在流形学习中，我们遇到测量距离的时候，更多的时候用的就是这个测地线。在我们要介绍的 GFK 方法中，也是利用了这个测地线距离。比如在下面的图中，从 A 到 C 最短的距离在就是展开后的线段，但是在三维球体上看，它却是一条曲线。

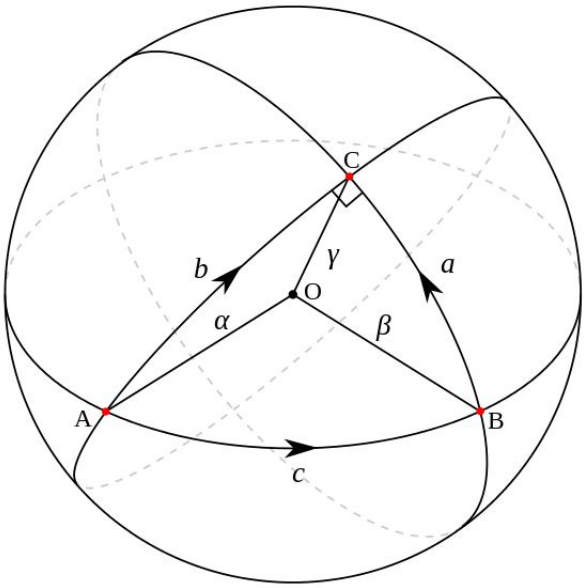


图 29: 三维空间中两点之间的距离示意图

由于在流形空间中的特征通常都有着很好的几何性质，可以避免特征扭曲，因此我们首先将原始空间下的特征变换到流形空间中。在众多已知的流形中，Grassmann 流形 $\mathbb{G}(d)$ 可以通过将原始的 d 维子空间 (特征向量) 看作它基础的元素，从而可以帮助学习分类器。在 Grassmann 流形中，特征变换和分布适配通常都有着有效的数值形式，因此在迁移学习问题中可以被很高效地表示和求解 [Hamm and Lee, 2008]。因此，利用 Grassmann 流形空间中来进行迁移学习是可行的。现存有很多方法可以将原始特征变换到流形空间中 [Gopalan et al., 2011, Baktashmotlagh et al., 2014]。

在众多的基于流形变换的迁移学习方法中，GFK(Geodesic Flow Kernel) 方法 [Gong et al., 2012] 是最为代表性的一个。GFK 是在 2011 年发表在 ICCV 上的 SGF 方法 [Gopalan et al., 2011] 发展起来的。我们首先介绍 SGF 方法。

SGF 方法从增量学习中得到启发：人类从一个点想到达另一个点，需要从这个点一步一步走到那一个点。那么，如果我们把源域和目标域都分别看成是高维空间中的两个点，由源域变换到目标域的过程不就完成了迁移学习吗？也就是说，路是一步一步走出来的。

于是 SGF 就做了这个事情。它是怎么做的呢？把源域和目标域分别看成高维空间 (即 Grassmann 流形) 中的两个点，在这两个点的测地线距离上取 d 个中间点，然后依次连接起来。这样，源域和目标域就构成了一条测地线的路径。我们只需要找到合适的每一步的变换，就能从源域变换到目标域了。图 30是 SGF 方法的示意图。

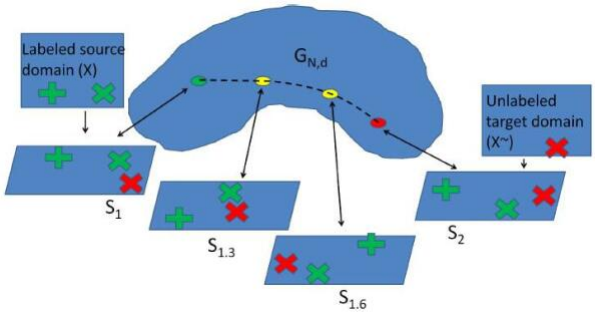


图 30: SGF 流形迁移学习方法示意图

SGF 方法的主要贡献在于：提出了这种变换的计算及实现了相应的算法。但是它有很明显的缺点：到底需要找几个中间点？SGF 也没能给出答案，就是说这个参数 d 是没法估计的，没有一个好的方法。这个问题在 GFK 中被回答了。

GFK 方法首先解决 SGF 的问题：如何确定中间点的个数 d。它通过提出一种核学习的方法，利用路径上的无穷个点的积分，把这个问题解决了。这是第一个贡献。然后，它又解决了第二个问题：当有多个源域的时候，我们如何决定使用哪个源域跟目标域进行迁移？GFK 通过提出 Rank of Domain 度量，度量出跟目标域最近的源域，来解决这个问题。图 31是 GFK 方法的示意图。

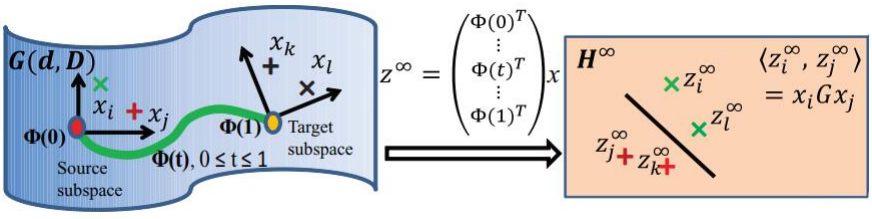


图 31: GFK 流形迁移学习方法示意图

用 $\mathcal{S}_s$ 和 $\mathcal{S}_t$ 分别表示源域和目标域经过主成分分析 (PCA) 之后的子空间，则 G 可以视为所有的 d 维子空间的集合。每一个 d 维的原始子空间都可以被看作 $\mathbb{G}$ 上的一个点。因此，在两点之间的测地线 $\{\Phi(t) : 0 \leq t \leq 1\}$ 可以在两个子空间之间构成一条路径。如果我们令 $\mathcal{S}_s = \Phi(0)$，$\mathcal{S}_t = \Phi(1)$，则寻找一条从 $\Phi(0)$ 到 $\Phi(1)$ 的测地线就等同于将原始的特征变换到一个无穷维度的空间中，最终减小域之间的漂移现象。这种方法可以被看作是一种从 $\Phi(0)$ 到 $\Phi(1)$ 的增量式“行走”方法。

特别地，流形空间中的特征可以被表示为 $\mathbf{z} = \Phi(t)^\top \mathbf{x}$。变换后的特征 $\mathbf{z}_i$ 和 $\mathbf{z}_j$ 的内积定义了一个半正定 (positive semidefinite) 的测地线流式核

$$\langle \mathbf{z}_i, \mathbf{z}_j \rangle = \int_0^1 (\Phi(t)^T \mathbf{x}_i)^T (\Phi(t)^T \mathbf{x}_j)\, dt = \mathbf{x}_i^T \mathbf{G} \mathbf{x}_j \tag{8.7}$$

GFK 方法详细的计算过程可以参考原始的文章，我们在这里不再赘述。

8.3 扩展与小结

子空间学习方法和概率分布自适应方法可以有机地进行组合，克服各自的缺点。下面是一些相关工作。

- DIP (Domain-Invariant Projection) [Baktashmotlagh et al., 2013]: 边缘分布自适应 + 流形变换
- [Baktashmotlagh et al., 2014]: 统计流形法，在黎曼流形上进行距离度量。

最近的一些工作 [Sun and Saenko, 2016] 显示，子空间学习法和神经网络的结合会更好。

9 深度迁移学习

随着深度学习方法的大行其道，越来越多的研究人员使用深度神经网络进行迁移学习。对比传统的非深度迁移学习方法，深度迁移学习直接提升了在不同任务上的学习效果。并且，由于深度学习直接对原始数据进行学习，所以其对比非深度方法还有两个优势：自动化地提取更具表现力的特征，以及满足了实际应用中的端到端 *(End-to-End)* 需求。

近年来，以生成对抗网络 (Generative Adversarial Nets, GAN) [Goodfellow et al., 2014] 为代表的对抗学习也吸引了很多研究者的目光。基于 GAN 的各种变体网络不断涌现。对抗学习网络对比传统的深度神经网络，极大地提升了学习效果。因此，基于对抗网络的迁移学习，也是一个热门的研究点。

图 32展示了近几年的一些代表性方法在相同数据集上的表现。从图中的结果我们可以看出，深度迁移学习方法 (BA、DDC、DAN) 对比传统迁移学习方法 (TCA、GFK 等)，在精度上具有无可匹敌的优势。

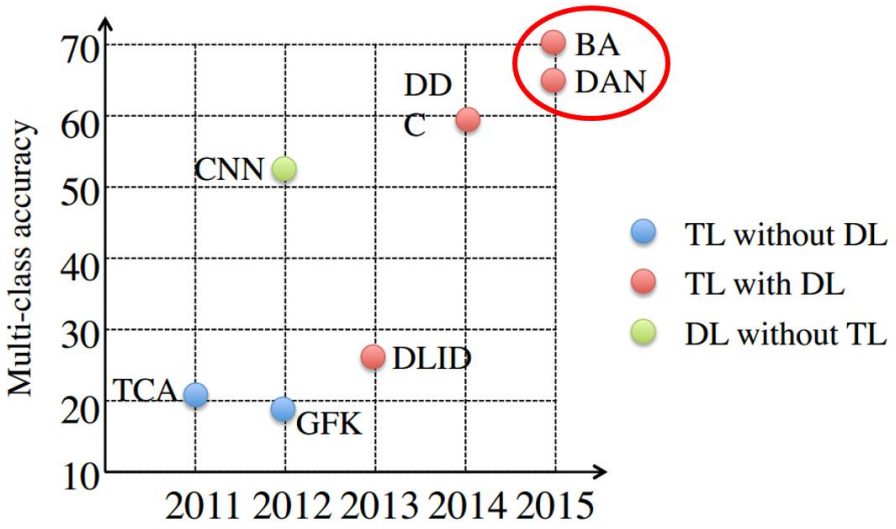


图 32: 深度与非深度迁移学习方法的结果对比

本部分重点介绍深度迁移学习的基本思路。首先我们回答一个最基本的问题：为什么深度网络是可迁移的？然后，我们介绍最简单的深度网络迁移形式：finetune。接着分别介绍使用深度网络和深度对抗网络进行迁移学习的基本思路和核心方法。值得注意的是，由于深度迁移学习方面的研究工作层出不穷，我们不可能覆盖到所有最新的方法。但是基本上，这些方法的原理都大同小异。因此，我们的介绍是具有普适性的。

9.1 深度网络的可迁移性

随着 AlexNet [Krizhevsky et al., 2012] 在 2012 年的 ImageNet 大赛上获得冠军，深度学习开始在机器学习的研究和应用领域大放异彩。尽管取得了很好的结果，但是神经网络本身就像一个黑箱子，看得见，摸不着，解释性不好。由于神经网络具有良好的层次结构，很自然地就有人开始关注，能否通过这些层次结构来很好地解释网络？于是，有了我们熟知的例子：假设一个网络要识别一只猫，那么一开始它只能检测到一些边边角角的东西，和猫根本没有关系；然后可能会检测到一些线条和圆形；慢慢地，可以检测到有猫的区域；接着是猫腿、猫脸等等。图 33是一个简单的示例。

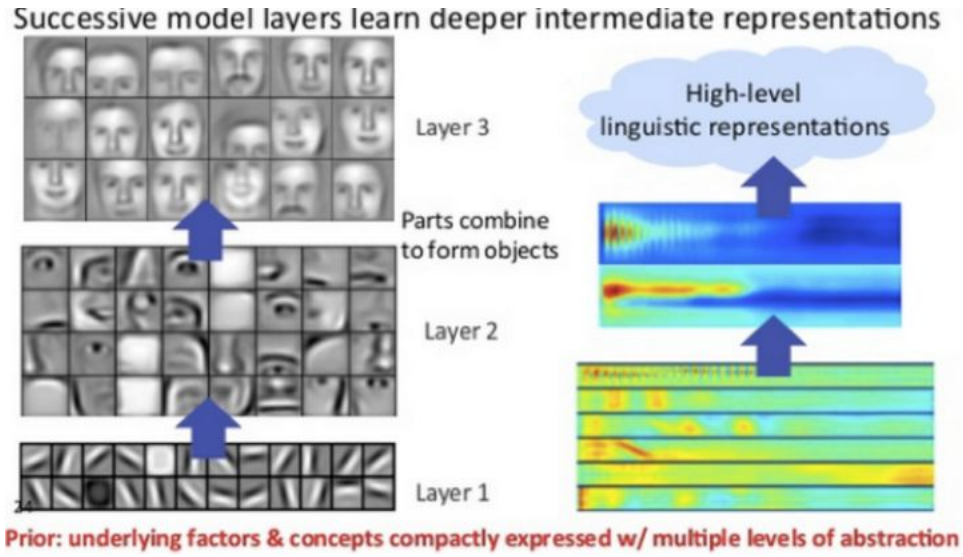


图 33: 深度神经网络进行特征提取到分类的简单示例

这表达了一个什么事实呢？概括来说就是：前面几层都学习到的是通用的特征（general feature）；随着网络层次的加深，后面的网络更偏重于学习任务特定的特征（specific feature）。这非常好理解，我们也都很好接受。那么问题来了：如何得知哪些层能够学习到 general feature，哪些层能够学习到 specific feature。更进一步：*如果应用于迁移学习，如何决定该迁移哪些层、固定哪些层？*

这个问题对于理解神经网络以及深度迁移学习都有着非常重要的意义。

来自康奈尔大学的 Jason Yosinski 等人 [Yosinski et al., 2014] 率先进行了深度神经网络可迁移性的研究，将成果发表在 2014 年机器学习领域顶级会议 NIPS 上并做了口头汇报。该论文是一篇实验性质的文章（通篇没有一个公式）。其目的就是要探究上面我们提到的几个关键性问题。因此，文章的全部贡献都来自于实验及其结果。（别说为啥做实验也能发文章：都是高考，我只上了个普通一本，我高中同学就上了清华）

在 ImageNet 的 1000 类上，作者把 1000 类分成两份（A 和 B），每份 500 个类别。然后，分别对 A 和 B 基于 Caffe 训练了一个 AlexNet 网络。一个 AlexNet 网络一共有 8 层，除去第 8 层是类别相关的网络无法迁移以外，作者在 1 到 7 这 7 层上逐层进行 finetune 实验，探索网络的可迁移性。

为了更好地说明 finetune 的结果，作者提出了有趣的概念：AnB 和 BnB。

迁移 A 网络的前 n 层到 B（AnB）vs 固定 B 网络的前 n 层（BnB）

简单说一下什么叫 AnB：（所有实验都是针对数据 B 来说的）将 A 网络的前 n 层拿来并将它 frozen，剩下的 $8-n$ 层随机初始化，然后对 B 进行分类。

相应地，有 BnB：把训练好的 B 网络的前 n 层拿来并将它 frozen，剩下的 $8-n$ 层随机初始化，然后对 B 进行分类。

实验结果

实验结果如下图 (图 34) 所示：

这个图说明了什么呢？我们先看蓝色的 BnB 和 BnB+(就是 BnB 加上 finetune)。对 BnB 而言，原训练好的 B 模型的前 3 层直接拿来就可以用而不会对模型精度有什么损失。到了第 4 和第 5 层，精度略有下降，不过还是可以接受。然而到了第 6 第第 7 层，精度居然奇迹般地回升了！这是为什么？原因如下：对于一开始精度下降的第 4 第 5 层来说，确

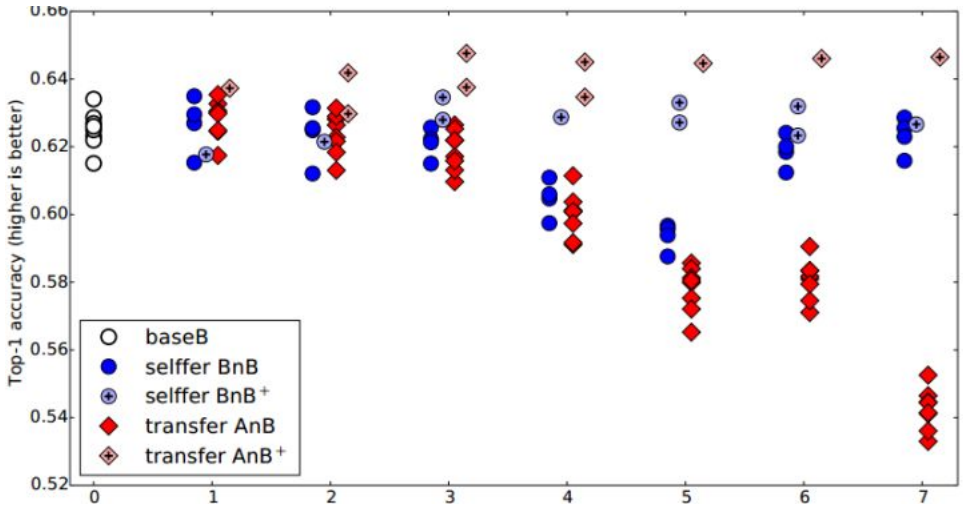


图 34: 深度网络迁移实验结果 1

实是到了这一步，feature 变得越来越 specific，所以下降了。那对于第 6 第 7 层为什么精度又不变了？那是因为，整个网络就 8 层，我们固定了第 6 第 7 层，这个网络还能学什么呢？所以很自然地，精度和原来的 B 网络几乎一致！

对 BnB+ 来说，结果基本上都保持不变。说明 finetune 对模型结果有着很好的促进作用！

我们重点关注 AnB 和 AnB+。对 AnB 来说，直接将 A 网络的前 3 层迁移到 B，貌似不会有什么影响，再一次说明，网络的前 3 层学到的几乎都是 general feature！往后，到了第 4 第 5 层的时候，精度开始下降，我们直接说：一定是 feature 不 general 了！然而，到了第 6 第 7 层，精度出现了小小的提升后又下降，这又是为什么？作者在这里提出两点：co-adaptation 和 feature representation。就是说，第 4 第 5 层精度下降的时候，主要是由于 A 和 B 两个数据集的差异比较大，所以会下降；到了第 6 第 7 层，由于网络几乎不迭代了，学习能力太差，此时 feature 学不到，所以精度下降得更厉害。

再看 AnB+。加入了 finetune 以后，AnB+ 的表现对于所有的 n 几乎都非常好，甚至比 baseB（最初的 B）还要好一些！这说明：finetune 对于深度迁移有着非常好的促进作用！

把上面的结果合并就得到了下面一张图 (图 35)：

至此，AnB 和 BnB 基本完成。作者又想，是不是我分 A 和 B 数据的时候，里面存在一些比较相似的类使结果好了？比如说 A 里有猫，B 里有狮子，所以结果会好？为了排除这些影响，作者又分了一下数据集，这次使得 A 和 B 里几乎没有相似的类别。在这个条件下再做 AnB，与原来精度比较（0% 为基准）得到了下图 (图 36)：

这个图说明了什么呢？简单：随着可迁移层数的增加，模型性能下降。但是，前 3 层仍然还是可以迁移的！同时，与随机初始化所有权重比较，迁移学习的精度是很高的!

结论

虽然该论文并没有提出一个创新方法，但是通过实验得到了以下几个结论，对以后的深度学习和深度迁移学习都有着非常高的指导意义：

- 神经网络的前 3 层基本都是 general feature，进行迁移的效果会比较好；
- 深度迁移网络中加入 fine-tune，效果会提升比较大，可能会比原网络效果还好；
- Fine-tune 可以比较好地克服数据之间的差异性；

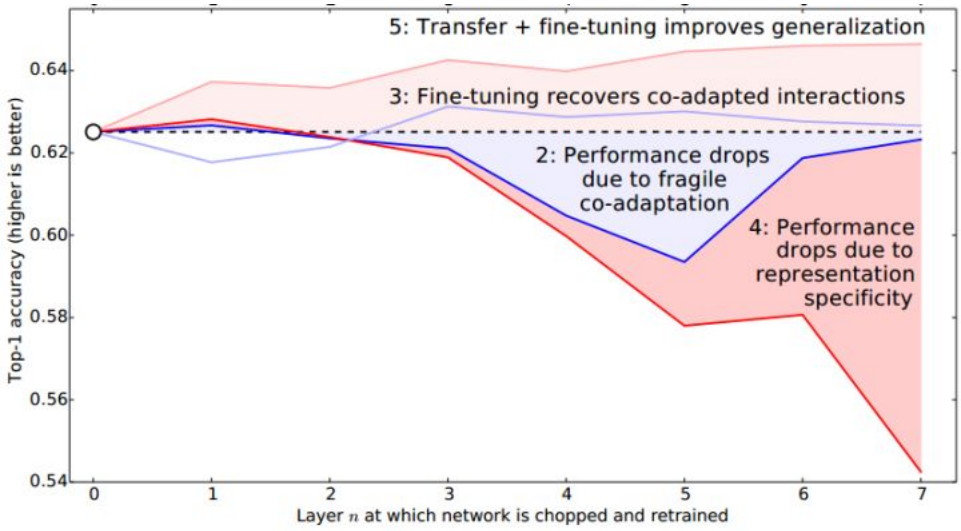


图 35: 深度网络迁移实验结果 2

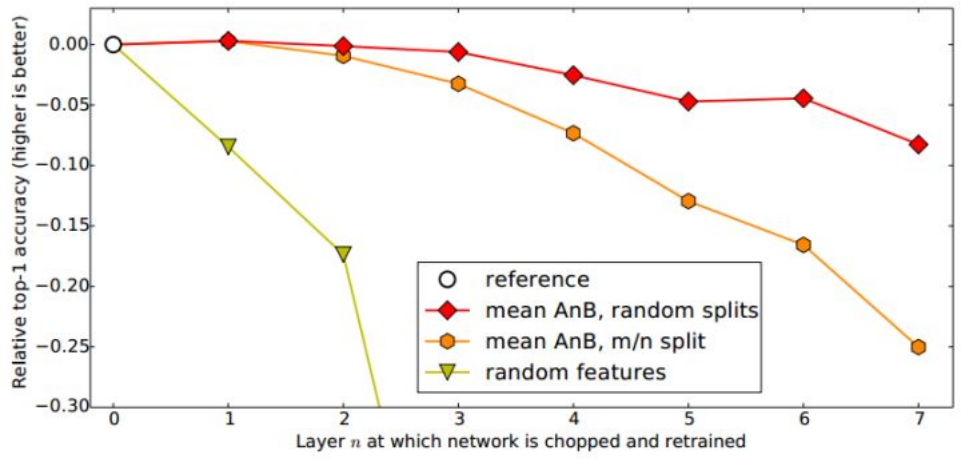


图 36: 深度网络迁移实验结果 3

- 深度迁移网络要比随机初始化权重效果好；
- 网络层数的迁移可以加速网络的学习和优化。

9.2 最简单的深度迁移：finetune

深度网络的 finetune 也许是最简单的深度网络迁移方法。**Finetune**，也叫微调、fine-tuning，是深度学习中的一个重要概念。简而言之，finetune 就是利用别人已经训练好的网络，针对自己的任务再进行调整。从这个意思上看，我们不难理解 finetune 是迁移学习的一部分。

1. 为什么需要已经训练好的网络？

在实际的应用中，我们通常不会针对一个新任务，就去从头开始训练一个神经网络。这样的操作显然是非常耗时的。尤其是，我们的训练数据不可能像 ImageNet 那么大，可以训练出泛化能力足够强的深度神经网络。即使有如此之多的训练数据，我们从头开始训练，其代价也是不可承受的。

那么怎么办呢？迁移学习告诉我们，利用之前已经训练好的模型，将它很好地迁移到自己的任务上即可。

2. 为什么需要 finetune？

因为别人训练好的模型，可能并不是完全适用于我们自己的任务。可能别人的训练数据和我们的数据之间不服从同一个分布；可能别人的网络能做比我们的任务更多的事情；可能别人的网络比较复杂，我们的任务比较简单。

举一个例子来说，假如我们想训练一个猫狗图像二分类的神经网络，那么很有参考价值的就是在 CIFAR-100 上训练好的神经网络。但是 CIFAR-100 有 100 个类别，我们只需要 2 个类别。此时，就需要针对我们自己的任务，固定原始网络的相关层，修改网络的输出层，以使结果更符合我们的需要。

图 37展示了一个简单的 finetune 过程。从图中我们可以看到，我们采用的预训练好的网络非常复杂，如果直接拿来从头开始训练，则时间成本会非常高昂。我们可以将此网络进行改造，固定前面若干层的参数，只针对我们的任务，微调后面若干层。这样，网络训练速度会极大地加快，而且对提高我们任务的表现也具有很大的促进作用。

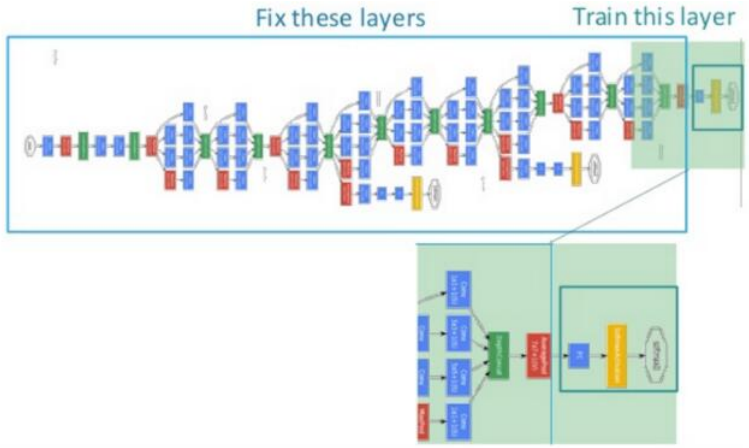


图 37: 一个简单的 finetune 示意图

3. Finetune 的优势

Finetune 的优势是显然的，包括：

- 不需要针对新任务从头开始训练网络，节省了时间成本；
- 预训练好的模型通常都是在大数据集上进行的，无形中扩充了我们的训练数据，使得模型更鲁棒、泛化能力更好；
- Finetune 实现简单，使得我们只关注自己的任务即可。

4. Finetune 的扩展

在实际应用中，通常几乎没有人会针对自己的新任务从头开始训练一个神经网络。Finetune 是一个理想的选择。

Finetune 并不只是针对深度神经网络有促进作用，对传统的非深度学习也有很好的效果。例如，finetune 对传统的人工提取特征方法就进行了很好的替代。我们可以使用深度网络对原始数据进行训练，依赖网络提取出更丰富更有表现力的特征。然后，将这些特征作为传统机器学习方法的输入。这样的好处是显然的：既避免了繁复的手工特征提取，又能自动地提取出更有表现力的特征。

比如，图像领域的研究，一直是以 SIFT、SURF 等传统特征为依据的，直到 2014 年，伯克利的研究人员提出了 DeCAF 特征提取方法 [Donahue et al., 2014]，直接使用深度卷积神经网络进行特征提取。实验结果表明，该特征提取方法对比传统的图像特征，在精度上有着无可匹敌的优势。另外，也有研究人员用卷积神经网络提取的特征作为 SVM 分类器的输入 [Razavian et al., 2014]，显著提升了图像分类的精度。

9.3 深度网络自适应

9.3.1 基本思路

深度网络的 finetune 可以帮助我们节省训练时间，提高学习精度。但是 finetune 有它的先天不足：它无法处理训练数据和测试数据分布不同的情况。而这一现象在实际应用中比比皆是。因为 finetune 的基本假设也是训练数据和测试数据服从相同的数据分布。这在迁移学习中也是不成立的。因此，我们需要更进一步，针对深度网络开发出更好的方法使之更好地完成迁移学习任务。

以我们之前介绍过的数据分布自适应方法为参考，许多深度学习方法 [Tzeng et al., 2014, Long et al., 2015a] 都开发出了自适应层 (Adaptation Layer) 来完成源域和目标域数据的自适应。自适应能够使得源域和目标域的数据分布更加接近，从而使得网络的效果更好。

从上述的分析我们可以得出，深度网络的自适应主要完成两部分的工作：

一是哪些层可以自适应，这决定了网络的学习程度；

二是采用什么样的自适应方法 (度量准则)，这决定了网络的泛化能力。

深度网络中最重要的是网络损失的定义。绝大多数深度迁移学习方法都采用了以下的损失定义方式：

$$\ell = \ell_c(\mathcal{D}_s, \mathbf{y}_s) + \lambda \ell_A(\mathcal{D}_s, \mathcal{D}_t) \tag{9.1}$$

其中，ℓ 表示网络的最终损失，$\ell_c(\mathcal{D}_s, \mathbf{y}_s)$ 表示网络在有标注的数据 (大部分是源域) 上的常规分类损失 (这与普通的深度网络完全一致)，$\ell_A(\mathcal{D}_s, \mathcal{D}_t)$ 表示网络的自适应损失。最后一部分是传统的深度网络所不具有的、迁移学习所独有的。此部分的表达与我们先前讨论过的源域和目标域的分布差异，在道理上是相同的。式中的 λ 是权衡两部分的权重参数。

上述的分析指导我们设计深度迁移网络的基本准则：决定自适应层，然后在这些层加入自适应度量，最后对网络进行 finetune。

9.3.2 核心方法

前期的研究者在 2014 年环太平洋人工智能大会 (PRICAI) 上提出了一个叫做 DaNN(Domain Adaptive Neural Network) 的神经网络 [Ghifary et al., 2014]。DaNN 的结构异常简单，它仅由两层神经元组成：特征层和分类器层。作者的创新工作在于，在特征层后加入了一项 MMD 适配层，用来计算源域和目标域的距离，并将其加入网络的损失中进行训练。

但是，由于网络太浅，表征能力有限，故无法很有效地解决 domain adaptation 问题。因此，后续的研究者大多数都基于其思想进行扩充，如将浅层网络改为更深层的 AlexNet、ResNet、VGG 等；如将 MMD 换为多核的 MMD 等。

1. 第一个方法：DDC

加州大学伯克利分校的 Tzeng 等人 [Tzeng et al., 2014] 首先提出了一个 DDC 方法 (Deep Domain Confusion) 解决深度网络的自适应问题。DDC 遵循了我们上述讨论过的基本思路，采用了在 ImageNet 数据集上训练好的 AlexNet 网络 [Krizhevsky et al., 2012] 进行自适应学习。

图 38是 DDC 方法的示意。DDC 固定了 AlexNet 的前 7 层，在第 8 层 (分类器前一层) 上加入了自适应的度量。自适应度量方法采用了被广泛使用的 MMD 准则。DDC 方法的损失函数表示为：

$$\ell = \ell_c(\mathcal{D}_s, \mathbf{y}_s) + \lambda MMD^2(\mathcal{D}_s, \mathcal{D}_t) \tag{9.2}$$

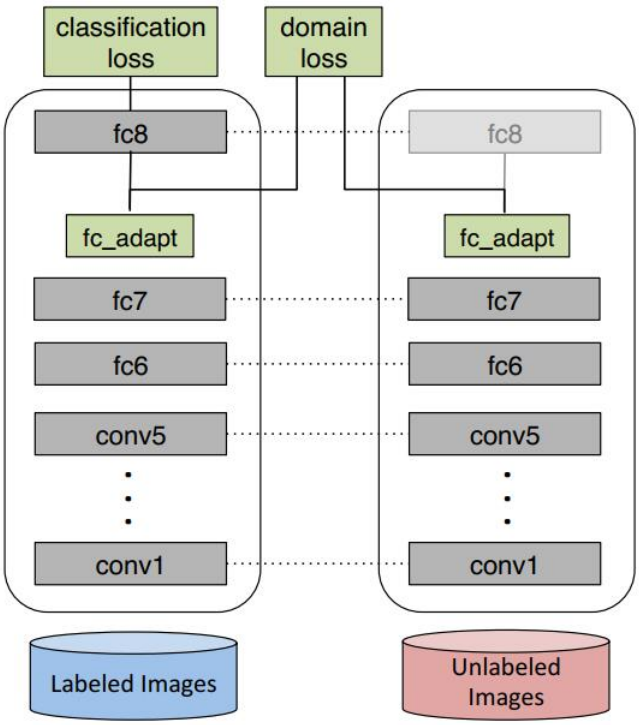


图 38: DDC 方法示意图

为什么选择了倒数第二层？ DDC 方法的作者在文章中提到，他们经过了多次实验，在不同的层进行了尝试，最终得出结论，在分类器前一层加入自适应可以达到最好的效果。

这也是与我们的认知相符合的。通常来说，分类器前一层即特征，在特征上加入自适应，也正是迁移学习要完成的工作。

2. DAN

来自清华大学的龙明盛等人在 2015 年发表在机器学习顶级会议 ICML 上的 DAN 方法 (Deep Adaptation Networks) [Long et al., 2015a] 对 DDC 方法进行了几个方面的扩展。首先，有别于 DDC 方法只加入一个自适应层，DAN 方法同时加入了三个自适应层 (分类器前三层)。其次，DAN 方法采用了表征能力更好的多核 MMD 度量 (MK-MMD) [Gretton et al., 2012] 代替了 DDC 方法中的单一核 MMD。然后，DAN 方法将多核 MMD 的参数学习融入到深度网络的训练中，不增加网络的额外训练时间。DAN 方法在多个任务上都取得了比 DDC 更好的分类效果。

为什么是适配 3 层？原来的 DDC 方法只是适配了一层，现在 DAN 也基于 AlexNet 网络，适配最后三层（第 6 第 7 第 8 层）。为什么是这三层？因为在 Jason 的文章 [Yosinski et al., 2014] 中已经说了，网络的迁移能力在这三层开始就会特别地 task-specific，所以要着重适配这三层。至于别的网络（比如 GoogLeNet、VGG）等是不是这三层就需要通过自己的实验来推测。DAN 只关注使用 AlexNet。

MK-MMD 的多核表示形式为

$$\mathcal{K} \triangleq \left\{ k = \sum_{u=1}^{m} \beta_u k_u : \beta_u \geq 0, \forall u \right\} \tag{9.3}$$

DAN 的优化目标也由两部分组成：损失函数和自适应损失。损失函数这个好理解，基本上所有的机器学习方法都会定义一个损失函数，它来度量预测值和真实值的差异。分布距离就是我们上面提到的 MK-MMD 距离。于是，DAN 的优化目标就是

$$\min_{\Theta} \frac{1}{n_a} \sum_{i=1}^{n_a} J(\theta(\mathbf{x}_i^a), y_i^a) + \lambda \sum_{l=l_1}^{l_2} d_k^2(\mathcal{D}_s^l, \mathcal{D}_t^l) \tag{9.4}$$

这个式子中，Θ 表示网络的所有权重和 bias 参数，是用来学习的目标。其中 l_1, l_2 分别是 6 和 8, 表示网络适配是从第 6 层到第 8 层，前面的不进行适配。$\mathbf{x}_a, n_a$ 表示源域和目标域中所有有标注的数据的集合。$J(\cdot)$ 就定义了一个损失函数，在深度网络中一般都是 cross-entropy。DAN 的网络结构如下图所示。

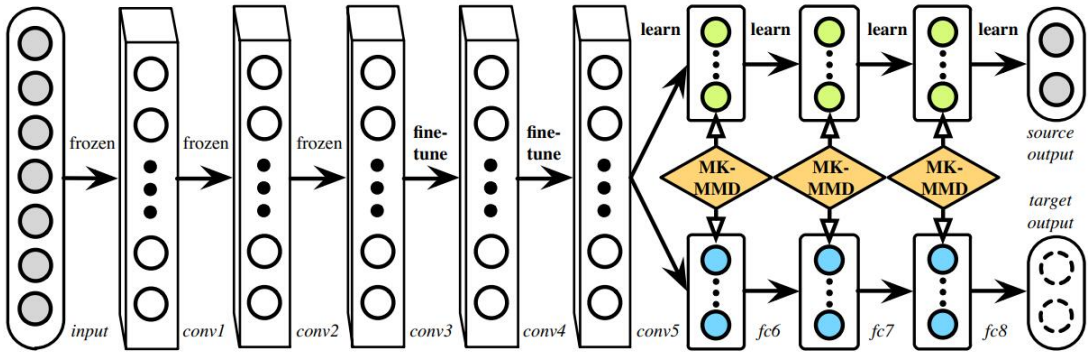


图 39: DAN 方法示意图

学习策略

DAN 的学习一共分为两大类参数：学习网络参数 Θ 和 MMD 的 β。

对 Θ 的学习依赖于 MK-MMD 距离的计算。通过 kernel trick(类比以前的 MMD 变换) 我们总是可以把 MK-MMD 展开成一堆内积的形式。然而，数据之间两两计算内积是非常复杂的，时间复杂度为 $O(n^2)$，这个在深度学习中的开销非常之大。怎么办？作者在这里采用了 Gretton 在文章 [Gretton et al., 2012] 提出的对 MK-MMD 的无偏估计：$d_k^2(p,q) = \frac{2}{n_s}\sum_{i=1}^{n_s/2} g_k(\mathbf{z}_i)$，其中的 $\mathbf{z}_i$ 是一个四元组：$\mathbf{z}_i \triangleq (\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^s, \mathbf{x}_{2i-1}^t, \mathbf{x}_{2i}^t)$。将 kernel 作用到 $\mathbf{z}_i$ 上以后，变成 $g_k(\mathbf{z}_i) \triangleq k(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^s) + k(\mathbf{x}_{2i-1}^t, \mathbf{x}_{2i}^t) - k(\mathbf{x}_{2i-1}^s, \mathbf{x}_{2i}^t) - k(\mathbf{x}_{2i}^s, \mathbf{x}_{2i-1}^t)$。

上面这些变换看着很复杂。简单来说，它就是只计算了连续的一对数据的距离，再乘以 2. 这样就可以把时间复杂度降低到 $O(n)$！至于具体的理论，可以去参考 Gretton 的论文，这里就不说了。

在具体进行 SGD 的时候，我们需要对所有的参数求导：对 Θ 求导。在实际用 multiple-kernel 的时候，作者用的是多个高斯核。

学习 β 主要是为了确定多个 kernel 的权重。学习的时候，目标是：确保每个 kernel 生成的 MMD 距离的方差最小。也就是

$$\max_{k\in\mathcal{K}} d_k^2(\mathcal{D}_s^l, \mathcal{D}_t^l)\sigma_k^{-2} \tag{9.5}$$

这里的 $\sigma_k^{-2} = E[g_k^2(\mathbf{z})] - [E(g_k(\mathbf{z}))]^2$ 是估计方差。实际求解的时候问题可以被规约成一个二次规划问题求解，具体可以参照文章。

3. 同时迁移领域和任务

DDC 的作者 Tzeng 在 2015 年扩展了 DDC 方法，提出了领域和任务同时迁移的方法 [Tzeng et al., 2015]。作者提出网络要进行两部分的迁移：

一是 domain transfer，就是适配分布，特别地是指适配 marginal distribution，但是没有考虑类别信息。如何做 domain transfer：在传统深度网路的 loss 上，再加另一个 confusion loss，作为 classifier 能否将两个 domain 进行分开的 loss。两个 loss 一起计算，就是 domain transfer。

二是 task transfer，就是利用 class 之间的相似度，其实特指的是 conditional distribution。类别之间有相似度，要利用上。类别之间的相似度：比如一个杯子与瓶子更相似，而与键盘不相似。文章的原话：it does not necessarily align the classes in the target with those in the source. Thus, we also explicity transfer the similarity structure amongst categories.

现有的深度迁移学习方法通常都只是考虑 domain transfer，而没有考虑到类别之间的信息。如何把 domain 和 task transfer 结合起来，是一个问题。

文章针对的情况是：target 的部分 class 有少量 label，剩下的 class 无 label。

作者提出的方法名字叫做 joint CNN architecture for domain and task transfer。最大的创新点是：现有的方法都是 domain classifier 加上一个 domain confusion，就是适配。作者提出这些是不够的，因为忽略了类别之间的联系，所以提出了还要再加一个 soft label loss。意思就是在 source 和 target 进行适配的时候，也要根据 source 的类别分布情况来进行调整 target 的。其实本意和 JDA 差不多。

相应地，文章的方法就是把这两个 loss 结合到一个新的 CNN 网络上，这个 CNN 是用

AlexNet 的结构修改成的。总的 loss 可以表示成几部分的和：

$$L(x_S, y_S, x_T, y_T, \theta_D; \theta_{repr}, \theta_C) = L_C(x_S, y_S, x_T, y_T; \theta_{repr}, \theta_C) + \lambda L_{conf}(x_S, x_T, \theta_D; \theta_{repr}) + vL_{soft}(x_T, y_T; \theta_{repr}, \theta_C) \tag{9.6}$$

Loss 由三部分组成：第一部分是普通训练的 loss，对应于经验风险最小化，所以作用到了所有有 label 的数据 x_S, y_S, x_T, y_T 上；第二部分是现有方法也都做过的，就是 domain adaptation，所以作用于 x_S, x_T 上；第三部分是作者的贡献：新加的 soft label 的 loss，只作用于 target 上。

网络结构如下图 40所示。网络由 AlexNet 修改而来，前面的几层都一样，区别只是在第 fc7 层后面加入了一个 domain classifier，也就是进行 domain adaptation 的一层；在 fc8 后计算网络的 loss 和 soft label 的 loss。

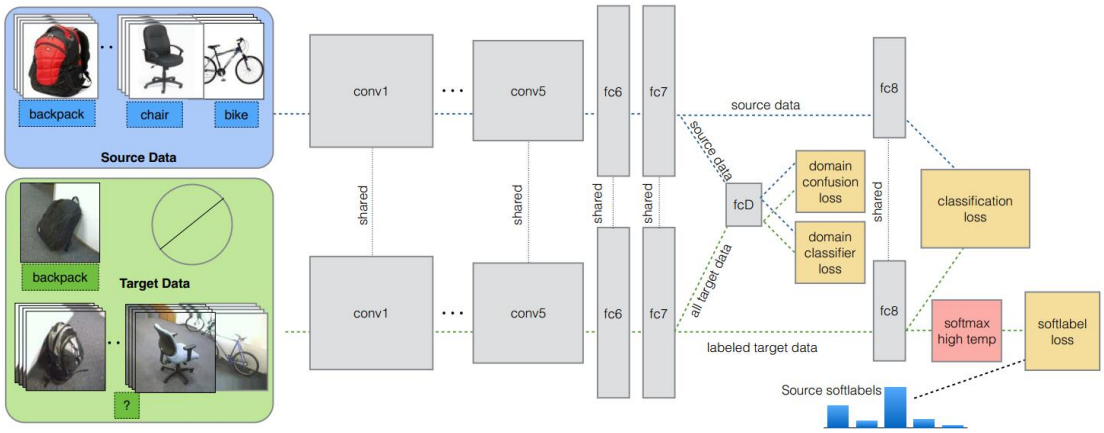


图 40: Joint CNN architecture for domain and task transfer 方法示意图

Domain confusion 就不用多说，和现有的一些比如 DAN 和 JAN 一样，直接对 source 和 target 的 margina distribution 进行估计即可。

什么是 soft label loss？这和作者的 motivation 有关。不仅仅要适配两个 domain 的 marginal distribution，也要把类别信息考虑进去。而 target 中有大量数据没有 label，怎么办呢？可以利用 source 中的 label 信息。具体做法是：在网络对 source 进行训练的时候，把 source 的每一个样本处于每一个类的概率都记下来，然后，对于所有样本，属于每一个类的概率就可以通过求和再平均得到。如下图所示。这样的目的是：根据 source 中的类别分布关系，来对 target 做相应的约束。比如，source 中和 bike 最相似的 class，肯定是 motorbike，而不是 truck。这样有一定的道理。

但是在我看来，这就是深度网络版的 conditional distribution adaptation。因为也对应于得到了每个类的 proportion。只不过作者在这里取了个“soft label loss”这个好听的名字，再冠以“task transfer”这样高大上的名字。图 41是 soft label 的示意。

4. 深度联合分布自适应

DAN 的作者、清华大学的龙明盛在 2017 年机器学习顶级会议 ICML 上提出了 JAN 方法 (Joint Adaptation Network) [Long et al., 2017]，在深度网络中同时进行联合分布的自

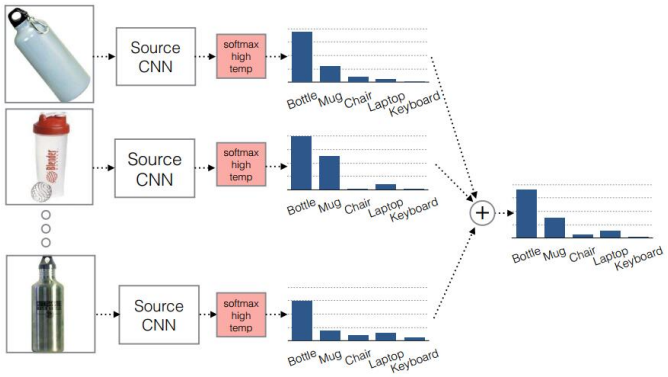


图 41: Softlabel 示意图

适应和对抗学习。JAN 方法将只对数据进行自适应的方式推广到了对类别的自适应，提出了 JMMD 度量 (Joint MMD)。图 42是 JAN 的示意图。

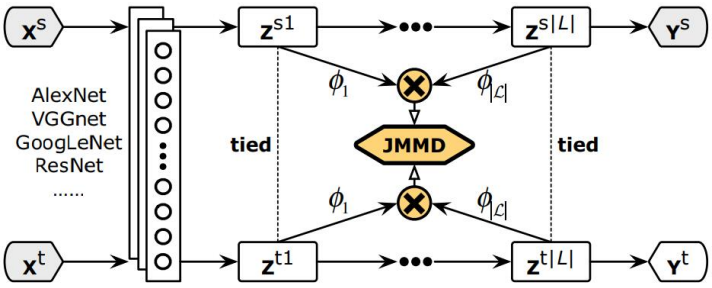


图 42: JAN 方法示意图

5. AdaBN

与上述工作选择在已有网络层中增加适配层不同的是，北京大学的 Haoyang Li 和图森科技的 Naiyan Wang 等人提出了 AdaBN(Adaptive Batch Normalization) [Li et al., 2018]，通过在归一化层加入统计特征的适配，从而完成迁移。

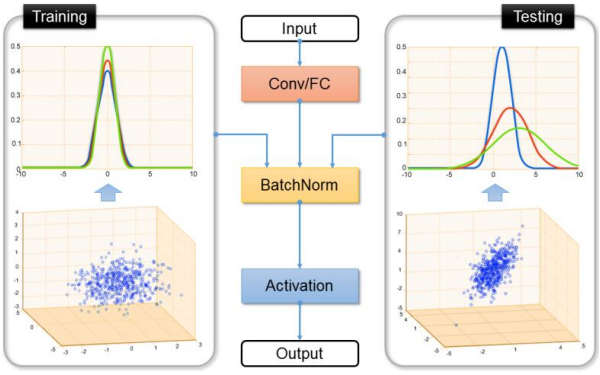


图 43: AdaBN 方法示意图

AdaBN 对比其他方法，实现相当简单。并且，方法本身不带有任何额外的参数。在许多公开数据集上都取得了很好的效果。

9.3.3 小结

基于深度网络进行迁移学习，其核心在于，找到网络需要进行自适应的层，并且对这些层加上自适应的损失度量。越来越多的研究者开始使用深度网络进行迁移学习 [Long et al., 2016, Zhuo et al., 2017, Zhuang et al., 2015, Sun and Saenko, 2016, Wei et al., 2016a, Luo et al., 2017]。在这其中，几乎绝大多数方法都采用了卷积神经网络，在已训练好的模型 (如 AlexNet、Inception、GoogLeNet、Resnet 等) 上进行迁移。

特别地，最近意大利的学者 Carlucci 等人在 2017 年计算机视觉领域顶级会议 ICCV 上提出了自动深度网络自适应层 (AutoDIAL, Automatic DomaIn Alignment Layers) [Carlucci et al., 2017]。该方法可以很简单地被加入现有的深度网络中，实现自动的自适应学习，使得深度网络的迁移更便捷。

由于篇幅和时间所限，我们未对所有新出现的研究工作给予介绍，仅在这里介绍了其中最具代表性的几种方法。但是绝大多数方法的原理都和我们已介绍过的大同小异。请读者持续关注最新发表的研究成果。

9.4 深度对抗网络迁移

生成对抗网络 GAN(Generative Adversarial Nets) [Goodfellow et al., 2014] 是目前人工智能领域最炙手可热的概念之一。其也被深度学习领军人物 Yann Lecun 评为近年来最令人欣喜的成就。由此发展而来的对抗网络，也成为了提升网络性能的利器。本小节介绍深度对抗网络用于解决迁移学习问题方面的基本思路以及代表性研究成果。

9.4.1 基本思路

GAN 受到自博弈论中的二人零和博弈 (two-player game) 思想的启发而提出。它一共包括两个部分：一部分为生成网络 (Generative Network)，此部分负责生成尽可能地以假乱真的样本，这部分被成为生成器 (Generator)；另一部分为判别网络 (Discriminative Network)，此部分负责判断样本是真实的，还是由生成器生成的，这部分被成为判别器 (Discriminator)。生成器和判别器的互相博弈，就完成了对抗训练。

GAN 的目标很明确：生成训练样本。这似乎与迁移学习的大目标有些许出入。然而，由于在迁移学习中，天然地存在一个源领域，一个目标领域，因此，我们可以免去生成样本的过程，而直接将其中一个领域的数据 (通常是目标域) 当作是生成的样本。此时，生成器的职能发生变化，不再生成新样本，而是扮演了特征提取的功能：不断学习领域数据的特征，使得判别器无法对两个领域进行分辨。这样，原来的生成器也可以称为特征提取器 (Feature Extractor)。

通常用 G_f 来表示特征提取器，用 G_d 来表示判别器。

正是基于这样的领域对抗的思想，深度对抗网络可以被很好地运用于迁移学习问题中。

与深度网络自适应迁移方法类似，深度对抗网络的损失也由两部分构成：网络训练的损失 ℓ_c 和领域判别损失 ℓ_d：

$$\ell = \ell_c(\mathcal{D}_s, \mathbf{y}_s) + \lambda \ell_d(\mathcal{D}_s, \mathcal{D}_t) \tag{9.7}$$

9.4.2 核心方法

1. DANN

Yaroslav Ganin 等人 [Ganin et al., 2016] 首先在神经网络的训练中加入了对抗机制，作者将他们的网络称之为 DANN(Domain-Adversarial Neural Network)。在此研究中，网络的学习目标是：生成的特征尽可能帮助区分两个领域的特征，同时使得判别器无法对两个领域的差异进行判别。该方法的领域对抗损失函数表示为：

$$\ell_d = \max \left[-\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z) - \frac{1}{n'} \sum_{i=n+1}^{N} \mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z) \right] \tag{9.8}$$

其中的 $\mathcal{L}_d$ 表示为

$$\mathcal{L}_d(G_d(G_f(\mathbf{x}_i)), d_i) = d_i \log \frac{1}{G_d(G_f(\mathbf{x}_i))} + (1 - d_i) \log \frac{1}{G_d(G_f(\mathbf{x}_i))} \tag{9.9}$$

2. DSN

来自 Google Brain 的 Bousmalis 等人通过提出 DSN 网络 (Domain Separation Networks) [Bousmalis et al., 2016] 对 DANN 进行了扩展。DSN 认为，源域和目标域都由两部分构成：公共部分和私有部分。公共部分可以学习公共的特征，私有部分用来保持各个领域独立的特性。DSN 进一步对损失函数进行了定义：

$$\ell = \ell_{task} + \alpha \ell_{recon} + \beta \ell_{difference} + \gamma \ell_{similarity} \tag{9.10}$$

除去网络的常规训练损失 ℓ_{task} 外，其他损失的含义如下：

- ℓ_{recon}: 重构损失，确保私有部分仍然对学习目标有作用
- $\ell_{difference}$: 公共部分与私有部分的差异损失
- $\ell_{similarity}$: 源域和目标域公共部分的相似性损失

图 44是 DSN 方法的示意图。

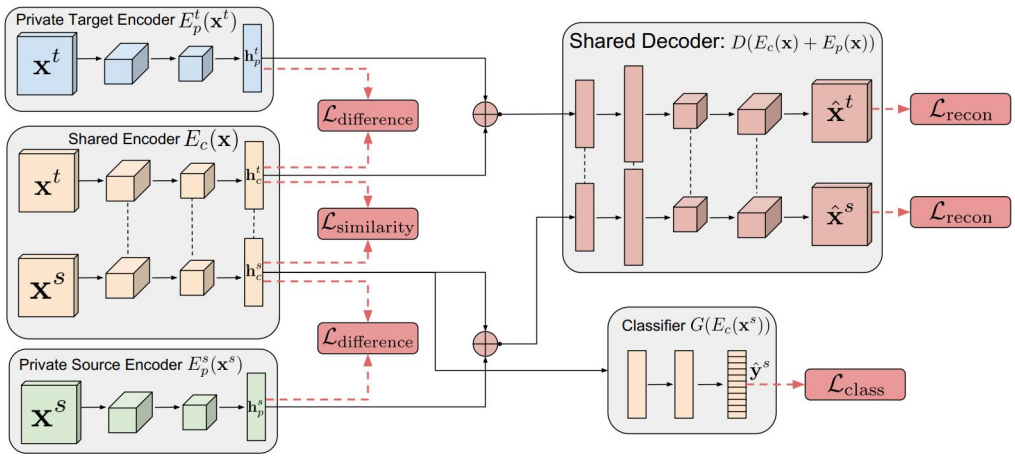


图 44: DSN 方法示意图

DDC 方法的作者、加州大学伯克利分校的 Tzeng 等人在 2017 年发表于计算机视觉顶级会议 CVPR 上的文章提出了 ADDA 方法 (Adversarial Discriminative Domain Adaptation) [Tzeng et al., 2017]。ADDA 是一个通用的框架，现有的很多方法都可被看作是 ADDA 的特例。上海交通大学的研究者们用 Wasserstein GAN 进行迁移学习 [Shen et al., 2018]，

Liu 等人提出了 Coupled GAN 用于迁移学习 [Liu and Tuzel, 2016]。这些工作都大体上按照之前思路进行。

3. SAN

清华大学龙明盛团队 2018 年发表在计算机视觉顶级会议 CVPR 上的文章提出了一个选择性迁移网络 (Partial Transfer Learning)。作者认为，在大数据时代，通常我们会有大量的源域数据。这些源域数据比目标域数据，在类别上通常都是丰富的。比如基于 ImageNet 训练的图像分类器，必然是针对几千个类别进行的分类。我们实际用的时候，目标域往往只是其中的一部分类别。这样就会带来一个问题：那些只存在于源域中的类别在迁移时，会对迁移结果产生负迁移影响。

这种情况通常来说是非常普遍的。因此，就要求相应的迁移学习方法能够对目标域，选择相似的源域样本 (类别)，同时也要避免负迁移。但是目标域通常是没有标签的，不知道和源域中哪个类别更相似。作者指出这个问题叫做 partial transfer learning。这个 partial，就是只迁移源域中那部分和目标域相关的样本。图 45展示了部分迁移学习的思想。

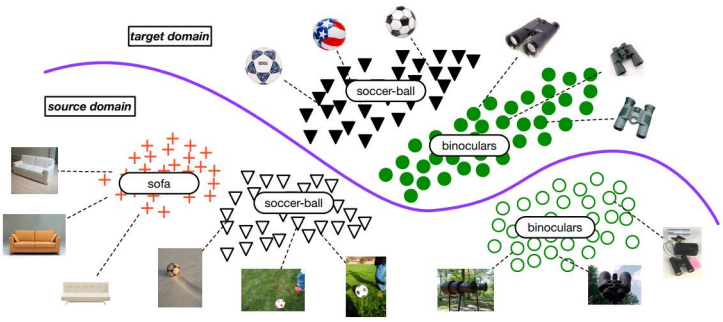


图 45: 部分迁移学习示意图

作者提出了一个叫做 Selective Adversarial Networks (SAN) [Cao et al., 2017] 的方法来处理 partial transfer 问题。在 partial 问题中，传统的对抗网络不再适用。所以就需要对进行一些修改，使得它能够适用于 partial 问题。

因为不知道目标域的标签，也就没办法知道到底是源域中哪些类是目标域的。为了达到这个目的，作者对目标域按照类别分组，把原来的一整个判别器分成了 $|\mathcal{C}_s|$ 个：G_d^k，每一个子判别器都对它所在的第 k 个类进行判别。作者观察到了这样的事实：对于每个数据点 $\mathbf{x}_i$ 来说，分类器的预测结果 $\hat{\mathbf{y}}_i$ 其实是对于整个类别空间的一个概率分布。因此，在进行对抗时，需要考虑每个样本属于每个类别的影响。这个影响就是由概率来刻画。所以作者提出了一个概率权重的判别器：

$$L_d' = \frac{1}{n_s + n_t} \sum_{k=1}^{|\mathcal{C}_s|} \sum_{\mathbf{x}_i \in \mathcal{D}_s + \mathcal{D}_t} \hat{y}_i^k L_d^k(G_d^k(G_f(\mathbf{x}_i)), d_i) \tag{9.11}$$

上面这个式子能很好地在 partial transfer 情景下，避免负迁移。这种约束是样本级别的，就是可以控制尽可能让更相似的样本参与迁移。除此之外，作者还介绍了一个类别级别的约束，可以很好地避免不在目标域中的那些类别不参与迁移。于是，进一步地，变成了下面的约束

$$L_d' = \frac{1}{n_s + n_t} \sum_{k=1}^{|\mathcal{C}_s|} \sum_{\mathbf{x}_i \in \mathcal{D}_s + \mathcal{D}_t} (\frac{1}{n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_t} \hat{y}_i^k) L_d^k(G_d^k(G_f(\mathbf{x}_i)), d_i) \tag{9.12}$$

上面这个式子比较依赖于之前说的每个样本的预测概率。为了消除这个影响，作者又在目标域上加了一项熵最小化

$$E = \frac{1}{n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_t} H(G_y(G_f(\mathbf{x}_i))) \tag{9.13}$$

4. DAAN

最近，Yu 等人在 [Yu et al., 2019] 中将动态分布适配的概念进一步扩展到了对抗网络中，证明了对抗网络中同样存在边缘分布和条件分布不匹配的问题。作者提出一个动态对抗适配网络 DAAN (Dynamic Adversarial Adaptation Networks) 来解决对抗网络中的动态分布适配问题，取得了当前的最好效果。图 46展示了 DAAN 的架构。

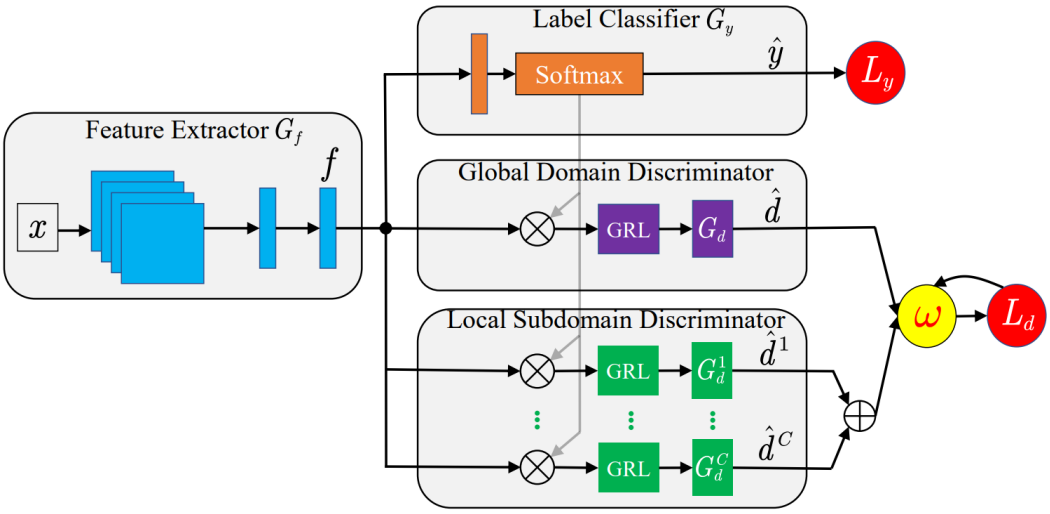


图 46: 动态对抗适配网络 DAAN 结构示意图

9.4.3 小结

使用对抗网络进行迁移学习是近年来的研究热点。我们期待在这个领域会有越来越多的工作发表。

10 上手实践

以上对迁移学习基本方法的介绍还只是停留在算法的阶段。对于初学者来说，不仅需要掌握基础的算法知识，更重要的是，需要在实验中发现问题。本章的目的是为初学者提供一个迁移学习上手实践的介绍。通过一步步编写代码、下载数据，完成迁移学习任务。在本部分，我们以迁移学习中最为流行的图像分类为实验对象，在流行的 Office+Caltech10 数据集上完成。

迁移学习方法主要包括：传统的非深度迁移、深度网络的 finetune、深度网络自适应、以及深度对抗网络的迁移。教程的目的是抛砖引玉，帮助初学者快速入门。由于网络上已有成型的深度网络的 finetune、深度网络自适应、以及深度对抗网络的迁移教程，因此我们不再叙述这些方法，只在这里介绍非深度方法的教程。其他三种方法的地址分别是：

- 深度网络的 finetune：用 Pytorch 对 Alexnet 和 Resnet 进行微调、使用 PyTorch 进行 finetune
- 深度网络的自适应：DDC/DCORAL 方法的 Pytorch 代码
- 深度对抗网络迁移：DANN 方法

更多深度迁移方法的代码，请见https://github.com/jindongwang/transferlearning/tree/master/code/deep。

在众多的非深度迁移学习方法中，我们选择最经典的迁移方法之一、发表于 IEEE TNN 2011 的 TCA(Transfer Component Analysis) [Pan et al., 2011] 方法进行实践。为了便于学习，我们同时用 Matlab 和 Python 实现了此代码。代码的链接为https://github.com/jindongwang/transferlearning/tree/master/code/traditional/TCA。下面我们对代码进行简单讲解。

10.1 TCA 方法代码实现

10.1.1 Matlab

1. 数据获取

由于我们要测试非深度方法，因此，选择 SURF 特征文件作为算法的输入。SURF 特征文件可以从网络上[2]下载。下载到的文件主要包含 4 个.mat 文件：Caltech.mat, amazon.mat, webcam.mat, dslr.mat。它们恰巧对应 4 个不同的领域。彼此之间两两一组，就是一个迁移学习任务。每个数据文件包含两个部分：fts 为 800 维的特征，labels 为对应的标注。在测试中，我们选择由 Caltech.mat 作为源域，由 amazon.mat 作为目标域。Office+Caltech10 数据集的介绍可以在本手册的第 13.4部分找到。

我们对数据进行加载并做简单的归一化，将最后的数据存入 X_s, Y_s, X_t, Y_t 这四个变量中。这四个变量分别对应源域的特征和标注、以及目标域的特征和标注。代码如下：

```
load('Caltech.mat');      % source domain
fts = fts ./ repmat(sum(fts,2),1,size(fts,2));
Xs = zscore(fts,1);     clear fts
Ys = labels;            clear labels

load('amazon.mat');     % target domain
```

[2] https://pan.baidu.com/s/1bp4g7Av

```
fts = fts ./ repmat(sum(fts,2),1,size(fts,2));
Xt = zscore(fts,1);      clear fts
Yt = labels;             clear labels
```

Matlab 加载数据

2. 算法精炼

TCA 主要进行边缘分布自适应。通过整理化简，TCA 最终的求解目标是：

$$\left(\mathbf{XMX}^{\top}+\lambda\mathbf{I}\right)\mathbf{A}=\mathbf{XHX}^{\top}\mathbf{A}\Phi \tag{10.1}$$

上述表达式可以通过 Matlab 自带的 eigs() 函数直接求解。**A** 就是我们要求解的变换矩阵。下面我们需要明确各个变量所指代的含义：

- **X**: 由源域和目标域数据共同构成的数据矩阵
- C: 总的类别个数。在我们的数据集中，$C=10$
- $\mathbf{M}_c$: MMD 矩阵。当 $c=0$ 时为全 MMD 矩阵；当 $c>1$ 时对应为每个类别的矩阵。
- **I**：单位矩阵
- λ：平衡参数，直接给出
- **H**: 中心矩阵，直接计算得出
- Φ: 拉格朗日因子，不用理会，求解用不到

3. 编写代码

我们直接给出精炼后的源码：

```
function [X_src_new,X_tar_new,A] = TCA(X_src,X_tar,options)
% The is the implementation of Transfer Component Analysis.
% Reference: Sinno Pan et al. Domain Adaptation via Transfer Component Analysis. TNN 2011.

% Inputs:
%%% X_src           :    source feature matrix, ns * n_feature
%%% X_tar           :    target feature matrix, nt * n_feature
%%% options         :    option struct
%%%%% lambda        :    regularization parameter
%%%%% dim           :    dimensionality after adaptation (dim <= n_feature)
%%%%% kernel_tpye   :    kernel name, choose from 'primal' | 'linear' | 'rbf'
%%%%% gamma         :    bandwidth for rbf kernel, can be missed for other kernels

% Outputs:
%%% X_src_new       :    transformed source feature matrix, ns * dim
%%% X_tar_new       :    transformed target feature matrix, nt * dim
%%% A               :    adaptation matrix, (ns + nt) * (ns + nt)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Set options
lambda = options.lambda;
dim = options.dim;
kernel_type = options.kernel_type;
gamma = options.gamma;
```

```
%% Calculate
X = [X_src',X_tar'];
X = X*diag(sparse(1./sqrt(sum(X.^2))));
[m,n] = size(X);
ns = size(X_src,1);
nt = size(X_tar,1);
e = [1/ns*ones(ns,1);-1/nt*ones(nt,1)];
M = e * e';
M = M / norm(M,'fro');
H = eye(n)-1/(n)*ones(n,n);
if strcmp(kernel_type,'primal')
[A,~] = eigs(X*M*X'+lambda*eye(m),X*H*X',dim,'SM');
Z = A' * X;
Z = Z * diag(sparse(1./sqrt(sum(Z.^2))));
X_src_new = Z(:,1:ns)';
X_tar_new = Z(:,ns+1:end)';
else
K = TCA_kernel(kernel_type,X,[],gamma);
[A,~] = eigs(K*M*K'+lambda*eye(n),K*H*K',dim,'SM');
Z = A' * K;
Z = Z*diag(sparse(1./sqrt(sum(Z.^2))));
X_src_new = Z(:,1:ns)';
X_tar_new = Z(:,ns+1:end)';
end
end

% With Fast Computation of the RBF kernel matrix
% To speed up the computation, we exploit a decomposition of the Euclidean distance (norm)
%
% Inputs:
%       ker:     'linear','rbf','sam'
%       X:       data matrix (features * samples)
%       gamma:   bandwidth of the RBF/SAM kernel
% Output:
%       K: kernel matrix
%
% Gustavo Camps-Valls
% 2006(c)
% Jordi (jordi@uv.es), 2007
% 2007-11: if/then -> switch, and fixed RBF kernel
% Modified by Mingsheng Long
% 2013(c)
% Mingsheng Long (longmingsheng@gmail.com), 2013
function K = TCA_kernel(ker,X,X2,gamma)

switch ker
case 'linear'

if isempty(X2)
K = X'*X;
else
K = X'*X2;
end

case 'rbf'

n1sq = sum(X.^2,1);
n1 = size(X,2);

```

```
if isempty(X2)
D = (ones(n1,1)*n1sq)' + ones(n1,1)*n1sq -2*X'*X;
else
n2sq = sum(X2.^2,1);
n2 = size(X2,2);
D = (ones(n2,1)*n1sq)' + ones(n1,1)*n2sq -2*X'*X2;
end
K = exp(-gamma*D);

case 'sam'

if isempty(X2)
D = X'*X;
else
D = X'*X2;
end
K = exp(-gamma*acos(D).^2);

otherwise
error(['Unsupported kernel ' ker])
end
end
```

TCA 方法的 Matlab 实现

我们将 TCA 方法包装成函数 TCA。注意到 TCA 是一个无监督迁移方法，不需要接受 label 作为参数。因此，函数共接受 3 个输入参数：

- $\mathrm{X}_{\mathrm{src}}$: 源域的特征，大小为 $n_s \times m$
- $\mathrm{X}_{\mathrm{tar}}$: 目标域的特征，大小为 $n_t \times m$
- options: 参数结构体，它包含：
  - λ: 平衡参数，可以自由给出
  - dim: 算法最终选择将数据将到多少维
  - $kerneltype$: 选择的核类型，可以选择 RBF、线性、或无核
  - γ: 如果选择 RBF 核，那么它的宽度为 γ

函数的输出包含 3 项：

- X_{srcnew}: TCA 后的源域
- X_{tarnew}: TCA 后的目标域
- A: 最终的变换矩阵

4. 测试算法

我们使用如下的代码对 TCA 算法进行测试：

```
options.gamma = 2;          % the parameter for kernel
options.kernel_type = 'linear';
options.lambda = 1.0;
options.dim = 20;
[X_src_new,X_tar_new,A] = TCA(Xs,Xt,options);

```

```
% Use knn to predict the target label
knn_model = fitcknn(X_src_new,Y_src,'NumNeighbors',1);
Y_tar_pseudo = knn_model.predict(X_tar_new);
acc = length(find(Y_tar_pseudo==Y_tar))/length(Y_tar);
fprintf('Acc=%0.4f\n',acc);
```

结果显示如下：

```
Acc=0.4499
```

10.1.2 Python

与 Matlab 代码类似，我们也可以用 Python 对 TCA 进行实现，其主要依赖于 Numpy 和 Scipy 两个强大的科学计算库。Python 版本的 TCA 代码如下：

```

import numpy as np
import scipy.io
import scipy.linalg
import sklearn.metrics
from sklearn.neighbors import KNeighborsClassifier


def kernel(ker, X1, X2, gamma):
K = None
if not ker or ker == 'primal':
K = X1
elif ker == 'linear':
if X2 is not None:
K = sklearn.metrics.pairwise.linear_kernel(np.asarray(X1).T, np.asarray(X2).T)
else:
K = sklearn.metrics.pairwise.linear_kernel(np.asarray(X1).T)
elif ker == 'rbf':
if X2 is not None:
K = sklearn.metrics.pairwise.rbf_kernel(np.asarray(X1).T, np.asarray(X2).T, gamma)
else:
K = sklearn.metrics.pairwise.rbf_kernel(np.asarray(X1).T, None, gamma)
return K


class TCA:
def __init__(self, kernel_type='primal', dim=30, lamb=1, gamma=1):
'''
Init func
:param kernel_type: kernel, values: 'primal' | 'linear' | 'rbf'
:param dim: dimension after transfer
:param lamb: lambda value in equation
:param gamma: kernel bandwidth for rbf kernel
'''
self.kernel_type = kernel_type
self.dim = dim
self.lamb = lamb
self.gamma = gamma

def fit(self, Xs, Xt):
```

```
    '''
    Transform Xs and Xt
    :param Xs: ns * n_feature, source feature
    :param Xt: nt * n_feature, target feature
    :return: Xs_new and Xt_new after TCA
    '''
    X = np.hstack((Xs.T, Xt.T))
    X /= np.linalg.norm(X, axis=0)
    m, n = X.shape
    ns, nt = len(Xs), len(Xt)
    e = np.vstack((1 / ns * np.ones((ns, 1)), -1 / nt * np.ones((nt, 1))))
    M = e * e.T
    M = M / np.linalg.norm(M, 'fro')
    H = np.eye(n) - 1 / n * np.ones((n, n))
    K = kernel(self.kernel_type, X, None, gamma=self.gamma)
    n_eye = m if self.kernel_type == 'primal' else n
    a, b = np.linalg.multi_dot([K, M, K.T]) + self.lamb * np.eye(n_eye), np.linalg.multi_dot([K,
        H, K.T])
    w, V = scipy.linalg.eig(a, b)
    ind = np.argsort(w)
    A = V[:, ind[:self.dim]]
    Z = np.dot(A.T, K)
    Z /= np.linalg.norm(Z, axis=0)
    Xs_new, Xt_new = Z[:, :ns].T, Z[:, ns:].T
    return Xs_new, Xt_new

    def fit_predict(self, Xs, Ys, Xt, Yt):
    '''
    Transform Xs and Xt, then make predictions on target using 1NN
    :param Xs: ns * n_feature, source feature
    :param Ys: ns * 1, source label
    :param Xt: nt * n_feature, target feature
    :param Yt: nt * 1, target label
    :return: Accuracy and predicted_labels on the target domain
    '''
    Xs_new, Xt_new = self.fit(Xs, Xt)
    clf = KNeighborsClassifier(n_neighbors=1)
    clf.fit(Xs_new, Ys.ravel())
    y_pred = clf.predict(Xt_new)
    acc = sklearn.metrics.accuracy_score(Yt, y_pred)
    return acc, y_pred


if __name__ == '__main__':
    domains = ['caltech.mat', 'amazon.mat', 'webcam.mat', 'dslr.mat']
    for i in [2]:
        for j in [3]:
            if i != j:
                src, tar = 'data/' + domains[i], 'data/' + domains[j]
                src_domain, tar_domain = scipy.io.loadmat(src), scipy.io.loadmat(tar)
                Xs, Ys, Xt, Yt = src_domain['feas'], src_domain['label'], tar_domain['feas'], tar_domain['
                    label']
                tca = TCA(kernel_type='linear', dim=30, lamb=1, gamma=1)
                acc, ypre = tca.fit_predict(Xs, Ys, Xt, Yt)
                print(acc)
```

TCA 方法的 Python 实现

5. 小结

通过以上过程，我们分别使用 Matlab 代码和 Python 代码对经典的 TCA 方法进行了实验，完成了一个迁移学习任务。其他的非深度迁移学习方法，均可以参考上面的过程。值得庆幸的是，许多论文的作者都公布了他们的文章代码，以方便我们进行接下来的研究。读者可以从 Github [3]或者相关作者的网站上获取其他许多方法的代码。

10.2 深度网络的 finetune 代码实现

本小节我们用 Pytorch 实现一个深度网络的 finetune。Pytorch 是一个较为流行的深度学习工具包，由 Facebook 进行开发，在 Github [4]上也进行了开源。

Finetune 指的是训练好的深度网络，拿来在新的目标域上进行微调。因此，我们假定读者具有基本的 Pytorch 知识，直接给出 finetune 的代码。完整的代码可以在这里 [5]找到。

我们定义一个叫做 finetune 的函数，它接受输入的一个已有模型，从目标数据中进行微调，输出最好的模型其结果。其代码如下：

```
def finetune(model, dataloaders, optimizer):
since = time.time()
best_acc = 0.0
acc_hist = []
criterion = nn.CrossEntropyLoss()
for epoch in range(1, N_EPOCH + 1):
# lr_schedule(optimizer, epoch)
print('Learning rate: {:.8f}'.format(optimizer.param_groups[0]['lr']))
print('Learning rate: {:.8f}'.format(optimizer.param_groups[-1]['lr']))
for phase in ['src', 'val', 'tar']:
if phase == 'src':
model.train()
else:
model.eval()
total_loss, correct = 0, 0
for inputs, labels in dataloaders[phase]:
inputs, labels = inputs.to(DEVICE), labels.to(DEVICE)
optimizer.zero_grad()
with torch.set_grad_enabled(phase == 'src'):
outputs = model(inputs)
loss = criterion(outputs, labels)
preds = torch.max(outputs, 1)[1]
if phase == 'src':
loss.backward()
optimizer.step()
total_loss += loss.item() * inputs.size(0)
correct += torch.sum(preds == labels.data)
epoch_loss = total_loss / len(dataloaders[phase].dataset)
epoch_acc = correct.double() / len(dataloaders[phase].dataset)
acc_hist.append([epoch_loss, epoch_acc])
print('Epoch: [{:02d}/{:02d}]---{}, loss: {:.6f}, acc: {:.4f}'.format(epoch, N_EPOCH, phase,
    epoch_loss,
epoch_acc))
if phase == 'tar' and epoch_acc > best_acc:
best_acc = epoch_acc
print()
fname = 'finetune_result' + model_name + \
```

[3] https://github.com/jindongwang/transferlearning/tree/master/code

[4] https://github.com/pytorch/pytorch

[5] https://github.com/jindongwang/transferlearning/tree/master/code/deep/finetune_AlexNet_ResNet

```
38 str(LEARNING_RATE) + str(args.source) + \
   '-' + str(args.target) + '.csv'
40 np.savetxt(fname, np.asarray(a=acc_hist, dtype=float), delimiter=',',
   fmt='%.4f')
42 time_pass = time.time() - since
   print('Training complete in {:.0f}m {:.0f}s'.format(
44 time_pass // 60, time_pass % 60))

46 return model, best_acc, acc_hist
```

深度网络的 finetune 代码实现

其中，model 可以是由任意深度网络训练好的模型，如 Alexnet、Resnet 等。

另外，有很多任务也需要用到深度网络来提取深度特征以便进一步处理。我们也进行了实现，代码在https://github.com/jindongwang/transferlearning/blob/master/code/feature_extractor中。

10.3 深度网络自适应代码

我们仍然以 Pytorch 为例，实现深度网络的自适应。具体地说，实现经典的 DDC (Deep Domain Confusion) [Tzeng et al., 2014] 方法和与其类似的 DCORAL (Deep CORAL) [Sun and Saenko, 2016] 方法。

此网络实现的核心是：如何正确计算 DDC 中的 MMD 损失、以及 DCORAL 中的 CORAL 损失，并且与神经网络进行集成。此部分对于初学者难免有一些困惑。如何输入源域和目标域、如何进行判断？因此，我们认为此部分应该是深度迁移学习的基础代码，读者应该努力地进行学习和理解。

网络结构

首先我们要定义好网络的架构，其应该是来自于已有的网络结构，如 Alexnet 和 Resnet。但不同的是，由于要进行深度迁移适配，因此，输出层要和 finetune 一样，和目标的类别数相同。其二，由于要进行距离的计算，我们需要加一个叫做 bottleneck 的层，用来将最高维的特征进行降维，然后进行距离计算。当然，bottleneck 层不加尚可。

我们的网络结构如下所示：

```
1 import torch.nn as nn
  import torchvision
3 from Coral import CORAL
  import mmd
5 import backbone


7

  class Transfer_Net(nn.Module):
9 def __init__(self, num_class, base_net='resnet50', transfer_loss='mmd', use_bottleneck=True,
       bottleneck_width=256, width=1024):
  super(Transfer_Net, self).__init__()
11 self.base_network = backbone.network_dict[base_net]()
  self.use_bottleneck = use_bottleneck
13 self.transfer_loss = transfer_loss
  bottleneck_list = [nn.Linear(self.base_network.output_num(
15 ), bottleneck_width), nn.BatchNorm1d(bottleneck_width), nn.ReLU(), nn.Dropout(0.5)]
  self.bottleneck_layer = nn.Sequential(*bottleneck_list)
17 classifier_layer_list = [nn.Linear(self.base_network.output_num(), width), nn.ReLU(), nn.
       Dropout(0.5),
  nn.Linear(width, num_class)]
```

```
self.classifier_layer = nn.Sequential(*classifier_layer_list)

self.bottleneck_layer[0].weight.data.normal_(0, 0.005)
self.bottleneck_layer[0].bias.data.fill_(0.1)
for i in range(2):
self.classifier_layer[i * 3].weight.data.normal_(0, 0.01)
self.classifier_layer[i * 3].bias.data.fill_(0.0)

def forward(self, source, target):
source = self.base_network(source)
target = self.base_network(target)
source_clf = self.classifier_layer(source)
if self.use_bottleneck:
source = self.bottleneck_layer(source)
target = self.bottleneck_layer(target)
transfer_loss = self.adapt_loss(source, target, self.transfer_loss)
return source_clf, transfer_loss

def predict(self, x):
features = self.base_network(x)
clf = self.classifier_layer(features)
return clf
```

深度迁移网络代码实现

其中 Transfer Net 是整个网络的模型定义。它接受参数有：

- num class: 目标域类别数
- base net: 主干网络，例如 Resnet 等，也可以是自己定义的网络结构
- Transfer loss: 迁移的损失，比如 MMD 和 CORAL，也可以是自己定义的损失
- use bottleneck: 是否使用 bottleneck
- bottleneck width: bottleneck 的宽度
- width: 分类器层的 width

迁移损失定义

迁移损失是核心。其定义如下：

```

 def adapt_loss(self, X, Y, adapt_loss):
"""Compute adaptation loss, currently we support mmd and coral
Arguments:
X {tensor} — source matrix
Y {tensor} — target matrix
adapt_loss {string} — loss type, 'mmd' or 'coral'. You can add your own loss
Returns:
[tensor] — adaptation loss tensor
"""
if adapt_loss == 'mmd':
mmd_loss = mmd.MMD_loss()
loss = mmd_loss(X, Y)
elif adapt_loss == 'coral':
loss = CORAL(X, Y)
else:
loss = 0
```

```
return loss
```

深度迁移网络代码实现

其中的 MMD 和 CORAL 是自己实现的两个 loss，MMD 对应 DDC 方法，CORAL 对应 DCORAL 方法。其代码在上述 github 中可以找到，我们不再赘述。

训练

训练时，我们一次输入一个 batch 的源域和目标域数据。为了方便，我们使用 pytorch 自带的 dataloader。

```
def train(source_loader, target_train_loader, target_test_loader, model, optimizer, CFG):
len_source_loader = len(source_loader)
len_target_loader = len(target_train_loader)
train_loss_clf = utils.AverageMeter()
train_loss_transfer = utils.AverageMeter()
train_loss_total = utils.AverageMeter()
for e in range(CFG['epoch']):
model.train()
iter_source, iter_target = iter(
source_loader), iter(target_train_loader)
n_batch = min(len_source_loader, len_target_loader)
criterion = torch.nn.CrossEntropyLoss()
for i in range(n_batch):
data_source, label_source = iter_source.next()
data_target, _ = iter_target.next()
data_source, label_source = data_source.to(
DEVICE), label_source.to(DEVICE)
data_target = data_target.to(DEVICE)

optimizer.zero_grad()
label_source_pred, transfer_loss = model(data_source, data_target)
clf_loss = criterion(label_source_pred, label_source)
loss = clf_loss + CFG['lambda'] * transfer_loss
loss.backward()
optimizer.step()
train_loss_clf.update(clf_loss.item())
train_loss_transfer.update(transfer_loss.item())
train_loss_total.update(loss.item())
if i % CFG['log_interval'] == 0:
print('Train Epoch: [{}/{} ({:02d}%)], cls_Loss: {:.6f}, transfer_loss: {:.6f}, total_Loss:
    {:.6f}'.format(
e + 1,
CFG['epoch'],
int(100. * i / n_batch), train_loss_clf.avg, train_loss_transfer.avg, train_loss_total.avg))

# Test
test(model, target_test_loader)
```

深度迁移网络代码实现

11 迁移学习前沿

从我们上述介绍的多种迁移学习方法来看，领域自适应 (Domain Adaptation) 作为迁移学习的重要分类，在近年来已经取得了大量的研究成果。但是，迁移学习仍然是一个活跃的领域，仍然有大量的问题没有被很好地解决。

本节我们简要介绍一些迁移学习领域较新的研究成果。并且，管中窥豹，展望迁移学习未来可能的研究方向。

11.1 机器智能与人类经验结合迁移

机器学习的目的是让机器从众多的数据中发掘知识，从而可以指导人的行为。这样看来，似乎“全自动”是我们的终极目标。我们理想中的机器学习系统，似乎就应该完全不依赖于人的干预，靠算法和数据就能完成所有的任务。Google Deepmind 公司最新发布的 AlphaZero [Silver et al., 2017] 就实现了这样的愿景：*算法完全不依赖于人提供知识，从零开始掌握围棋知识，最终打败人类围棋冠军*。随着机器学习的发展，似乎人的角色也会越来越不重要。

然而，在目前看来，机器想完全不依赖于人的经验，就必须付出巨大的时间和计算代价。普通人也许根本无法掌握这样的能力。那么，如果在机器智能中，特别是迁移学习的机器智能中，加入人的经验，可以大幅度提高算法的训练水平，这岂不是我们喜闻乐见的？

来自斯坦福大学的研究人员 2017 年发表在人工智能顶级会议 AAAI 上的研究成果就率先实践了这一想法 [Stewart and Ermon, 2017]。研究人员提出了一种无需人工标注的神经网络，对视频数据进行分析预测。在该成果中，研究人员的目标是用神经网络预测扔出的枕头的下落轨迹。不同于传统的神经网络需要大量标注，该方法完全不使用人工标注。取而代之的是，将人类的知识赋予神经网络。

我们都知道，抛出的物体往往会沿着*抛物线*的轨迹进行运动。这就是研究人员所利用的核心知识。计算机对于这一点并不知情。因此，在网络中，如果加入抛物线这一基本的先验知识，则会极大地促进网络的训练。并且，最终会取得比单纯依赖算法本身更好的效果。

我们认为将机器智能与人类经验结合起来的迁移学习应该是未来的发展方向之一。期待这方面有更多的研究成果发表。

11.2 传递式迁移学习

迁移学习的核心是找到两个领域的相似性。这是成功进行迁移的保证。但是，假如我们的领域数据本身就不存在相似性，或者相似性极小，这时候就很容易出现负迁移。负迁移是迁移学习研究中极力需要避免的。

我们由两个领域的相似性推广开来，其实世间万事万物都有一定的联系。表面上看似无关的两个领域，它们也可以由中间的领域构成联系。也就是一种传递式的相似性。例如，领域 A 和领域 B 从表面上看，完全不相似。那么，是否可以找到中间的一个领域 C，领域 C 与 A 和 B 都有一定的相似性？这样，知识原来不能直接从领域 A 迁移到领域 B，加入 C 以后，就可以先从 A 迁移到 C，再从 C 迁移到 B。这就是传递迁移学习。

香港科技大学杨强教授的团队率先在 2015 年数据挖掘顶级会议 KDD 上提出了这一概念：Transitive transfer learning [Tan et al., 2015]。随后，作者又进行了进一步的扩展，将三个领域的迁移，扩展到多个领域。这也是符合我们认知的：原先完全不相似的两个领域，如果它们中间存在若干领域都与这两个领域相似，那么就可以构成一条相似性链条，知识

就可以进行链式的迁移。作者提出了远领域的迁移学习 (Distant Domain Transfer Learning) [Tan et al., 2017]，用卷积神经网络解决了这一问题。

在远领域迁移学习中，作者做出了看起来并不符合常人认知的实验：由人脸图片训练好的分类器，迁移识别飞机图像。研究人员采用了人脸和飞机中间的一系列类别，例如头像、头盔、水壶、交通工具等。在实验中，算法自动地选择相似的领域进行迁移。结果表明，在初始迁移阶段，算法选择的大多是与源领域较为相似的类别；随着迁移的进行，算法会越来越倾向于选择与目标领域相似的类别。这也是符合我们的基本认知的。最终的对比实验表明，这种远领域的迁移学习比直接训练分类器的精度会有极大的提升。图 47展示了知识迁移的过程。

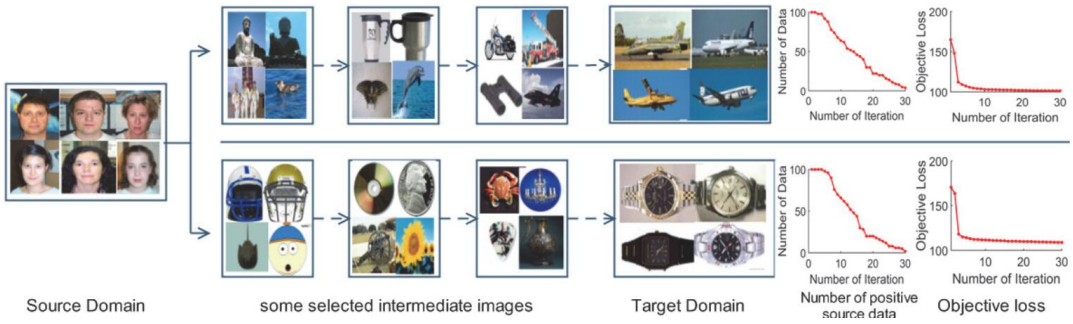


图 47: 远领域迁移学习示意图

传递迁移学习目前的研究成果还十分稀少。我们期待这一领域会有更多好的成果出现。

11.3 终身迁移学习

我们在进行迁移学习时，往往不知道应该选择怎么样的算法。通常都通过人为地不断尝试来确定要用的方法。这个过程无疑是浪费时间，而且充满了不确定性。已有的迁移学习方法，基本都是这样。我们在拿到一个新问题时，如何选择迁移学习算法达到最好的效果？从人的学习过程中来说，人总是可以从以前的经验中学习知识。那么，既然我们已经实验了很多次迁移学习算法，我们能不能让机器也可以从我们这些实验中选择知识？这是符合我们人类认知的：不可能每遇到一个新问题，我们就从头开始做吧？

同样是来自香港科技大学杨强教授团队就开始了这一方面的研究工作。他们提出一种学习迁移 (L2T, Learning to Transfer) 的框架 [Wei et al., 2017]，解决何时迁移、要迁移什么、怎么迁移的问题。方法分为两个部分：从已有的迁移学习方法和结果中学习迁移的经验，然后再把这些学习到的经验应用到新来的数据。

首先要明确学习目标。跟以前的迁移学习方法都有所不同，以往的方法都是要学习最好的迁移函数，而这个问题的目标是要使得方法尽可能地具有泛化能力。因此，它的学习目标是：**以往的经验**！对的，就是经验。这也是符合我们人类认知的。我们一般都是知道的越多，这个人越厉害 (当然，《权利的游戏》里的 Jon Snow 除外，因为他什么也不知道)。因此，这个方法的目标就是要尽可能多地从以往的迁移知识中学习经验，使之对于后来的问题具有最好的泛化能力。

那么什么是迁移的经验？这个在文中叫做 transfer learning experience。作者这样定义：$Ee = (S_e, T_e, a_e, l_e)$。其中，$S_e, T_e$ 分别是源域和目标域，这个我们都知道。a_e 表示一个迁移学习算法，这个算法有个下标叫 e，表示它是第 e 种算法。与之对应，选择了这种算法，它对不迁移情况下的表现有个提升效果，这个效果就叫做 le。

总结一下，什么叫迁移学习的经验？就是说，在一对迁移任务中，我选择了哪种算法后，这种算法对于我任务效果有多少提升。这个东西，就叫做迁移！这和我们人类学习也是具有相似性的：人们常说，失败是成功之母，爱迪生说，我实验了 2000 多种材料做灯泡都是失败的，但是我最起码知道了这 2000 多种材料不适合做灯泡！这就是人类的经验！我们人类就是从跌倒中爬起，从失败中总结教训，然后不断进步。学习算法也可以！

后面的过程就是，综合性地学习这个迁移过程，使得算法根据以往的经验，得到一个特征变换矩阵 **W**。学习到了这个变换矩阵以后，下一步的工作就是要把学习到的东西应用于新来的数据。如何针对新来的数据进行迁移？我们本能地要利用刚刚学习到的这个 **W**。但是不要忘了，这个变换矩阵只是对旧的那些经验学习到的，对新的数据可能效果不好，不能直接用。怎么办？我们要更新它！

这里要注意的是，针对新来的数据，我们的这个变换矩阵应该是有所改变的：这也对，数据变了，当然变换矩阵就要变。那么，如何更新？作者在这里提出的方法是，新的矩阵应该是能在新的数据上表现效果最好的那个。这时，我们的问题就完成了。

图 48简要表示了该算法的学习过程。

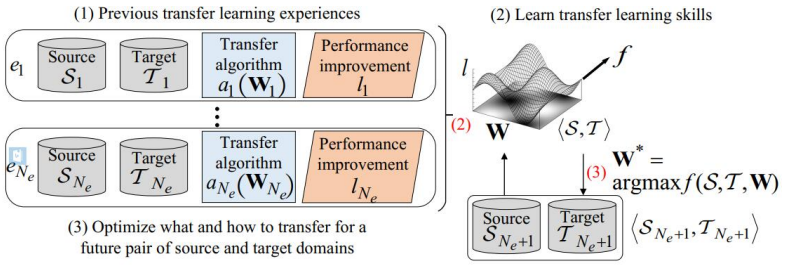


图 48: 终身迁移学习示意图

这有点类似增量学习的工作：模型针对新数据不断更新优化。这部分的研究才刚刚开始。

11.4 在线迁移学习

我们都知道迁移学习可以用来解决训练数据缺失的问题，很多迁移学习方法都获得了长足的进步。给定一个要学习的目标域数据，我们可以用已知标签的源域数据来给这个目标域数据构造一个分类器。但是这些方法都存在很大的一个问题：它们都是采用离线方式 (offline) 进行的。什么是离线方式？就是说，一开始，源域和目标域数据都是给出来的，我们直接做完迁移，这个过程就结束了。We are done。

但是真实的应用往往不是这样的：数据往往是一点一点源源不断送来的。也就是说，我们一开始的时候，也许只有源域数据，目标域数据要一点一点才能过来。这就是所谓的“在线迁移学习”。这个概念脱胎于“在线学习”的模式，在线学习是机器学习中一个重要的研究概念。

就目前来说，在线迁移学习方面的工作较少。第一篇在线迁移学习的工作由新加坡管理大学的 Steven Hoi 发表在 2010 年的机器学习顶级会议 ICML 上。作者提出了 OTL 框架 [Zhao and Hoi, 2010]，可以对同构和异构数据很好地进行迁移学习。

近年来，研究者发表了一些在线迁移学习相关的文章。其中包括在多个源域和目标域上的 OTL [Wu et al., 2017, Yan et al., 2017]，在线特征选择迁移变换 [Wang et al., 2014, Zhang et al., 2017b]，在线样本集成迁移 [Gao et al., 2012, Patil and Phursule, 2013] 等。

特别地，[Jaini et al., 2016] 提出了用贝叶斯的方法学习在线的 HMM 迁移学习模型，并应用于行为识别、睡眠监测，以及未来流量分析。这是一篇有代表性的应用工作。

总结来看，目前在在线迁移学习方面的研究工作总体较少，发展空间巨大。在可以预见的未来，我们期待更多的研究者可以从事这一领域的研究。将深度网络、对抗学习结合入在线迁移学习，使得这一领域的发展越来越好。

11.5 迁移强化学习

Google 公司的 AlphaGo 系列在围棋方面的成就让强化学习这一术语变得炙手可热。用深度神经网络来进行强化学习也理所当然地成为了研究热点之一。不同于传统的机器学习需要大量的标签才可以训练学习模型，强化学习采用的是边获得样例边学习的方式。特定的反馈函数决定了算法的最优决策。

深度强化学习同时也面临着重大的挑战：没有足够的训练数据。在这个方面，迁移学习却可以利用其他数据上训练好的模型帮助训练。尽管迁移学习已经被应用于强化学习 [Taylor and Stone, 2009]，但是它的发展空间仍然还很大。强化学习在自动驾驶、机器人、路径规划等领域正发挥着越来越重要的作用。我们期待在未来有更多的研究成果可以问世。

11.6 迁移学习的可解释性

深度学习取得众多突破性成果的同时，其面临的可解释性不强却始终是一个挑战。现有的深度学习方法还停留在” 黑盒子” 阶段，无法产生足够有说服力的解释。同样的，迁移学习也有这个问题。即使世间万物都有联系，它们更深层次的关系也尚未得到探索。领域之间的相似性也正如同海森堡” 测不准原理” 一般无法给出有效的结论。为什么领域 A 和领域 B 更相似，而和领域 C 较不相似？目前也只是停留在经验阶段，缺乏有效的理论证明。

另外，迁移学习算法也存在着可解释性弱的问题。现有的算法均只是完成了一个迁移学习任务。但是在学习过程中，知识是如何进行迁移的，这一点还有待进一步的实验和理论验证。最近，澳大利亚悉尼大学的研究者们发表在国际人工智能联合会 IJCAI 2017 上的研究成果有助于理解特征是如何迁移的 [Liu et al., 2017]。

用深度网络来做迁移学习，其可解释性同样有待探索。最近，Google Brain 的研究者们提出了神经网络的” 核磁共振” 现象 [6]，对神经网络的可解释性进行了有趣的探索。

[6] https://github.com/tensorflow/lucid

12 总结语

本手册是在迁移学习领域的研究经验总结。主要简明地介绍了迁移学习的基本概念、迁移学习的必要性、研究领域和基本方法。重点介绍了几大类常用的迁移学习方法：数据分布自适应方法、特征选择方法、子空间学习方法、以及目前最热门的深度迁移学习方法。除此之外，我们也结合最近的一些研究成果对未来迁移学习进行了一些展望。附录中提供了一些迁移学习领域的常用学习资源，以方便感兴趣的读者快速开始学习。

13 附录

13.1 迁移学习相关的期刊和会议

迁移学习仍然是一个蓬勃发展的研究领域。最近几年，在顶级期刊和会议上，越来越多的研究者开始发表文章，不断提出迁移学习的新方法，不断开拓迁移学习的新应用领域。

在这里，我们对迁移学习相关的国际期刊和会议作一小结，以方便初学者寻找合适的论文。这些期刊会议可以在表 4中找到。

表 4: 迁移学习相关的期刊和会议

序号	简称	全称	领域
国际期刊			
1	JMLR	Journal of Machine Learning Research	机器学习
2	MLJ	Machine Learning Journal	机器学习
3	AIJ	Artificial Intelligence Journal	人工智能
4	TKDE	IEEE Transactions on Knowledge and Data Engineering	数据挖掘
5	TIST	ACM Transactions on Intelligent Science and Technology	数据挖掘
6	PAMI	IEEE Transactions on Pattern Analysis and Machine Intelligence	计算机视觉
7	IJCV	International Journal of Computer Vision	计算机视觉
8	TIP	IEEE Transactions on Image Processing	计算机视觉
9	PR	Pattern Recognition	模式识别
10	PRL	Pattern Recognition Letters	模式识别
国际会议			
1	ICML	International Conference on Machine Learning	机器学习
2	NIPS	Annual Conference on Neural Information Processing System	机器学习
3	IJCAI	International Joint Conference on Artificial Intelligence	人工智能
4	AAAI	AAAI conference on Artificial Intelligence	人工智能
5	KDD	ACM SIGKDD Conference on Knowledge Discovery and Data Mining	数据挖掘
6	ICDM	IEEE International Conference on Data Mining	数据挖掘
7	CVPR	IEEE Conference on Computer Vision and Pattern Recognition	计算机视觉
8	ICCV	IEEE International Conference on Computer Vision	计算机视觉
9	ECCV	European Conference on Computer Vision	计算机视觉
10	WWW	International World Wide Web Conferences	文本、互联网
11	CIKM	International Conference on Information and Knowledge Management	文本分析
12	ACMMM	ACM International Conference on Multimedia	多媒体

13.2 迁移学习研究学者

这里列出了一些迁移学习领域代表性学者以及他们的最具代表性的工作，以供分享。

一般这些工作都是由他们一作，或者是由自己的学生做出来的。当然，这里所列的文章比起这些大牛发过的文章会少得多，这里仅仅列出他们最知名的工作。本部分开源在了Github [7]，会一直有更新，欢迎补充！

应用研究

1. Qiang Yang @ HKUST

[7] https://github.com/jindongwang/transferlearning/blob/master/doc/scholar_TL.md

迁移学习领域权威大牛。他所在的课题组基本都做迁移学习方面的研究。迁移学习综述《A survey on transfer learning》就出自杨强老师课题组。他的学生们：

1). Sinno J. Pan @ NTU

现为老师，详细介绍见第二条。

2). Ben Tan

主要研究传递迁移学习 (transitive transfer learning)，现在腾讯做高级研究员。代表文章：

Transitive Transfer Learning. KDD 2015.

Distant Domain Transfer Learning. AAAI 2017.

3). Derek Hao Hu

主要研究迁移学习与行为识别结合，目前在 Snap 公司。代表文章：

Transfer Learning for Activity Recognition via Sensor Mapping. IJCAI 2011.

Cross-domain activity recognition via transfer learning. PMC 2011.

Bridging domains using world wide knowledge for transfer learning. TKDE 2010.

4). Vencent Wencheng Zheng

也做行为识别与迁移学习的结合，目前在新加坡一个研究所当研究科学家。代表文章：

User-dependent Aspect Model for Collaborative Activity Recognition. IJCAI 2011.

Transfer Learning by Reusing Structured Knowledge. AI Magazine.

Transferring Multi-device Localization Models using Latent Multi-task Learning. AAAI 2008.

Transferring Localization Models Over Time. AAAI 2008.

Cross-Domain Activity Recognition. Ubicomp 2009.

Collaborative Location and Activity Recommendations with GPS History Data. WWW 2010.

5). Ying Wei

做迁移学习与数据挖掘相关的研究。代表工作：

Instilling Social to Physical: Co-Regularized Heterogeneous Transfer Learning. AAAI 2016.

Transfer Knowledge between Cities. KDD 2016.

Learning to Transfer. arXiv 2017.

其他还有很多学生都做迁移学习方面的研究，更多请参考杨强老师主页。

2. Sinno J. Pan @ NTU

杨强老师学生，比较著名的工作是 TCA 方法。现在在 NTU 当老师，一直都在做迁移学习研究。代表工作：

A Survey On Transfer Learning. TKDE 2010. [最著名的综述]

Domain Adaptation via Transfer Component Analysis. TNNLS 2011. [著名的 TCA 方法]

Cross-domain sentiment classification via spectral feature alignment. WWW 2010. [著名的 SFA 方法]

Transferring Localization Models across Space. AAAI 2008.

3. Lixin Duan @ UESTC

毕业于 NTU，现在在 UESTC 当老师。代表工作：

Domain Transfer Multiple Kernel Learning. PAMI 2012.

Visual Event Recognition in Videos by Learning from Web Data. PAMI 2012.

4. Mingsheng Long @ THU

毕业于清华大学，现在在清华大学当老师，一直在做迁移学习方面的工作。代表工作：

Dual Transfer Learning. SDM 2012.

Transfer Feature Learning with Joint Distribution Adaptation. ICCV 2013.

Transfer Joint Matching for Unsupervised Domain Adaptation. CVPR 2014.

Learning transferable features with deep adaptation networks. ICML 2015. [著名的 DAN 方法]

Deep Transfer Learning with Joint Adaptation Networks. ICML 2017.

5. Judy Hoffman @ UC Berkeley & Stanford

Feifei Li 的博士后，现在当老师。她有个学生叫做 Eric Tzeng，做深度迁移学习。代表工作：

Simultaneous Deep Transfer Across Domains and Tasks. ICCV 2015.

Deep Domain Confusion: Maximizing for Domain Invariance. arXiv 2014.

Adversarial Discriminative Domain Adaptation. CVPR 2017.

6. Fuzhen Zhuang @ ICT, CAS

中科院计算所当老师，主要做迁移学习与文本结合的研究。代表工作：

Transfer Learning from Multiple Source Domains via Consensus Regularization. CIKM 2008.

7. Kilian Q. Weinberger @ Cornell U.

现在康奈尔大学当老师。Minmin Chen是他的学生。代表工作：

Distance metric learning for large margin nearest neighbor classification. JMLR 2009.

Feature hashing for large scale multitask learning. ICML 2009.

An introduction to nonlinear dimensionality reduction by maximum variance unfolding. AAAI 2006. [著名的 MVU 方法]

Co-training for domain adaptation. NIPS 2011. [著名的 Co-training 方法]

8. Fei Sha @ USC

USC 教授。他曾经的学生Boqing Gong提出了著名的 GFK 方法。代表工作：

Connecting the Dots with Landmarks: Discriminatively Learning Domain-Invariant Features for Unsupervised Domain Adaptation. ICML 2013.

Geodesic flow kernel for unsupervised domain adaptation. CVPR 2012. [著名的 GFK 方法]

9. Mahsa Baktashmotlagh @ U. Queensland

现在当老师。主要做流形学习与 domain adaptation 结合。代表工作：

Unsupervised Domain Adaptation by Domain Invariant Projection. ICCV 2013.

Domain Adaptation on the Statistical Manifold. CVPR 2014.

Distribution-Matching Embedding for Visual Domain Adaptation. JMLR 2016.

10. Baochen Sun @ Microsoft

现在在微软。著名的 CoRAL 系列方法的作者。代表工作：

Return of Frustratingly Easy Domain Adaptation. AAAI 2016.

Deep coral: Correlation alignment for deep domain adaptation. ECCV 2016.

11. Wenyuan Dai

著名的第四范式创始人，虽然不做研究了，但是当年求学时几篇迁移学习文章至今都很高引。代表工作：

Boosting for transfer learning. ICML 2007. [著名的 TrAdaboost 方法]

Self-taught clustering. ICML 2008.

理论研究

1. Arthur Gretton @ UCL

主要做 two-sample test。代表工作：

A Kernel Two-Sample Test. JMLR 2013.

Optimal kernel choice for large-scale two-sample tests. NIPS 2012. [著名的 MK-MMD]

2. Shai Ben-David @ U.Waterloo

很多迁移学习的理论工作由他给出。代表工作：

Analysis of representations for domain adaptation. NIPS 2007.

A theory of learning from different domains. Machine Learning 2010.

3. Alex Smola @ CMU

做一些机器学习的理论工作，和上面两位合作比较多。代表工作非常多，不列了。

4. John Blitzer @ Google

著名的 SCL 方法提出者，现在也在做机器学习。代表工作：

Domain adaptation with structural correspondence learning. ECML 2007. [著名的 SCL 方法]

5. Yoshua Bengio @ U.Montreal

深度学习领军人物，主要做深度迁移学习的一些理论工作。代表工作：

Deep Learning of Representations for Unsupervised and Transfer Learning. ICML 2012.

How transferable are features in deep neural networks? NIPS 2014.

Unsupervised and Transfer Learning Challenge: a Deep Learning Approach. ICML 2012.

6. Geoffrey Hinton @ U.Toronto

深度学习领军人物，也做深度迁移学习的理论工作。

Distilling the knowledge in a neural network. NIPS 2014.

13.3 迁移学习资源汇总

- (可能是有史以来) 最全的迁移学习资料库,(文章/资料/代码/数据): https://github.com/jindongwang/transferlearning
- 迁移学习视频教程: https://www.youtube.com/watch?v=qD6iD4TFsdQ
- 知乎专栏“机器有颗玻璃心”中《小王爱迁移》系列: https://zhuanlan.zhihu.com/p/27336930, 用浅显易懂的语言深入讲解经典 + 最新的迁移学习文章
- 迁移学习与领域自适应论文分享与笔记 Paperweekly:http://www.paperweekly.site/collections/231/papers
- 迁移学习与领域自适应公开数据集 https://github.com/jindongwang/transferlearning/blob/master/doc/dataset.md

13.4 迁移学习常用算法及数据资源

为了研究及测试算法性能，收集了若干来自图像、文本、及时间序列的公开数据集。并且，为了与自己所研究的算法进行对比，收集了领域内若干较前沿的算法及其相关代码，以便后期进行算法的比较测试。

数据集：

表 5收集了迁移学习领域常用的数据集。这些数据集的详细介绍和下载地址，在 Github [8] 上可以找到。我们还在 Benchmark [9] 上提供了一些常用算法的实验结果。

表 5: 迁移学习相关的图像、文本、和时间序列数据集统计信息

序号	数据集	类型	样本数	特征数	类别数
1	USPS	字符识别	1800	256	10
2	MNIST	字符识别	2000	256	10
3	PIE	人脸识别	11554	1024	68
4	COIL20	对象识别	1440	1024	20
5	Office+Caltech	对象识别	2533	800	10
6	ImageNet	图像分类	7341	4096	5
7	VOC2007	图像分类	3376	4096	5
8	LabelMe	图像分类	2656	4096	5
9	SUN09	图像分类	3282	4096	5
10	Caltech101	图像分类	1415	4096	5
11	20newsgroup	文本分类	25804	/	6
12	Reuters-21578	文本分类	4771	/	3
13	OPPORTUNITY	行为识别	701366	27	4
14	DSADS	行为识别	2844868	27	19
15	PAMAP2	行为识别	1140000	27	18

各数据集的简要介绍：

1. 手写体识别图像数据集

MNIST 和 USPS 是两个通用的手写体识别数据集，它们被广泛地应用于机器学习算法评测的各个方面。USPS 数据集包括 7,291 张训练图片和 2,007 张测试图片，图片大小为 16×16。MNIST 数据集包括 60,000 张训练图片和 10,000 张测试图片，图片大小 28×28。USPS 和 MNIST 数据集分别服从显著不同的概率分布，两个数据集都包含 10 个类别，每个类别是 1-10 之间的某个字符。为了构造迁移学习人物，在 USPS 中随机选取 1,800 张图片作为辅助数据、在 MNIST 中随机选取 2,000 张图片作为目标数据。交换辅助领域和目标领域可以得到另一个分类任务 MNIST vs USPS。图片预处理包括: 将所有图片大小线性缩放为 16×16，每幅图片用 256 维的特征向量表征，编码了图片的像素灰度值信息。辅助领域和目标领域共享特征空间和类别空间，但数据分布显著不同。

2. 人脸识别图像数据集

PIE 代表“朝向、光照、表情”的英文单词首字母，该数据集是人脸识别的基准测试集，包括 68 个不同人物的 41,368 幅人脸照片，图片大小为 32×32，每个人物的照片由 13

[8] https://github.com/jindongwang/transferlearning/blob/master/doc/dataset.md

[9] https://github.com/jindongwang/transferlearning/blob/master/doc/benchmark.md

个同步的相机 (不同朝向)、21 个不同曝光程度拍摄。简单起见，实验中采用 PIE 的预处理集，包括 2 个不同子集 PIE1 和 PIE2，是从正面朝向的人脸照片集合 (C27) 中按照不同的光照和曝光条件随机选出。按如下方法构造分类任务 PIEI vs PIE2：将 PIE1 作为辅助领域、PIE2 作为目标领域; 交换辅助领域和目标领域可以得到分类任务 PIE2 vs PIEI。这样，辅助领域和目标领域分别由不同光照、曝光条件的人脸照片组成，从而服从显著不同的概率分布。

3. 对象识别数据集

COIL20 包含 20 个对象类别共 1,440 张图片; 每个对象类别包括 72 张图片，每张图片拍摄时对象水平旋转 5 度 (共 360 度)。每幅图片大小为 32×32，表征为 1,024 维的向量。实验中将该数据集划分为两个不相交的子集 COIL1 和 COIL2：COIL1 包括位于拍摄角度为 $[0°, 85°] \cup [180°, 265°]$(第一、三象限) 的所有图片；COIL2 包括位于拍摄角度为 $[90°, 175°] \cup [270°, 355°]$(第二、四象限) 的所有图片。这样，子集 COIL1 和 COIL2 的图片因为拍摄角度不同而服从不同的概率分布。将 COIL1 作为辅助领域、COIL2 作为目标领域，可以构造跨领域分类任务 COIL1 vs COIL2；交换辅助领域和目标领域，可以得到另外一个分类任务 COIL2 vs COIL1。

Office 是视觉迁移学习的主流基准数据集，包含 3 个对象领域 Amazon(在线电商图片)、Webcam(网络摄像头拍摄的低解析度图片)、DSLR(单反相机拍摄的高解析度图片)，共有 4,652 张图片 31 个类别标签。Caltech-256 是对象识别的基准数据集，包括 1 个对象领域 Caltech，共有 30,607 张图片 256 个类别标签。对每张图片抽取 SURF 特征，并向量化为 800 维的直方图表征，所有直方图向量都进行减均值除方差的归一化处理，直方图码表由 K 均值聚类算法在 Amazon 子集上生成。具体共有 4 个领域 C(Caltech-256), A(Amazon), W(Webcam) 和 D(DSLR)，从中随机选取 2 个不同的领域作为辅助领域和目标领域，则可构造 $4 \times 3 = 12$ 个跨领域视觉对象识别任务，如 $A \to D, A \to C, \cdots, C \to W$。

4. 大规模图像分类数据集

大规模图像分类数据集包含了来自 5 个域的图像数据：ImageNet、VOC 2007、SUN、LabelMe、以及 Caltech。它们包含 5 个类别的图像数据：鸟，猫，椅子，狗，人。对于每个域的数据，均使用 DeCaf [Donahue et al., 2014] 进行特征提取，并取第 6 层的特征作为实验使用，简称 DeCaf6 特征。每个样本有 4096 个维度。

5. 通用文本分类数据集

20-Newsgroups 数据集包含约 20,000 个文档，4 个大类分别为 comp, rec,sci 和 talk，每个大类包含 4 个子类，详细信息如表 2.2 所示。在实验中构造了 6 组跨领域二分类任务，每组任务由 4 个大类中随机选取 2 个大类构成，一个大类记为正例，另一个大类记为负例，6 个任务组具体为 comp vs rec, comp vs sci, comp vs talk, rec vs sci, rec vs talk 和、sci vs talk。每个跨领域分类任务 (包括辅助领域和目标领域) 按如下方法生成: 每个任务组 p VS 的两个大类 p 和 Q 分别包含 4 个子类 P1、P2、P3、P4 和 Q1、Q2、Q3、Q4; 随机选取 p 的两个子类 (如 P1、P2) 与 Q 的两个子类 (如 Q1、Q2) 构成辅助领域，其余子类 (P 的 P3 和 P4 和 Q3 和 Q4 构成目标领域。以上构造策略既保证辅助领域和目标领域是相关的，因为它们都来自同样的大类; 又保证辅助领域和目标领域是不同的，因为它们来自不同的子类。每个任务组 P VS Q 可以生成 36 个分类任务，总计 6 个任务组共生成 6×36 = 216 个分类任务。数据集经过文本预处理后包含 25,804 个词项特征和 15,033 个文档，每个文档由 tf-idf 向量表征。

Reuters-21578 是一个较难的文本数据集，包含多个大类和子类。其中最大 3 个大类为

orgs, people 和 place，可构造 6 个跨领域文本分类任务 orgs vs people,people vs orgs, orgs vs place, place vs orgs, people vs place 和 place vs people。

6. 行为识别公开数据集

行为识别是典型的时间序列分类任务。为了测试算法在时间序列任务上的性能，收集了 3 个公开的行为识别数据集：OPPORTUNITY、DASDS 和 PAMAP2。OPPORTUNITY 数据集包含 4 个用户在智能家居中的多种不同层次的行为。DAADS 数据集包含 8 个人的 19 种日常行为。PAMAP2 数据集包含 9 个人的 18 种日常生活行为。所有数据集均包括加速度计、陀螺仪和磁力计三种运动传感器。

算法：

收集的数据表征、迁移学习等相关领域的基准算法如表 6所示。我们还在持续更新的 Github [10]上提供了各种算法的实现代码。

表 6: 收集的公开算法

序号	算法简称	算法全称	出处
1	PCA	principal component analysis	[Fodor, 2002]
2	KPCA	kernel principal component analysis	[Fodor, 2002]
3	TCA	transfer component analysis	[Pan et al., 2011]
4	GFK	geodesic flow kernel	[Gong et al., 2012]
5	TKL	transfer kernel learning	[Long et al., 2015b]
6	TSL	transfer subspace learning	[Si et al., 2010]
7	JDA	joint distribution adaptation	[Long et al., 2013]
8	TJM	transfer joint matching	[Long et al., 2014b]
9	DAN	deep adaptation network	[Long et al., 2015a]
10	JAN	joint adaptation network	[Long et al., 2017]
11	DTMKL	domain transfer multiple kernel learning	[Duan et al., 2012]
12	JGSA	joint geometrical and statistical adaptation	[Zhang et al., 2017a]
13	SCA	scatter component analysis	[Ghifary et al., 2017]
14	ARTL	adaptation regularization	[Long et al., 2014a]
15	TrAdaBoost	transfer learning based adaboost	[Dai et al., 2007]
16	GNMF	graph regularized NMF	[Cai et al., 2011]
17	CORAL	Correlation Alignment	[Sun et al., 2016]
18	SDA	Subspace Distribution Alignment	[Sun and Saenko, 2015]

参考文献

[Baktashmotlagh et al., 2013] Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. (2013). Unsupervised domain adaptation by domain invariant projection. In *ICCV*, pages 769–776.

[Baktashmotlagh et al., 2014] Baktashmotlagh, M., Harandi, M. T., Lovell, B. C., and Salzmann, M. (2014). Domain adaptation on the statistical manifold. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2481–2488.

[10] https://github.com/jindongwang/transferlearning

[Belkin et al., 2006] Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434.

[Ben-David et al., 2010] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Vaughan, J. W. (2010). A theory of learning from different domains. *Machine learning*, 79(1-2):151–175.

[Ben-David et al., 2007] Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. (2007). Analysis of representations for domain adaptation. In *NIPS*, pages 137–144.

[Blitzer et al., 2008] Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J. (2008). Learning bounds for domain adaptation. In *Advances in neural information processing systems*, pages 129–136.

[Blitzer et al., 2006] Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128. Association for Computational Linguistics.

[Borgwardt et al., 2006] Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H.-P., Schölkopf, B., and Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57.

[Bousmalis et al., 2016] Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., and Erhan, D. (2016). Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351.

[Cai et al., 2011] Cai, D., He, X., Han, J., and Huang, T. S. (2011). Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560.

[Cao et al., 2017] Cao, Z., Long, M., Wang, J., and Jordan, M. I. (2017). Partial transfer learning with selective adversarial networks. *arXiv preprint arXiv:1707.07901*.

[Carlucci et al., 2017] Carlucci, F. M., Porzi, L., Caputo, B., Ricci, E., and Bulò, S. R. (2017). Autodial: Automatic domain alignment layers. In *International Conference on Computer Vision*.

[Cook et al., 2013] Cook, D., Feuz, K. D., and Krishnan, N. C. (2013). Transfer learning for activity recognition: A survey. *Knowledge and information systems*, 36(3):537–556.

[Cortes et al., 2008] Cortes, C., Mohri, M., Riley, M., and Rostamizadeh, A. (2008). Sample selection bias correction theory. In *International Conference on Algorithmic Learning Theory*, pages 38–53, Budapest, Hungary. Springer.

[Dai et al., 2007] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In *ICML*, pages 193–200. ACM.

[Davis and Domingos, 2009] Davis, J. and Domingos, P. (2009). Deep transfer via second-order markov logic. In *Proceedings of the 26th annual international conference on machine learning*, pages 217–224. ACM.

[Deng et al., 2014] Deng, W., Zheng, Q., and Wang, Z. (2014). Cross-person activity recognition using reduced kernel extreme learning machine. *Neural Networks*, 53:1–7.

[Donahue et al., 2014] Donahue, J., Jia, Y., et al. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, pages 647–655.

[Dorri and Ghodsi, 2012] Dorri, F. and Ghodsi, A. (2012). Adapting component analysis. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 846–851. IEEE.

[Duan et al., 2012] Duan, L., Tsang, I. W., and Xu, D. (2012). Domain transfer multiple kernel learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):465–479.

[Fernando et al., 2013] Fernando, B., Habrard, A., Sebban, M., and Tuytelaars, T. (2013). Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, pages 2960–2967.

[Fodor, 2002] Fodor, I. K. (2002). A survey of dimension reduction techniques. *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory*, 9:1–18.

[Ganin et al., 2016] Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35.

[Gao et al., 2012] Gao, C., Sang, N., and Huang, R. (2012). Online transfer boosting for object tracking. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 906–909. IEEE.

[Ghifary et al., 2017] Ghifary, M., Balduzzi, D., Kleijn, W. B., and Zhang, M. (2017). Scatter component analysis: A unified framework for domain adaptation and domain generalization. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1414–1430.

[Ghifary et al., 2014] Ghifary, M., Kleijn, W. B., and Zhang, M. (2014). Domain adaptive neural networks for object recognition. In *PRICAI*, pages 898–904.

[Gong et al., 2012] Gong, B., Shi, Y., Sha, F., and Grauman, K. (2012). Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

[Gopalan et al., 2011] Gopalan, R., Li, R., and Chellappa, R. (2011). Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006. IEEE.

[Gretton et al., 2012] Gretton, A., Sejdinovic, D., Strathmann, H., Balakrishnan, S., Pontil, M., Fukumizu, K., and Sriperumbudur, B. K. (2012). Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems*, pages 1205–1213.

[Gu et al., 2011] Gu, Q., Li, Z., Han, J., et al. (2011). Joint feature selection and subspace learning. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1294.

[Hamm and Lee, 2008] Hamm, J. and Lee, D. D. (2008). Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, pages 376–383. ACM.

[Hou et al., 2015] Hou, C.-A., Yeh, Y.-R., and Wang, Y.-C. F. (2015). An unsupervised domain adaptation approach for cross-domain visual classification. In *Advanced Video and Signal Based Surveillance (AVSS), 2015 12th IEEE International Conference on*, pages 1–6. IEEE.

[Hsiao et al., 2016] Hsiao, P.-H., Chang, F.-J., and Lin, Y.-Y. (2016). Learning discriminatively reconstructed source data for object recognition with few examples. *IEEE Transactions on Image Processing*, 25(8):3518–3532.

[Hu and Yang, 2011] Hu, D. H. and Yang, Q. (2011). Transfer learning for activity recognition via sensor mapping. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 1962, Barcelona, Catalonia, Spain. IJCAI.

[Huang et al., 2007] Huang, J., Smola, A. J., Gretton, A., Borgwardt, K. M., Schölkopf, B., et al. (2007). Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19:601.

[Jaini et al., 2016] Jaini, P., Chen, Z., Carbajal, P., Law, E., Middleton, L., Regan, K., Schaekermann, M., Trimponias, G., Tung, J., and Poupart, P. (2016). Online bayesian transfer learning for sequential data modeling. In *ICLR 2017*.

[Kermany et al., 2018] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131.

[Khan and Heisterkamp, 2016] Khan, M. N. A. and Heisterkamp, D. R. (2016). Adapting instance weights for unsupervised domain adaptation using quadratic mutual information and subspace learning. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 1560–1565, Mexican City. IEEE.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Li et al., 2012] Li, H., Shi, Y., Liu, Y., Hauptmann, A. G., and Xiong, Z. (2012). Cross-domain video concept detection: A joint discriminative and generative active learning approach. *Expert Systems with Applications*, 39(15):12220–12228.

[Li et al., 2016] Li, J., Zhao, J., and Lu, K. (2016). Joint feature selection and structure preservation for domain adaptation. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 1697–1703. AAAI Press.

[Li et al., 2018] Li, Y., Wang, N., Shi, J., Hou, X., and Liu, J. (2018). Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117.

[Liu et al., 2011] Liu, J., Shah, M., Kuipers, B., and Savarese, S. (2011). Cross-view action recognition via view knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3209–3216, Colorado Springs, CO, USA. IEEE.

[Liu and Tuzel, 2016] Liu, M.-Y. and Tuzel, O. (2016). Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477.

[Liu et al., 2017] Liu, T., Yang, Q., and Tao, D. (2017). Understanding how feature structure transfers in transfer learning. In *IJCAI*.

[Long et al., 2015a] Long, M., Cao, Y., Wang, J., and Jordan, M. (2015a). Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105.

[Long et al., 2016] Long, M., Wang, J., Cao, Y., Sun, J., and Philip, S. Y. (2016). Deep learning of transferable representation for scalable domain adaptation. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2027–2040.

[Long et al., 2014a] Long, M., Wang, J., Ding, G., Pan, S. J., and Yu, P. S. (2014a). Adaptation regularization: A general framework for transfer learning. *IEEE TKDE*, 26(5):1076–1089.

[Long et al., 2014b] Long, M., Wang, J., Ding, G., Sun, J., and Yu, P. S. (2014b). Transfer joint matching for unsupervised domain adaptation. In *CVPR*, pages 1410–1417.

[Long et al., 2013] Long, M., Wang, J., et al. (2013). Transfer feature learning with joint distribution adaptation. In *ICCV*, pages 2200–2207.

[Long et al., 2017] Long, M., Wang, J., and Jordan, M. I. (2017). Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217.

[Long et al., 2015b] Long, M., Wang, J., Sun, J., and Philip, S. Y. (2015b). Domain invariant transfer kernel learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1519–1532.

[Luo et al., 2017] Luo, Z., Zou, Y., Hoffman, J., and Fei-Fei, L. F. (2017). Label efficient learning of transferable representations acrosss domains and tasks. In *Advances in Neural Information Processing Systems*, pages 164–176.

[Mihalkova et al., 2007] Mihalkova, L., Huynh, T., and Mooney, R. J. (2007). Mapping and revising markov logic networks for transfer learning. In *AAAI*, volume 7, pages 608–614.

[Mihalkova and Mooney, 2008] Mihalkova, L. and Mooney, R. J. (2008). Transfer learning by mapping with minimal target data. In *Proceedings of the AAAI-08 workshop on transfer learning for complex tasks*.

[Nater et al., 2011] Nater, F., Tommasi, T., Grabner, H., Van Gool, L., and Caputo, B. (2011). Transferring activities: Updating human behavior analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1737–1744, Barcelona, Spain. IEEE.

[Pan et al., 2008a] Pan, S. J., Kwok, J. T., and Yang, Q. (2008a). Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI conference on Artificial intelligence*, volume 8, pages 677–682.

[Pan et al., 2008b] Pan, S. J., Shen, D., Yang, Q., and Kwok, J. T. (2008b). Transferring localization models across space. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 1383–1388.

[Pan et al., 2011] Pan, S. J., Tsang, I. W., Kwok, J. T., and Yang, Q. (2011). Domain adaptation via transfer component analysis. *IEEE TNN*, 22(2):199–210.

[Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE TKDE*, 22(10):1345–1359.

[Patil and Phursule, 2013] Patil, D. M. and Phursule, R. (2013). Knowledge transfer using cost sensitive online learning classification. *International Journal of Science and Research*, pages 527–529.

[Razavian et al., 2014] Razavian, A. S., Azizpour, H., Sullivan, J., and Carlsson, S. (2014). Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE.

[Saito et al., 2017] Saito, K., Ushiku, Y., and Harada, T. (2017). Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning*.

[Sener et al., 2016] Sener, O., Song, H. O., Saxena, A., and Savarese, S. (2016). Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 2110–2118.

[Shen et al., 2018] Shen, J., Qu, Y., Zhang, W., and Yu, Y. (2018). Wasserstein distance guided representation learning for domain adaptation. In *AAAI*.

[Si et al., 2010] Si, S., Tao, D., and Geng, B. (2010). Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(7):929–942.

[Silver et al., 2017] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of go without human knowledge. *Nature*, 550(7676):354.

[Stewart and Ermon, 2017] Stewart, R. and Ermon, S. (2017). Label-free supervision of neural networks with physics and domain knowledge. In *AAAI*, pages 2576–2582.

[Sun et al., 2016] Sun, B., Feng, J., and Saenko, K. (2016). Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8.

[Sun and Saenko, 2015] Sun, B. and Saenko, K. (2015). Subspace distribution alignment for unsupervised domain adaptation. In *BMVC*, pages 24–1.

[Sun and Saenko, 2016] Sun, B. and Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision*, pages 443–450. Springer.

[Tahmoresnezhad and Hashemi, 2016] Tahmoresnezhad, J. and Hashemi, S. (2016). Visual domain adaptation via transfer feature learning. *Knowledge and Information Systems*, pages 1–21.

[Tan et al., 2015] Tan, B., Song, Y., Zhong, E., and Yang, Q. (2015). Transitive transfer learning. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1155–1164. ACM.

[Tan et al., 2017] Tan, B., Zhang, Y., Pan, S. J., and Yang, Q. (2017). Distant domain transfer learning. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[Taylor and Stone, 2009] Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685.

[Tzeng et al., 2015] Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, Santiago, Chile. IEEE.

[Tzeng et al., 2017] Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. (2017). Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971.

[Tzeng et al., 2014] Tzeng, E., Hoffman, J., Zhang, N., et al. (2014). Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

[Vapnik and Vapnik, 1998] Vapnik, V. N. and Vapnik, V. (1998). *Statistical learning theory*, volume 1. Wiley New York.

[Wang et al., 2019] Wang, J., Chen, Y., Feng, W., Yu, H., Huang, M., and Yang, Q. (2019). Transfer learning with dynamic distribution adaptation. *ACM Intelligent Systems and Technology (TIST)*.

[Wang et al., 2017] Wang, J., Chen, Y., Hao, S., et al. (2017). Balanced distribution adaptation for transfer learning. In *ICDM*, pages 1129–1134.

[Wang et al., 2018a] Wang, J., Chen, Y., Hu, L., Peng, X., and Yu, P. S. (2018a). Stratified transfer learning for cross-domain activity recognition. In *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*.

[Wang et al., 2018b] Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., and Yu, P. S. (2018b). Visual domain adaptation with manifold embedded distribution alignment. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 402–410. ACM.

[Wang et al., 2014] Wang, J., Zhao, P., Hoi, S. C., and Jin, R. (2014). Online feature selection and its applications. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):698–710.

[Wei et al., 2016a] Wei, P., Ke, Y., and Goh, C. K. (2016a). Deep nonlinear feature coding for unsupervised domain adaptation. In *IJCAI*, pages 2189–2195.

[Wei et al., 2017] Wei, Y., Zhang, Y., and Yang, Q. (2017). Learning to transfer. *arXiv preprint arXiv:1708.05629*.

[Wei et al., 2016b] Wei, Y., Zhu, Y., Leung, C. W.-k., Song, Y., and Yang, Q. (2016b). Instilling social to physical: Co-regularized heterogeneous transfer learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

[Weiss et al., 2016] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1):1–40.

[Wu et al., 2017] Wu, Q., Zhou, X., Yan, Y., Wu, H., and Min, H. (2017). Online transfer learning by leveraging multiple source domains. *Knowledge and Information Systems*, 52(3):687–707.

[xinhua, 2016] xinhua (2016). http://mp.weixin.qq.com/s?__biz=MjM5ODYzNzAyMQ==&mid=2651933920&idx=1\&sn=ae2866bd12000f1644eae1094497837e.

[Yan et al., 2017] Yan, Y., Wu, Q., Tan, M., Ng, M. K., Min, H., and Tsang, I. W. (2017). Online heterogeneous transfer by hedge ensemble of offline and online decisions. *IEEE transactions on neural networks and learning systems*.

[Yosinski et al., 2014] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

[Yu et al., 2019] Yu, C., Wang, J., Chen, Y., and Huang, M. (2019). Transfer learning with dynamic adversarial adaptation network. In *The IEEE International Conference on Data Mining (ICDM)*.

[Zadrozny, 2004] Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the twenty-first international conference on Machine learning*, page 114, Alberta, Canada. ACM.

[Zellinger et al., 2017] Zellinger, W., Grubinger, T., Lughofer, E., Natschläger, T., and Saminger-Platz, S. (2017). Central moment discrepancy (cmd) for domain-invariant representation learning. *arXiv preprint arXiv:1702.08811*.

[Zhang et al., 2017a] Zhang, J., Li, W., and Ogunbona, P. (2017a). Joint geometrical and statistical alignment for visual domain adaptation. In *CVPR*.

[Zhang et al., 2017b] Zhang, X., Zhuang, Y., Wang, W., and Pedrycz, W. (2017b). Online feature transformation learning for cross-domain object category recognition. *IEEE transactions on neural networks and learning systems*.

[Zhao and Hoi, 2010] Zhao, P. and Hoi, S. C. (2010). Otl: A framework of online transfer learning. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 1231–1238.

[Zhao et al., 2010] Zhao, Z., Chen, Y., Liu, J., and Liu, M. (2010). Cross-mobile elm based activity recognition. *International Journal of Engineering and Industries*, 1(1):30–38.

[Zhao et al., 2011] Zhao, Z., Chen, Y., Liu, J., Shen, Z., and Liu, M. (2011). Cross-people mobile-phone based activity recognition. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence (IJCAI)*, volume 11, pages 2545–2550. Citeseer.

[Zheng et al., 2008] Zheng, V. W., Pan, S. J., Yang, Q., and Pan, J. J. (2008). Transferring multi-device localization models using latent multi-task learning. In *AAAI*, volume 8, pages 1427–1432, Chicago, Illinois, USA. AAAI.

[Zhuang et al., 2015] Zhuang, F., Cheng, X., Luo, P., Pan, S. J., and He, Q. (2015). Supervised representation learning: Transfer learning with deep autoencoders. In *IJCAI*, pages 4119–4125.

[Zhuo et al., 2017] Zhuo, J., Wang, S., Zhang, W., and Huang, Q. (2017). Deep unsupervised convolutional domain adaptation. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 261–269. ACM.