

## 一、什么是迁移学习

迁移学习 (Transfer Learning) 目标是将从一个环境中学到的知识用来帮助新环境中的学习任务。

迁移学习就是把已训练好的模型参数迁移到新的模型帮助新模型的训练(感觉不一定是把参数迁移, 还有的可以是输出特征向量+新模型、整个模型+fine-tune)。因为大多数数据或者任务存在相关性, 所以通过迁移学习我们可以将已经学到的模型参数(也可理解为模型学到的知识)通过某种方式来分享给新模型从而加快并优化模型的学习效率不用像大多数网络那样从零学习。

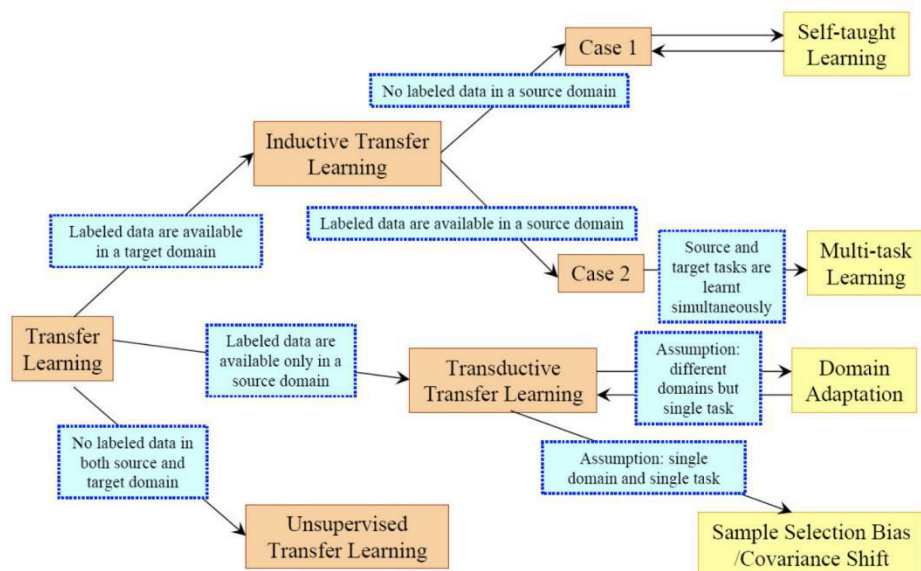
数学定义如下:

Given a source domain  $D_S = \{X_S, f_S(X)\}$  and learning task  $T_S$ , a target domain  $D_T = \{X_T, f_T(X)\}$  and learning task  $T_T$ , transfer learning aims to help improve the learning of the target predictive function  $f_T(\cdot)$  in  $D_T$  using the knowledge in  $D_S$  and  $T_S$ , where  $D_S \neq D_T$ , or  $T_S \neq T_T$ .

## 二、迁移学习大致分类

根据领域和任务的相似性, 迁移学习大致分类如下:

Transfer Learning Settings	Related Areas	Source Domain Labels	Target Domain Labels	Tasks
Inductive Transfer Learning	Multi-task Learning	Available	Available	Regression, Classification
	Self-taught Learning	Unavailable	Available	Regression, Classification
Transductive Transfer Learning	Domain Adaptation, Sample Selection Bias, Co-variate Shift	Available	Unavailable	Regression, Classification
Unsupervised Transfer Learning		Unavailable	Unavailable	Clustering, Dimensionality Reduction



### 1. Inductive Transfer Learning (归纳式迁移学习)

Inductive 的语义解释为归纳, 我们可以直接理解为学习适用于 Target Domain 的网络特征。

尤其注意，在 inductive 的迁移学习里，我们有 Target Domain 的 Ground Truth Label，这就意味着我们想要迁移的数据所在 Domain 将直接会学习网络和其特征提供指导作用（假设用深度学习方法，则通过 Back-propagation 方式）。

### 1.1 Self-Taught Learning（自我学习）

原论文思想：根据原作者引用的文献来看，他们先从网络上下载大量的图片，通过 Sparse Coding 的方式学习通用普世的图片特征，再直接从中找给定的任务比如图片分类所用的图片特征来做迁移学习。把这个思想放在现在的深度学习上，有点类同于通过 ImageNet 或其他大型数据集学习到的网络特征运用于一个图片分类或其他基于图片特征的任务。

缺点：由于神经网络训练局限于一个性质叫 Catastrophic Forgetting，意思是如果我们将 pre-trained 的模型作为初始值训练一个全新的任务，模型之前学习到的特征将灾难性遗忘，也意味着之前网络训练的任务结果将非常糟糕。

解决方法：

- Progressive Neural Networks - 通过 Lateral Connection 的方式将一学习好的模型参数通过另一层网络教给一个新的任务。由于在网络设计中，已训练完成的模型将会被 Freeze 因此在 back-propagation 优化参数的方式并不影响已学习完成的网络。

- Overcoming catastrophic forgetting in neural networks - 显然 DeepMind 自己也不会满足这样的网络设计，对此问题再次进行研究。这次他们不直接用于 Deep Learning 的 back-propagation 方式而专注于 Bayesian Learning 通过 Full Bayesian posterior distribution 来计算网络参数。他们的方法，Elastic Weight Consolidation (EWC)，通过 Laplace Approximation 的方式来计算 Bayesian 那也无法进行数值计算的 log-likelihood，最后取得相当不错的效果。

### 1.2 Multi-task Learning（多任务学习）

多任务学习 (multi-task learning) 与上个章节介绍的方向最大的不同是我们需要把 Source 和 Target 两个或者更多任务同时学好。在多任务学习中，任一任务的训练目标都是同等重要的。因此，这里 Source/Target Domain 的分类姑且没有太大意义（除非特定情况）。最简单直接的多任务学习网络就是直接应用一个普通的网络结构在最后输出层前分叉到不同的任务预测层。

## 2. Transductive Transfer Learning（直推式学习）

Transductive 的字面意思是转导，通常来说要比 inductive transfer learning 要难的多。由于直接缺乏 Target Domain Label，在这个情况，我们只能依赖于 Source Domain 信息来尽可能的训练出泛化能力强的特征。

### 2.1 Domain Adaptation / Generalisation（领域适应性--->特征空间相同/普适化）

Domain Adaptation 目标是学习 domain-invariant feature 使得学习到的特征不受限于 Source Domain 而导致 over-fitting。其主要会测试于 Office Dataset - Domain Adaptation - UC Berkeley，其中分为三个 Domain 其一是 Amazon 上扒的图片，另外两个是实体拍摄图片但一个高分辨率一个是低分辨率。

代表性作品：

- Unsupervised Domain Adaptation by Backpropagation
- Learning Transferable Features with Deep Adaptation Networks

不可否认，现在尝试解决 Domain Adaptation 的方法都不得不利用 Target Domain 的信息去缩小 co-variant shift. 完全只利用于 Source Domain 的信息仍然是个重要的课题。

## 3. Unsupervised Transfer Learning（无监督迁移学习）

在这个方向可适用于的任务非常有限，比如 Dimension Reduction。由于缺乏 Source Label 因此也无法采用绝大部分的 Deep Learning 方法，（但比如 Auto-encoder 是其中一个基于 Deep Learning 的降维方式，但很难归类于 Transfer Learning）。

根据 源 Domain 和目前 Domain 之间的关系，源 Task 和 目标 Task 之间的关系，以及任务方法更详细的整理为下表：

其中归纳迁移学习是应用最广泛的方法

		源Domain & 目标Domain	源Task & 目标Task	源Data & 目标Data	任务方法
传统机器学习		相同	相同	有标签   有标签	
迁移学习	归纳式迁移学习	相同/相关	相关	多任务学习 - 有标签   有标签 自我学习 - 无标签   有标签	分类回归
	直推式迁移学习	相关	相同	有标签   无标签	分类回归
	无监督迁移学习	相关	相关	无标签   无标签	聚类降维

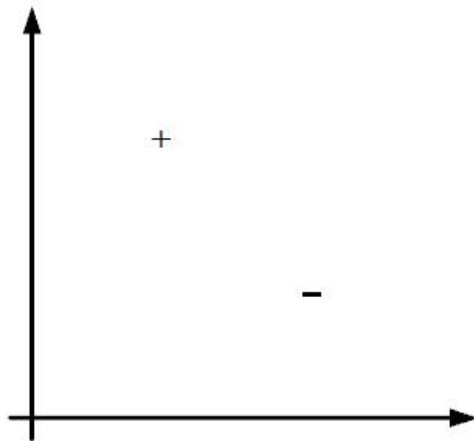
根据技术方法，我们将迁移学习的方法划分为：

	说明	归纳式	直推式	无监督
基于样本的迁移学习	通过调整 源Domain的标签（辅助）和 目标Domain标签的权重，协同训练得到目标模型。 典型方法：TrAdaBoost	√	√	
基于特征的迁移学习	找到“好”特征 来减少源Domain和目标Domain之间的不同，能够降低分类、回归误差。 典型方法：Self-taught learning，multi-task structure learning	√	√	√
基于参数的迁移学习	发现源Domain和目标Domain之间的共享参数或先验关系。 典型方法：Learning to learn，Regularized multi-task learning	√		
基于相关性的迁移学习	建立源Domain和目标Domain之间的相关知识映射。 典型方法：Mapping 方法	√		

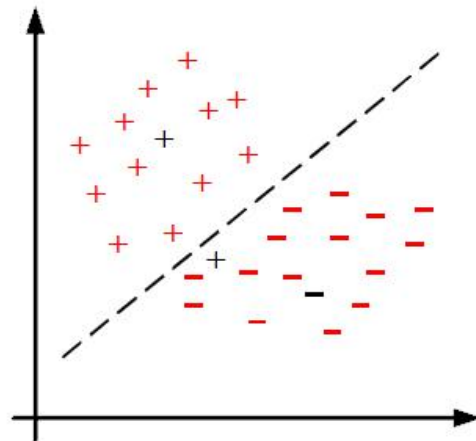
三、 经典算法

1. TrAdaBoost by 戴文渊

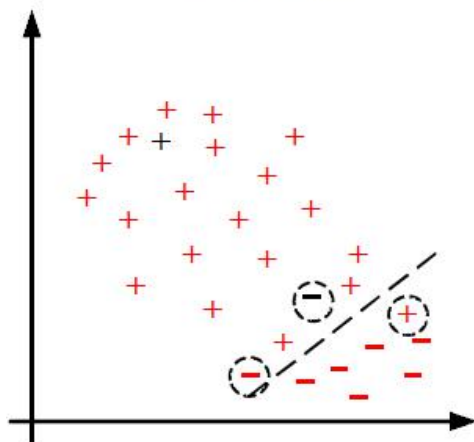
算法的基本思想是从源 Domain 数据中 筛选有效数据，过滤掉与目标 Domain 不 match 的数据，通过 Boosting 方法建立一种权重调整机制，增加有效数据权重，降低无效数据权重，下图是 TrAdaBoost 算法的示意图（截图来自于 庄福振 - 迁移学习研究进展）：



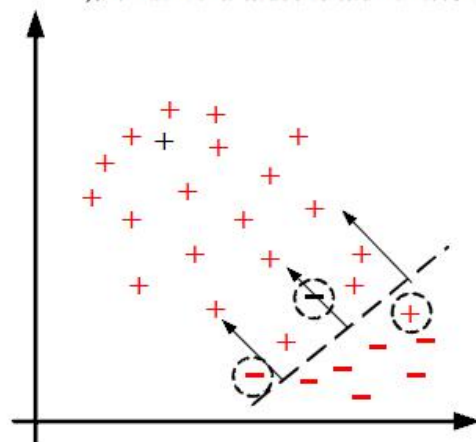
(a) 当有标注的训练样本很少的时候, 分类学习是非常困难的



(b) 如果能有大量的辅助训练数据(红色的“+”和“-”),则可能可以根据辅助数据估计出分类面



(c) 有时辅助数据也可能会误导分类结果, 例如图中黑色的“-”即被分错



(d) TrAdaBoost算法通过增加误分类的源训练数据的权重,同时减小误分类的目标训练数据的权重,使得分类面朝正确的方向移动

TrAdaBoost 算法比较简单,用一句话概括就是从过期数据里面找出和目标数据最接近的样本数据。

TrAdaBoost 的算法步骤:

---

**算法 1** TrAdaBoost 算法描述

---

输入 两个训练数据集  $T_a$  和  $T_b$ （根据公式(3.1)，合并的训练数据集  $T = T_a \cup T_b$ ），一个未标注的测试数据集  $S$ ，一个基本分类算法 **Learner**，和迭代次数  $N$ 。

**初始化**

1. 初始权重向量  $\mathbf{w}^1 = (w_1^1, \dots, w_{n+m}^1)^a$ ，其中，

$$w_i^1 = \begin{cases} 1/n & \text{当 } i = 1, \dots, n \\ 1/m & \text{当 } i = n+1, \dots, n+m \end{cases}$$

2. 设置  $\beta = 1/(1 + \sqrt{2 \ln n/N})$ .

**For**  $t = 1, \dots, N$

1. 设置  $\mathbf{p}^t$  满足

$$\mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{i=1}^{n+m} w_i^t}.$$

2. 调用 **Learner**，根据合并后的训练数据  $T$  以及  $T$  上的权重分布  $\mathbf{p}^t$  和未标注数据  $S$ ，得到一个在  $S$  的分类器  $h_t: X \mapsto Y$ 。

3. 计算  $h_t$  在  $T_b$  上的错误率：

$$\epsilon_t = \sum_{i=n+1}^{n+m} \frac{w_i^t |h_t(x_i) - c(x_i)|}{\sum_{i=n+1}^{n+m} w_i^t}.$$

4. 设置  $\beta_t = \epsilon_t / (1 - \epsilon_t)^b$

5. 设置新的权重向量如下

$$w_i^{t+1} = \begin{cases} w_i^t \beta^{|h_t(x_i) - c(x_i)|}, & \text{当 } i = 1, \dots, n \\ w_i^t \beta^{-|h_t(x_i) - c(x_i)|}, & \text{当 } i = n+1, \dots, n+m \end{cases}$$

**输出 最终分类器**

$$h_f(x) = \begin{cases} 1, & \sum_{t=\lceil N/2 \rceil}^N \ln(1/\beta_t) h_t(x) \geq \frac{1}{2} \sum_{t=\lceil N/2 \rceil}^N \ln(1/\beta_t) \\ 0, & \text{其他} \end{cases}$$

PS: 权重的更新方式，对于辅助样本来讲，预测值和标签越接近，权重越大；而对于目标数据则是相反，预测值和标签差异越大，权重越大。这种策略很容易理解，我们想找到辅助样本中 和 目标数据分布最接近的样本，同时放大目标样本 Loss 的影响，那么理想的结果就是：

目标样本预测值与标签尽量匹配（不放过一个没匹配好的数据），辅助样本在前面的基础上筛选出最 match（权重大的）的部分。

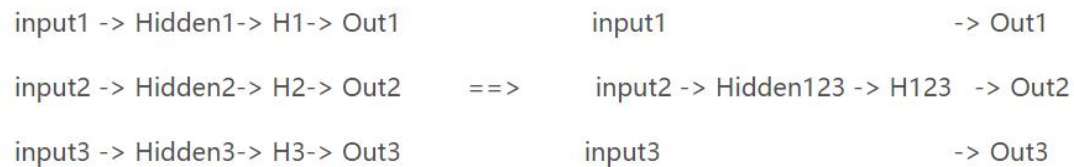
实验发现，当 同分布数据（目标数据）占比当低于 0.1 时，算法效果明显，当比例超过 0.1



时, TrBoost 退化为 SVM 的效果。

## 2. 多任务学习

多任务学习 (Multi-Task Learning, MTL) 是一种同时学习多个任务的机器学习方法, 该方法由来已久, 和深度学习没什么关系。



相当于把多个 Task 网络进行合并, 同时训练多个任务, 这种情况并不鲜见, 比如以下 2 个方向:

### 2.1 目标检测 — 复合多任务

目标检测是 分类问题+回归问题的组合, 这是一个典型的 Multi-Task, 比如:

Detection=Classification+Location

Mask RCNN = Classification+Location+Segmentation

### 2.2 特征提取

多任务特征提取, 多个输出, 这一类问题代表就是 数据结构化, 特征识别。

代表性作品:

• TCDCN (Facial Landmark Detection by Deep Multi-task Learning) 在这里 Multi-Task 被同时用作 人脸关键点定位、姿态估计和属性预测 (比如性别、年龄、人种、微笑? 戴眼镜?)

### 2.3 多任务学习适用于这样的情况:

- 1) 多个任务之间存在关联, 比如行人和车辆检测, 对于深度网络也可以理解为有部分共同的网络结构;
- 2) 每个独立任务的训练数据比较少, 单独训练无法有效收敛;
- 3) 多个任务之间存在相关性信息, 单独训练时无法有效挖掘;

## 四、面临问题

### 1. 哪种情况适合做迁移学习?

分类和回归问题是比较适合做迁移学习的场景, 有标签的源数据是最好的辅助。

### 2. 该选择哪种方法?

简单而行之有效的办法是首选, 领域在快速发展, 也不必拘泥算法本身, 改善结果才是硬道理。

### 3. 如何避免负迁移?

迁移学习的目标是改善目标域的 Task 效果, 这里面 **负迁移 (Negative Transfer)** 是很多研究者面临的一个问题, 如何得到行之有效的改进, 避免负迁移是需要大家去评估和权衡的。

参考资料:

- [1] [http://202.38.196.91/cache/9/03/www.cse.ust.hk/a57cd36fe0f93d6397c1acf60c5f5feb/tkde\\_transfer\\_learning.pdf](http://202.38.196.91/cache/9/03/www.cse.ust.hk/a57cd36fe0f93d6397c1acf60c5f5feb/tkde_transfer_learning.pdf)
- [2] <https://www.zhihu.com/question/41979241>
- [3] <http://home.cse.ust.hk/~qyang/Docs/2007/tradaboost.pdf>
- [4] <https://arxiv.org/pdf/1606.04671.pdf>
- [5] <http://www.public.asu.edu/~jye02/Software/MALSAR/MTL-SDM12.pdf>
- [6] <https://blog.csdn.net/linolzhang/article/details/73358219>

