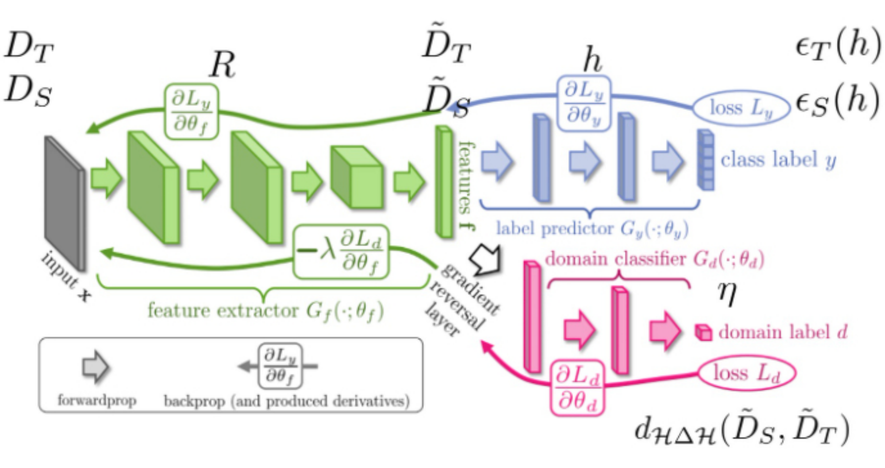


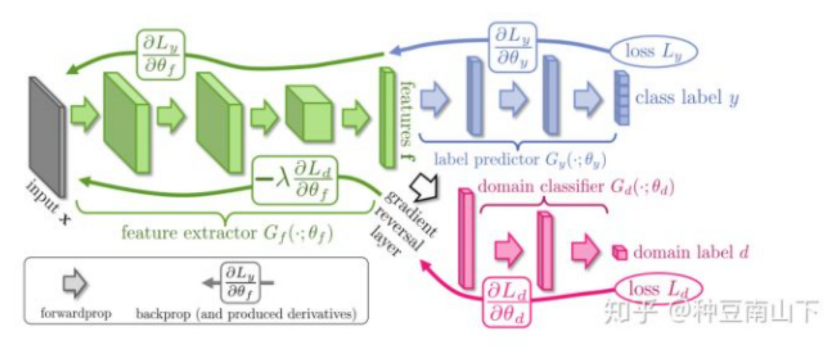
$$\epsilon_T(h) \le \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}^*}(\tilde{D}_S, \tilde{D}_T) + \lambda$$

《迁移学习》：领域自适应(Domain Adaptation)的理论分析

小蚂蚁曹凯
中国科学院大学 数学与系统科学研究院博士在读

领域自适应即***Domain Adaptation***是迁移学习中很重要的一部分内容，目的是把分布不同的源域和目标域的数据，映射到一个特征空间中，使其在该空间中的距离尽可能近。于是在特征空间中对source domain训练的目标函数，就可以迁移到target domain上，提高target domain上的准确率。我最近看了一些理论方面的文章，大致整理了一下，交流分享。

想必大家对GAN都不陌生，GAN是基于对抗的生成网络，主要目标是生成与训练集分布一致的数据。而在迁移学习领域，对抗也是一种常用的方式，如***Ganin[1]***的论文，使用的网络结构如下图，由三部分组成：**特征映射网络** $G_f(x;\theta_f)$ ，**标签分类网络** $G_y(z;\theta_y)$ 和**域判别网络** $G_d(z;\theta_d)$ 。



其中，source domain的数据是有标签的，target domain的数据是无标签的。 G_f 将source和target domain的数据都映射到一个特征空间 $\mathcal{Z}$ 上， G_y 预测标签y， G_d 预测数据来自于target还是source domain。所以流入 G_y 的是带标签的source数据，流入 G_d 的是不带标签的source和target的数据。

G_f：将数据映射到feature space，使 G_y 能分辨出source domain数据的label， G_d 分辨不出数据来自source domain还是target domain。

G_y ：对feature space的source domain数据进行分类，尽可能分出正确的label。

G_d：对feature space的数据进行领域分类，尽量分辨出数据来自于哪一个domain。

最终，希望 G_f 与 G_d 博弈的结果是source和target domain的数据在feature space上分布已经很一致， G_d 无法区分。于是，可以愉快的用 G_y 来分类target domain的数据啦。

理论分析

首先Domain Adaptation基本思想是既然源域和目标域数据分布不一样，那么就把数据都映射到一个特征空间中，在特征空间中找一个度量准则，使得源域和目标域数据的特征分布尽量接近，于是基于源域数据特征训练的判别器，就可以用到目标域数据上。

- **问题建立**

假设 $\mathcal{X}$ 是一个实例集(instance set)。

$\mathcal{Z}$ 是一个特征空间(feature space)。

$\mathcal{D}_S$ 是定义在 $\mathcal{X}$ 上的源域数据分布， $\tilde{\mathcal{D}}_S$ 是定义在 $\mathcal{Z}$ 上的源域特征分布。

$\mathcal{D}_T$ 、 $\tilde{\mathcal{D}}_T$ 一样定义目标域数据分布和特征分布。

$\mathcal{R}: \mathcal{X} \rightarrow \mathcal{Z}$ 是表示函数(representation function)将实例 $\mathcal{X}$ 映射到 $\mathcal{Z}$ 上，即上图的 G_f 。

$f: \mathcal{X} \rightarrow \{0,1\}$ 是真实的标签函数，是二值函数。我们并不知道 f 是什么，希望通过训练得到它。

$h: \mathcal{Z} \rightarrow \{0,1\}$ 是我们自己设计的预测函数,，给定一个特征 z ，得到一个其对应的标签，即上图的 G_y 。

$\mathcal{H}$：二值函数的集合， $h \in \mathcal{H}$ 。

接下来需要定义特征到标签的真实映射函数:

$$\tilde{f}(z) \overset{def}{=} E_{x\sim\tilde{\mathcal{D}}_S}[f(x)|R(x)=z]$$

注：这边 $\tilde{f}$ 是随机的是因为，即使 f 是确定的映射，给定特征 z 的情况下， z 也有可能以不同的概率来自于不同的 x 。

那么我们自己设计的预测函数 h 在源域上的错误率：

$$\epsilon_S(h) = E_{z\sim\tilde{\mathcal{D}}_S}|\tilde{f}(z) - h(z)|$$

- **度量准则**

接着就需要设计一个度量准则，度量通过 $\mathcal{R}$ 映射到特征空间的特征分布 $\tilde{\mathcal{D}}_S$ 、 $\tilde{\mathcal{D}}_T$ 之间的距离。这个距离必须满足的条件是：能通过有限个样本数据计算。

这边找到距离叫 $\mathcal{A}$ 距离，如下：

$$d_{\mathcal{A}}(\tilde{D}_S, \tilde{D}_T) = 2 \sup_{A\in\mathcal{A}} |Pr_{\tilde{D}_S}[A] - Pr_{\tilde{D}_T}[A]|$$

其中花体 $\mathcal{A}$ 是波莱尔集， A 是其一个子集。意思就是取遍所有 $\mathcal{A}$ 的子集，找出在 $\tilde{\mathcal{D}}_S$ 、 $\tilde{\mathcal{D}}_T$ 上的概率差的最大值。

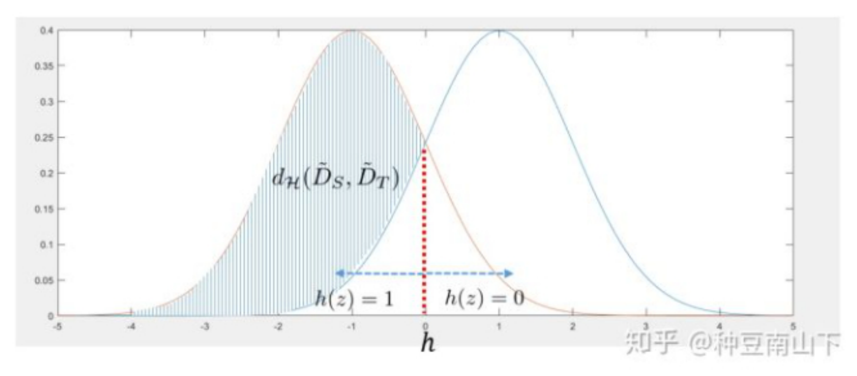
给 A 一个具体的取值，

$$A \rightarrow I(h) = \{z \in \mathcal{Z} : h(z) = 1, h \in \mathcal{H}\}$$

则此时的 $\mathcal{A}$ 距离可记作 $\mathcal{H}$ 距离：

$$d_{\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) = 2 \sup_{h\in\mathcal{H}} |Pr_{\tilde{D}_S}[I(h)] - Pr_{\tilde{D}_T}[I(h)]|$$

给一个简单的例子，如下：



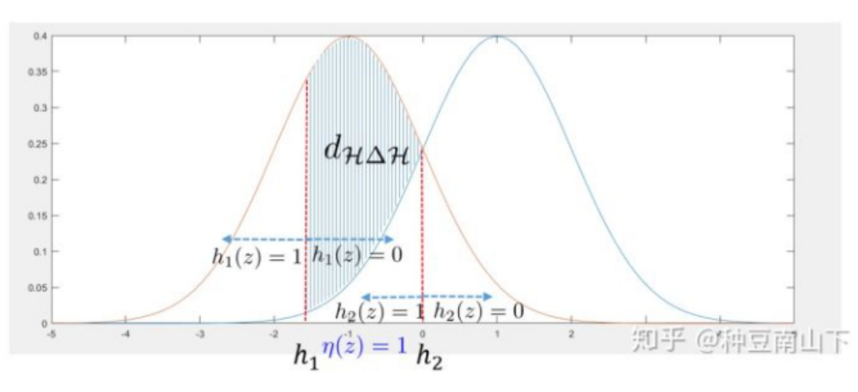
两个高斯分布分别代表源域和目标域的特征分布。由于要取上确界，所以找到的集合 $I(h)$ 为 $(-\infty, 0)$ 。

在 $\mathcal{H}$ 距离的基础上，再定义 $\mathcal{H}\Delta\mathcal{H}$ 距离：

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) &= 2 \sup_{h_1,h_2\in\mathcal{H}} |Pr_{\tilde{D}_S}[\{z : h_1(z) \neq h_2(z)\}] \\ &\quad - Pr_{\tilde{D}_T}[\{z : h_1(z) \neq h_2(z)\}]| \\ &= 2 \sup_{\eta\in\mathcal{H}\Delta\mathcal{H}} |Pr_{\tilde{D}_S}[z : \eta(z) = 1] - Pr_{\tilde{D}_T}[z : \eta(z) = 1]| \end{aligned}$$

$z^* = \{z : h_1(z) \oplus h_2(z), h_1, h_2 \in \mathcal{H}\}$ ， $\mathcal{H}\Delta\mathcal{H} = \{\eta : \eta(z^*) = 1\}$ ，$\oplus$：XOR operator

简单例子如下：



于是， $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T)$ 可以用下面的界限定：

$$\begin{aligned} &d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) \\ &= 2 \sup_{\eta\in\mathcal{H}\Delta\mathcal{H}} |Pr_{\tilde{D}_S}[z : \eta(z) = 1] - Pr_{\tilde{D}_T}[z : \eta(z) = 1]| \\ &\le 2 \sup_{\eta\in\mathcal{H}_d} |Pr_{\tilde{D}_S}[z : \eta(z) = 1] - Pr_{\tilde{D}_T}[z : \eta(z) = 1]| \\ &= 2 \sup_{\eta\in\mathcal{H}_d} |Pr_{\tilde{D}_S}[z : \eta(z) = 1] + Pr_{\tilde{D}_T}[z : \eta(z) = 0] - 1| \end{aligned}$$

其中，函数集合 $\mathcal{H}_d$ 只要取的比 集合 $\mathcal{H}\Delta\mathcal{H}$ 复杂即可，这个很容易达到，我们只需把神经网络设计的复杂一些就行。

从最后的式子来看，如果我们能一个在复杂度足够（能够实现比异或更复杂的操作）的函数类 $\mathcal{H}_d$ 中，找到一个函数 η ，使得 η 将来自于 $\tilde{\mathcal{D}}_S$ 的特征 z 都判为1，将来自于 $\tilde{\mathcal{D}}_T$ 的特征 z 都判为0的概率最大，那么我们就能得到 $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T)$ 的上界。

其实这边的 η 就是最开始图中红色的那部分网络 G_d 啦！

- **误差界**

好了，有了度量准则，那么下面就要介绍最重要的一个定理了。

Theorem: Let R be a fixed representation function from $\mathcal{X}$ to $\mathcal{Z}$, $\mathcal{H}$ is a binary function class, for every $h \in \mathcal{H}$:

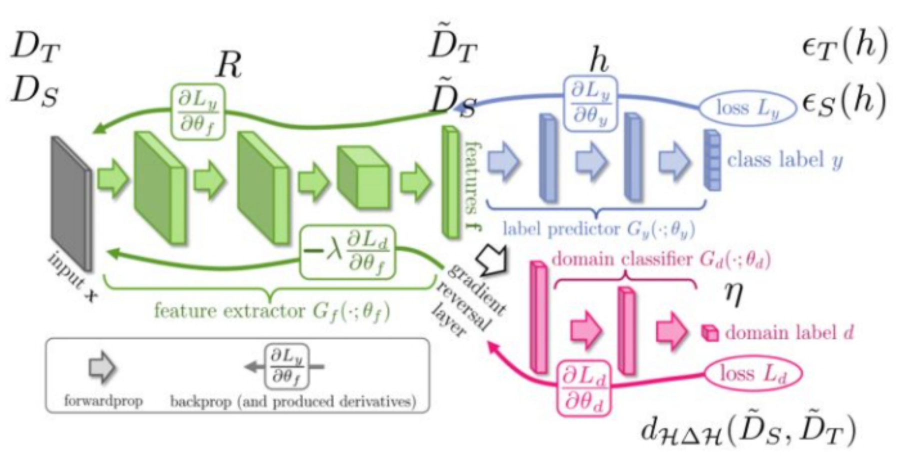
$$\epsilon_T(h) \le \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) + \lambda$$

where,

$$\begin{aligned} \lambda &= \epsilon_S(h^*) + \epsilon_T(h^*), \\ h^* &= arg \min_{h\in\mathcal{H}} \epsilon_S(h) + \epsilon_T(h) \end{aligned}$$

这个定理说的是，我们训练得到的分类函数 h 在目标域数据上的错误率，被三个项所限定。第一项是 h 在源域上的错误率。第二项是通过 $\mathcal{R}$ 将源域、目标域数据都映射到特征空间后，两者特征分布的距离， 即 $\mathcal{H}\Delta\mathcal{H}$ 距离。第三项是一个常数项可以不管。

如果把这些字母都加到开始的图上：



$$\epsilon_T(h) \le \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}^*}(\tilde{D}_S, \tilde{D}_T) + \lambda$$

可以看出，要降低 $\epsilon_T(h)$ ，表示函数 R (即 G_f)承担两项任务，需要降低 h 在源域上的错误率，还需要减小 $\mathcal{H}\Delta\mathcal{H}$ 距离。而 h (即 G_y)承担一项任务目标，就是降低在源域上的错误率。

对于 η (即 G_d)，要做的就是尽量能取到 $d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T)$ 中的上确界，让自己尽量能代表这个距离。其实，我个人想法，这边严格来说并不存在对抗， G_d 并不是一个坏蛋想要增大我们的错误率，它只是在默默的做自己本职的工作，想取到上确界，让自己能代表这个 $\mathcal{H}\Delta\mathcal{H}$ 距离。而 G_f 也不是去妨碍 G_d 去取上确界，而且想减小上确界本身。

到此，三个网络为什么这么设计应该就很清楚了叭！(至少我觉得讲清楚了233333，当然，最重要的定理的证明我省略了，有兴趣看下面参考的论文。)

- **实际计算**

如果引入VC维那一套关于泛化误差的理论，可以得到如下结论：

$\epsilon_S(h) \le \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})}$ 以 $1-\delta$ 概率成立

$\hat{\epsilon}_S(h) = \frac{1}{m}\sum_{i=1}^{m} I[h(x_i^s) \neq y_i^s]$ 是经验损失，可以通过有限数据计算。

m 是源域数据个数， d 是VC维的维数， e 是自然底数。

$d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) \le \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) + 4\sqrt{\frac{d\log(2m) + \log(\frac{2}{\delta})}{m}}$ 以 $1-\delta$ 概率成立。

$\hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) = 2(1 - \min_{h\in\mathcal{H}\Delta\mathcal{H}}[\frac{1}{m}\sum_{x:h(x)=0} I[x \in \tilde{D}_S] + \frac{1}{m'}\sum_{x:h(x)=1} I[x \in \tilde{D}_T])$

可以通过给定数据计算。m、m' 表示源域和目标域数据个数。

最终，我们的误差界由下式界定：

$$\epsilon_T(h) \le \hat{\epsilon}_S(h) + \sqrt{\frac{4}{m}(d\log\frac{2em}{d} + \log\frac{4}{\delta})} + \hat{d}_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) + 4\sqrt{\frac{d\log(2m) + \log(\frac{2}{\delta})}{m}} + \lambda$$

- **定理证明**

Theorem: Let R be a fixed representation function from $\mathcal{X}$ to $\mathcal{Z}$, $\mathcal{H}$ is a binary function class, for every $h \in \mathcal{H}$:

$$\epsilon_T(h) \le \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) + \lambda$$

where,

$$\begin{aligned} h^* &= arg \min_{h\in\mathcal{H}} \epsilon_S(h) + \epsilon_T(h) \\ \lambda_S &= \epsilon_S(h^*), \lambda_T = \epsilon_T(h^*), \\ \lambda &= \lambda_S + \lambda_T \end{aligned}$$

proof:

令 $Z_h = \{z \in \mathcal{Z} : h(z) = 1\}$ 表示特征空间 $\mathcal{Z}$ 中被 h 判为类别1的那些特征的集合。

则有：

$$\epsilon_T(h) \le \lambda_T + Pr_{\mathcal{D}_T}[Z_h \Delta Z_{h^*}] \quad (1)$$

这里的 Δ 是亦或，也就是 h、h^* 意见不一致的特征组成的集合，即：

$$Pr_{\mathcal{D}_T}[Z_h \Delta Z_{h^*}] = Pr_{\mathcal{D}_T}[\{z \in \mathcal{Z} : h(z) = 1\} \oplus \{z \in \mathcal{Z} : h^*(z) = 1\}], \quad \oplus : XOR$$

所以为什么不等式(1)成立？因为第一项 $\lambda_T = \epsilon_T(h^*)$ 是 h^* 的错误率，包含 h^*、h 意见一致时的判断错误的情况，第二项是意见不一致时的概率，包含 h^*、h 意见不一致时 h 判断错误的概率。所以 h 所有判断错误的概率，都包含在后面两项中！

继续往下推：

$$\lambda_T + Pr_{\mathcal{D}_T}[Z_h \Delta Z_{h^*}] \le \lambda_T + Pr_{\mathcal{D}_S}[Z_h \Delta Z_{h^*}] + |Pr_{\mathcal{D}_S}[Z_h \Delta Z_{h^*}] - Pr_{\mathcal{D}_T}[Z_h \Delta Z_{h^*}]| \quad (2)$$

这一步就没什么好说的了，就是一个数的绝对值大于等于其本身。

不等式(2)的第二项 $Pr_{\mathcal{D}_S}[Z_h \Delta Z_{h^*}]$ 是在源域上， h、h^* 意见不一致的概率。一旦意见不一致，那么必然有一方是错的，所以这项必然小于 h 和 h^* 的错误率之和：

$$Pr_{\mathcal{D}_S}[Z_h \Delta Z_{h^*}] \le \lambda_S + \epsilon_S(h) \quad (3)$$

根据上文

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) &= 2 \sup_{h_1,h_2\in\mathcal{H}} |Pr_{\tilde{D}_S}[\{z : h_1(z) \neq h_2(z)\}] - Pr_{\tilde{D}_T}[\{z : h_1(z) \neq h_2(z)\}]| \\ &= 2 \sup_{h_1,h_2\in\mathcal{H}} |Pr_{\tilde{D}_S}[Z_{h_1} \Delta Z_{h_2}] - Pr_{\tilde{D}_T}[Z_{h_1} \Delta Z_{h_2}]| \end{aligned}$$

所以不等式(2)的第三项

$$|Pr_{\mathcal{D}_S}[Z_h \Delta Z_{h^*}] - Pr_{\mathcal{D}_T}[Z_h \Delta Z_{h^*}]| \le \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) \quad (4)$$

所以综合不等式(1-4)，有：

$$\begin{aligned} \epsilon_T(h) &\le \lambda_T + \lambda_S + \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) \\ &\le \lambda_T + \epsilon_S(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\tilde{D}_S, \tilde{D}_T) \end{aligned}$$

定理得证。

参考论文：

1. Ben-David, Shai, Blitzer, John, Crammer, Koby, and Pereira, Fernando. Analysis of representations for domain adaptation. In NIPS, pp. 137–144, 2006.
2. Ben-David, Shai, Blitzer, John, Crammer, Koby, Kulesza, Alex, Pereira, Fernando, and Vaughan, Jennifer Wort-man. A theory of learning from different domains. JMLR, 79, 2010.
3. Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Fran¸cois Laviolette, Mario Marchand, Victor Lempitsky. Domain-Adversarial Training of Neural Networks. Journal of Machine Learning Research 17 (2016) 1-35
4. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. arXiv preprint arXiv:1409.7495 (2014)