

可能是由于文章提供的 filter_k 这个超参数只进行了{1, 5, 20, 50, 100, N/A}这几个数据的尝试，或许在不同的 filter_k 下能有更好的结果。

表二: Entorpy

Algorithm	VLCS	PACS	OfficeHome
<i>ERM+Resnet50+T3A</i>			
ERM	0.0897	0.0782	0.3210
Resnet50+T3A	0.0382	0.0132	0.0293
<i>ERM+Resnet18+T3A</i>			
ERM	0.2752	0.1094	0.3533
Resnet18+T3A	0.1219	0.0636	0.0584

在熵的方面，T3A 引入了熵过滤机制，很明显在使用 T3A 进行测试后，在所有数据集上熵都有明显降低。

三、T3A 算法理解

1、T3A 的作用域

按照文中所说，我们讲一个深度神经网络（DNN）分成两个部分，一个是线性分类器 $q\omega$ ，一个是特征提取器 $f\theta$ ，而 T3A 就作用在线性分类器上。

2、为什么考虑在线性分类器上作调整

正常来说，模型在进行预测时，就是将通过线性分类器中的参数 ω 作为模板来衡量是属于哪个类别的，但是由于模板是在源域中训练的，所以当预测在目标域中进行的时候， ω 这个模板就可能不再那么有效了。

3、算法核心：支持集更新

于是 T3A 把目光放在了调整这个模板 ω 上，通过更新支持集的方法来适应新的领域的的数据。

$$S_t^k = \begin{cases} S_{t-1}^k \cup \left\{ \frac{f_{\theta}(x)}{\|f_{\theta}(x)\|} \right\} & \text{if } \hat{y} = y^k \\ S_{t-1}^k & \text{else,} \end{cases}$$

假设 $t=0$ 时，支持集 S 应该就是 ω 的归一化表示，然后随着 t 不断往后（也就是不断在线接收新样本），支持集 S 会根据特征提取器 $f\theta$ 得到的特征归一化后去对每个类别的支持集 S 作更新（根据预测标签 (\hat{y}) 属于哪个类别 (k) ，来更新对应的支持集 (S_k) ）

我的理解是这样就可以在测试的过程中一定程度上让线性分类器学到新领域的的数据分布

4、支持集更新中的过滤器

而这中间支持集也不是每个特征都要

and may deteriorate performance. To avoid this issue, we use the prediction entropy $H_{\omega}(\hat{Y}|z) = -\sum_k q_{\omega}(\hat{Y} = y^k|z) \log q_{\omega}(\hat{Y} = y^k|z)$ to filter unreliable pseudo-labeled data. Specifically, before making a prediction using eq. 4, only a part of the support set is restored as follows:

$$\mathbb{S}_t^k = \{z \mid z \in \mathbb{S}_t, H_{\omega}(\hat{Y}|z) \leq \alpha^k\}, \quad (6)$$

where α^k is the M -th largest entropy of the support set \mathbb{S}_t^k (M is a hyperparameter).

文章通过计算预测熵和选取熵的范围来过滤他们认为不可靠的特征（文章解释：熵越低说明模型越有信心），其中 α_k 是用于过滤的超参数，支持集将过滤比支持集中第 α_k 大的熵还大的熵（应该就对应着跑出的数据结果中 filter_k 这一栏）

但文章说他们没有显示地减少预测熵，我对这个地方有所疑问。

5、算法总结和我的理解

我的粗浅理解下，T3A 的算法就是对线性分类器作进一步处理，通过向支持集中添加测试过程中见过的特征来增加分类器对目标领域的数据分布把握，再通过添加一个预测熵的阈值来过滤感觉不好的特征，以此达到分类更精准的目标。

文章后续实验了冻结特征提取器的参数，只对线性分类器进行微调的可行性，发现只微调线性分类器，对领域适应内容有相当不错的提升，而即便对整体模型都进行调整，所得的提升也没有超过微调线性分类器太多，并且微调线性分类器的代价并不大，说明微调线性分类器方向应该是个可行的优化模型方向