Domain Generalization using Causal Matching

problem

domain generalization 问题:the task of learning a machine learning model that can generalize to unseen data distributions, after training on more than one data distributions.

Related Work

1. **Learning common representation.** 尝试学习能够独立与域存在的表征$\Phi(X)$,或者某个class内部的不随环境而改变的表征$\Phi(X)$
2. **Causality and domain generalization.** 现有的将因果理论与泛化性能融合的方法,有假设$Y \rightarrow X$的,也有假设$X \rightarrow Y$的.本文的工作融合了上述两个方向

其他方法包括元学习,数据增强,参数分解,以及IRM之类的方法,作者也设置了对照实现

contribution

Insufficiency of class-conditional invariance

现有的主流方法核心思想一般是找到表征$\Phi(X)$,使其满足

$$\Phi(X) \perp\!\!\!\perp D|Y \tag{1}$$

.

作者在这里给出了这种情形的反例,即说明满足(1),依旧无法做到泛化.

考虑一个二维的问题,两个表征的生成过程如下:

$$x_1 = x_c + \alpha_d \tag{2}$$
$$x_2 = \alpha_d \tag{3}$$

其中x_c, α_d是未观测到的量,并且后者会随环境而发生改变.并且 $y = f(x_c) = I(x_c \geq 0)$ 关系如下:

$$Domain\, 1: X_c|Y = 1 \sim \mathcal{U}(1,3); X_c|Y = 0 \sim \mathcal{U}(-2,0); \alpha = 1 \tag{4}$$
$$Domain\, 2: X_c|Y = 1 \sim \mathcal{U}(0,2); X_c|Y = 0 \sim \mathcal{U}(-3,-1); \alpha = 2 \tag{5}$$

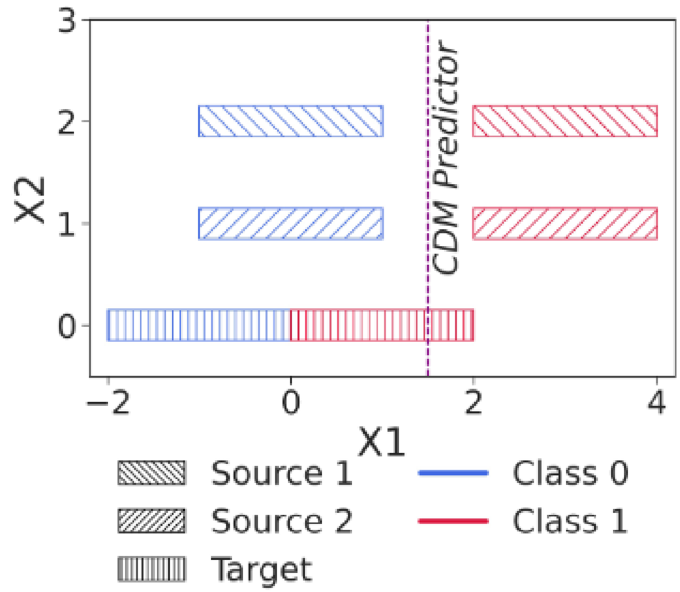
于是就可以发现$x_1|Y$能在各个Domain中的分布都是稳定的.于是给出

$$\Phi(x) = x_1 \tag{6}$$

于是依赖(6)进行训练会发现当$x_1 > 1.5$(0到1之间任意一个数)时$Y = 1$,否则$Y = 0$,但当测试环境为

$$\alpha_d = 0; X_c|Y = 1 \sim \mathcal{U}(0,2); X_c|Y = 0 \sim U(-2,0), \tag{7}$$

那么这个分类器的就无法进行泛化了,正确率只有65%,过程如下图所示:



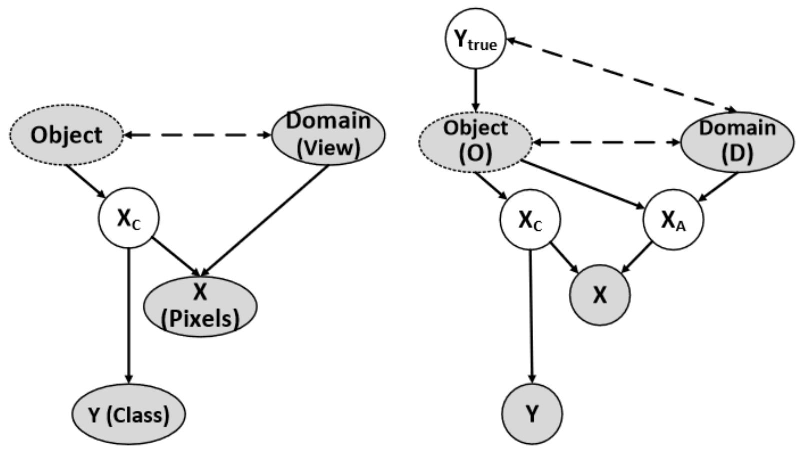
(a) Simple Example

这个例子中最合适的表征是$x_1 - x_2$即x_c,但这个表征并不满足(1)式.于是作者给出了以下的推论,来说明何时(1)可以用于模型的泛化:

Proposition 1. *Under the domain generalization setup as above, if $P(X_c|Y)$ remains the same across domains where x_c is the stable feature, then the class-conditional domain-invariant objective for learning representations yields a generalizable classifier such that the learnt representation $\Phi(\mathbf{x})$ is independent of the domain given x_c. Specifically, the entropy $H(d|x_c) = H(d|\Phi, x_c)$.*

但在实际中$P(X_c|Y)$会变,上述推论就很难成立

A Causal View of Domain Generalization



(a) Image classification. (b) General SCM.

Figure 2. Structural causal models for the data-generating process. Observed variables are shaded; dashed arrows denote correlated nodes. *Object* may not be observed.

数据生成过程

考虑图像分类的任务,对于相同的对象(object,变量O),从不同的视角进行观测形成了不同的view或者说Domain.数据生成过程如(a)所示,高层的causal feature与Domain共同作用产生了像素信息X.同时causal feature也产生了label Y.

更加宽泛的causal graph是(b).其中有与object相关的变量X_C以及与Domain相关的X_A.SCM模型描述如下:

$$o := g_o(y_{true}, \epsilon_o, \epsilon_{od}) \tag{8}$$

$$\mathbf{x}_c = g_{xc}(o) \tag{9}$$

$$\mathbf{x}_a := g_{xa}(d, o, \epsilon_{xa}) \tag{10}$$

$$\mathbf{x} := g_x(\mathbf{x}_c, \mathbf{x}_a, \epsilon_x) \tag{11}$$

$$y := h(\mathbf{x}_c, \epsilon_y) \tag{12}$$

数据生成过程

从(b)中可以发现$Y \perp\!\!\!\perp$,于是目标就转变为学习$y = h(\mathbf{x_c})$,损失函数可以写成

$$\arg\min_f \mathbb{E}_{(d,\mathbf{x},y)} l(y, f(\mathbf{x})) = \arg\min_h \mathbb{E}[l(y, h(\mathbf{x}_c))]$$

但由于X_C是无法关测到的,于是我们需要学习映射$\Phi : \mathcal{X} \to \mathcal{C}$.于是整体的要学习的是$h(\Phi(X)) : \mathcal{X} \to \mathcal{Y}$

作者证明了

Proposition 2. *Given observed data distribution $P(Y, X, D, O)$ that may also include data obtained from interventions on domain D, multiple values of X_C yield exactly the same observational and interventional distributions and hence X_c is unidentifiable.*

即给定观测数据$P(X, Y, D, O)$,依旧无法识别出唯一确定的X_C,除非对X, Y的作用机制有一定的约束才有可能确定唯一的Φ

A “perfect-match” invariant

因为直接识别X_c不太可行,于是作者尝试找到一个变量来表征X_c.通过观察可知X_c具有

1. $X_C \perp\!\!\!\perp D|O$
2. $X_C \not\perp\!\!\!\perp O$

第一个特征说明相同的object的X_C不会随着Domain而发生变化,为此作者引入了如下的损失项:

$$\sum_{\Omega(j,k)=1;d\neq d'} \operatorname{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) = 0. \tag{13}$$

其中$\Omega : \mathcal{X} \times \mathcal{X} \rightarrow \{0, 1\}$等于1说明两个case中的object是一致的.

另外还需要让X_C能够反映足够的O的信息.于是损失函数设置为

$$f_{\text{perfectmatch}} = \arg\min_{h,\Phi} \sum_{d=1}^{m} L_d(h(\Phi(X)), Y) \tag{14}$$

$$\text{s.t.} \sum_{\Omega(j,k)=1;d\neq d'} \operatorname{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) = 0 \tag{15}$$

其中$L_d(h(\Phi(X), Y)) = \sum_{i=1}^{n_d} l(h(\Phi(\mathbf{x}_i^{(d)}), y_i^{(d)})$.

实际上满足(2)的$\Phi(X)$会有很多,而且一般都有比较不错的效果,但是作者想要寻找**能够stable的**Φ,并且不能与X_a之间存在关联.

作者给出了以下的定理:

Theorem 1. *For a finite number of domains m, as the number of examples in each domain $n_d \to \infty$,*
1. The set of representations that satisfy the condition $\sum_{\Omega(j,k)=1; d\neq d'} \text{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) = 0$ contains the optimal $\Phi(\mathbf{x}) = X_C$ that minimizes the domain generalization loss in (1).
2. Assuming that $P(X_a|O, D) < 1$ for every high-level feature X_a that is directly caused by domain, and for P-admissible loss functions (Miller et al., 1993) whose minimization is conditional expectation (e.g., ℓ_2 or cross-entropy), a loss-minimizing classifier for the following loss is the true function f^*, for some value of λ.

$$f_{\texttt{perfectmatch}} = \arg\min_{h,\Phi} \sum_{d=1}^{m} L_d(h(\Phi(X)), Y) + \lambda \sum_{\Omega(j,k)=1; d\neq d'} \text{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) \quad (3)$$

这个定理想说明,$\Phi(\mathbf{x}) = X_C$必然是满足(13)的,并且利用(14),(15)得到的最优解f一定是关于X_C的函数而不是一个常数项.**但在证明过程中存在假设:至少有一个object或者说Domain,$P(X_a|D=d, O=o) < 1$** 即对于某个object,在某个Domain中,生成的非causal feature是随机的而不是确定性的,**那么这个假设可能无法应用于Domain只有两个并且作用是确定性的情况,因为这时很有可能学到的Φ是依赖于X_a的**,作者认为通过对模型大小的正则可以缓解这个问题,或者我觉得使用多个Domain的数据可能也可以解决这个问题

考虑我们有两个Domain的数图片,一个Domain为原图,另一个Domain为原图旋转过一个固定的角度α,那么这时X_a就会是角度α,而学到的Φ可能就

是倒转α,此时就可能两个Domain输入之后结果是一样的,但这种模型是不存在泛化能力的,没有办法泛化至其他Domain之上.

MatchDG: Matching without objects

很多情况下object information是未知的,这时损失函数(3)就没办法使用了,因为无法直接从数据集中找到源自同一object但是Domain不同的反事实case,于是作者使用了两阶段的迭代对比学习方法来模拟object match,目标为学习matching:$\Omega : \mathcal{X} \times \mathcal{X} \rightarrow \{0,1\}$,即学习一种判别机制来判别两个样本是否属于同一个object.并且当这个判别机制判别出$\Omega(\mathbf{x}, \mathbf{x}') = 1$时,$x_c, x_c'$之间的差距要非常小.这里作者使用了以下的假设:

Assumption 1. *Let $(\mathbf{x}_i^{(d)}, y)$, $(\mathbf{x}_j^{(d')}, y)$ be any two points that belong to the same class, and let $(\mathbf{x}_k^{(d)} y')$ be any other point that has a different class label. Then the distance in causal features between $\mathbf{x}_i$ and $\mathbf{x}_j$ is smaller than that between $\mathbf{x}_i$ and $\mathbf{x}_k$ or $\mathbf{x}_j$ and $\mathbf{x}_k$:* $\mathrm{dist}(x_{c,i}^{(d)}, x_{c,j}^{(d')}) \leq \mathrm{dist}(x_{c,i}^{(d)}, x_{c,k}^{(d')})$ *and* $\mathrm{dist}(x_{c,j}^{(d)}, x_{c,i}^{(d')}) \leq \mathrm{dist}(x_{c,j}^{(d)}, x_{c,k}^{(d')})$.

相同类的样本的causal feature之间的距离要大于不同类的样本的causal feature的距离

Contrastive Loss

对比学习是一种无监督的学习范式,将相同标签但是源自不同域的特征输入到两个特征学习器之中,最终要求label相同的特征学习到的表征相似,但label不同的表征学习到的特征差距尽量大.

作者在这里使用了迭代的方法,来确定哪些样本之间是要进行配对的,或者说那些样本之间有相同的label并且来自不同的Domain.具体的匹配过程如

下:

具体的实现过程是这样的

Initialization（构造random match）：首先我们对每一个类选择一个基域（包含该类元素最多的类），对基类的所有数据点进行遍历。对每个数据点，我们随机的在剩下 $K-1$ 个域中给他匹配标签相同的元素，因此会构造出一个 (N', K) 大小的数据矩阵，这里 N' 即所有类的基域大小之和，K是总共的域的数目。

Phase 1: 采样一个batch的数据 (B, K)，对batch中的每个数据点最小化对比损失，和他具有相同object不同域的样本作为正样本，不同object样本作为负样本。

$$l(\mathbf{x}_j, \mathbf{x}_k) = -\log \frac{e^{\mathrm{sim}(j,k)/\tau}}{e^{\mathrm{sim}(j,k)/\tau} + \sum_{i=0, y_i \neq y_j}^{B} e^{\mathrm{sim}(j,i)/\tau}}$$

每 t 个epoch使用通过对比学习学到的representation更新一次我们的match。首先还是要选基域，但是在基域选定后，我们不再随机的在剩下域中挑选sample，我们为基域中的该类的每个样本在其他域中找representation距离最近的点作为正样本。

在Phase 1结束时，我们根据学习到的最终表示的 L_2 距离更新匹配的数据矩阵。我们称这些匹配为推论匹配。

接着使用第一阶段学习到的表征映射Φ来进行匹配,第二阶段的表征再从头开始学:

Phase 2：我们使用下列损失函数，但是match使用我们第一阶段学到的。网络从头开始训练（第一阶段学到的网络只是用来做匹配而已）。但是第一阶段学到的匹配可能不能包含所有的数据点，因此作者在每次训练除了从数据矩阵采样 (B, K) 的数据外，还通过随机匹配再产生 (B, K) 的数据。

$$f_{\mathtt{randommatch}} = \arg\min_{h,\Phi} \sum_{d=1}^{m} L_d(h(\Phi(X)), Y) + \lambda \sum_{\Omega_Y(j,k)=1; d \neq d'} \mathrm{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')}))$$

实验测试

模型测试中有以下的实验结果:

Dataset	Source	ERM	MASF	CSD	IRM	RandMatch	MatchDG	PerfMatch (Oracle)
Rotated MNIST	15, 30, 45, 60, 75	93.0 (0.11)	93.2 (0.2)	94.5 (0.35)	92.8 (0.53)	93.4 (0.26)	**95.1** (0.25)	96.0 (0.41)
	30, 45, 60	76.2 (1.27)	69.4 (1.32)	77.7 (1.88)	75.7 (1.11)	78.3 (0.55)	**83.6** (1.44)	89.7 (1.68)
	30, 45	59.7 (1.75)	60.8 (1.53)	62.0 (1.31)	59.5 (2.61)	63.8 (3.92)	**69.7** (1.30)	80.4 (1.79)
Rotated Fashion MNIST	15, 30, 45, 60, 75	77.9 (0.13)	72.4 (2.9)	78.7 (0.38)	77.8 (0.02)	77.0 (0.42)	**80.9** (0.26)	81.6 (0.46)
	30, 45, 60	36.1 (1.91)	29.7 (1.73)	36.3 (2.65)	37.8 (1.85)	38.4 (2.73)	**43.8** (1.33)	54.0 (2.79)
	30, 45	26.1 (1.10)	22.8 (1.26)	24.2 (1.69)	26.6 (1.06)	26.9 (0.34)	**33.0** (0.72)	41.8 (1.78)

可以看到作者**还验证了当训练使用的dataset减少时模型的测试效果**(这是可以借鉴的点)