Model Adaption : Unsupervised Domain Adaption without Source Data

abstract

- 本文考虑一个更加具有挑战性的setting, 即没有source data的情况下, 实现unsupervised model adapation (UDA)
- 由于source data在很多情况下是not available的, 因此提出collaborative class conditional GAN, 并结合两个regularization实现
- 在过程中, prediction model会使用generated target-style data被提升, 同时这样也可以对generator提供更好的guidance, 形成了collaborate
- 提出parameter regularization和cluster-based regularization. parameter regularization会使得模型更加和source model相似, cluster-based regularization会introduced more discriminative feature in target domain
- 在很多实验中证明了本文的有效性 (没提SOTA?)

Introduction

- 经典开头, 深度学习的性能依赖大量数据集blabla.., 并且都assume 训练数据和实际用的数据是I.I.D的
- 当test data和source domain的数据不同时, 性能会极大的下降, 这通常称为domain shift. transfer learning主要研究domain shift. 然而transfer learning的大多数方法都需要source data, 而在很多情况下, source data是不可得的
- 由于source data可能是涉及隐私, 因此开发不需要source domain的UDA方法是有很大的实用价值的
- 近期的domain adaption方法可以分为两类, 1) 通过minimizing在source domain和target domain之间的一个specific distribution distance来学习domain-invariant features. 2) 通过使用GAN直接将source data转化为target data. 然而这些方法都需要source data
- 本文提出一种方法不需要source data进行UDA. 主要contribute在于 1) 提出一个新setting, 即无source domain data的UDA. 2) 提出collaborate class conditional GAN (3CGAN)来解决这个问题. 3) 在实验中取得了SOTA效果, 表明了方法的有效性

Method

整个frame work由一个classifier C, generator G以及一个discriminator D 组成, 另外, objective function 除了标准的GAN Loss以外, 还增加了 parameter regularization 和cluster-based regularization

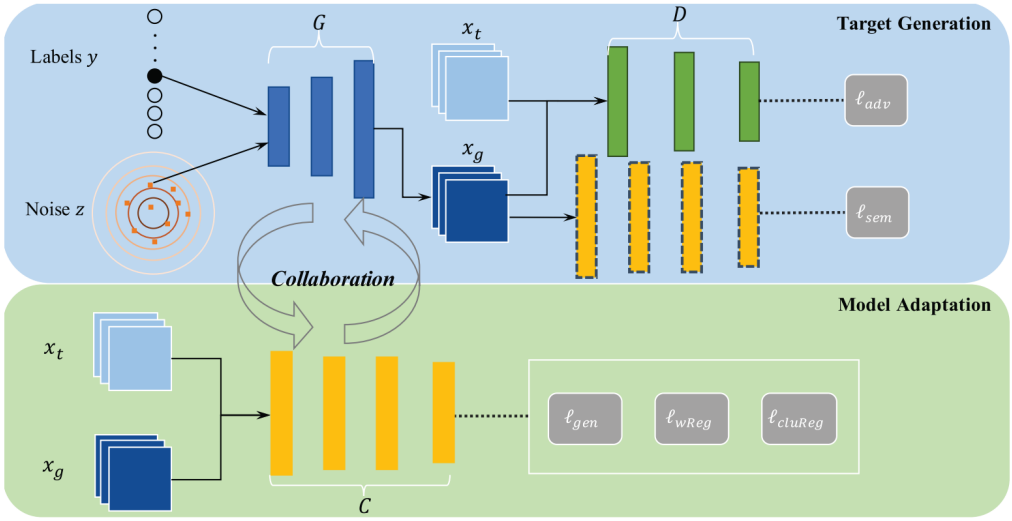


Figure 2. An overview of the proposed architecture. During target generation (top), we aim to learn a class conditional generator G for producing target-style training samples $x_g = G(y, z)$ via the discriminator D and the prediction model C (which is fixed as denoted by dashline). The generated images and proposed regularizations are used for model adaptation (bottom). These two procedures are repeated, with G and C collaborating with each other. (See text for details)

- generator从一个uniform的分布中sampling, 并且用pre-defined label进行condition, 部分loss为

$$\ell_{adv}(G) = \mathbb{E}_{y,z}[\log D(1 - G(y, z))].$$

- discriminator区分x_t和x_g, loss为

$$\max_{\theta_D} \mathbb{E}_{x_t \sim \mathcal{D}_t}[\log D(x_t)] + \mathbb{E}_{y,z}[\log(1 - D(G(y, z)))].$$

- semantic similarity regularization: 这个regularization能够使得generator生成的样本be semantic to source classifier

$$\ell_{sem}(G) = \mathbb{E}_{y,z}[-y \log p_{\theta_C}(G(y, z))],$$

因此对于generator的整体loss为, λ用于平衡loss项的参数

$$\min_{\theta_G} \ell_{adv} + \lambda_s \ell_{sem},$$

- 同时还加入了cluster_based regularization, 使得整个模型的决策边界更加靠近低数据密度区域. 因此classifier整体的objective function为

$$\min_{\theta_C} \lambda_g \ell_{gen} + \lambda_w \ell_{wReg} + \lambda_{clu} \ell_{cluReg}$$

- Weight Regularization

  虽然如上的3CGAN已经能对classifier提供较好的提升, 但training process不总是stable, 因为缺少了accurate supervised from labeled data. 由[46, 57]尝试学习source和target domain两个独立但相关的prediction model的启发, 本文提出了weight regularization 来避免新训练的模型C 偏离pre-trained模型太多, 可以被表示为

$$\ell_{wReg} = \|\theta_C - \theta_{C_s}\|^2$$

一方面, 该项损失避免更新的模型变化得太显著, 另一方便要求模型参数和原来的模型相似可以被视为保持source knowledge

- Clustering-based Regularization

  大多数UD方法都注重于adaptation process, 无标签的 real target data仅仅用于评估target dsitribution, 本文作者考虑unlabeled data能够被用于发现在target domain的discriminative information. cluster assumption指出预测模型的决策边界不应该是穿过高密度的数据区域. 因此本文minimize 在target domain上预测概率的conditional entropy

$$\mathbb{E}_{x_t \sim \mathcal{D}_t}[-p_{\theta_C}(x_t) \log p_{\theta_C}(x_t)].$$

  由于[14]中指出 conditional entropy可能不locally smooth. 为了提高在target domain上conditional entropy的approximation, 添加了一个local smoothness constraint

$$\mathbb{E}_{x_t \sim \mathcal{D}_t}\left[\max_{\|r\| \leq \xi} \mathrm{KL}(p_{\theta_C}(x_t)||p_{\theta_C}(x_t + r))\right]$$

  考虑增加一个微小的perturbation使得模型在x_t和x_t+r的输出相似, 这样提高local smooth, 整体的loss项如下

$$\begin{aligned}\ell_{cluReg} = &\mathbb{E}_{x_t \sim \mathcal{D}_t}[-p_{\theta_C}(x_t) \log p_{\theta_C}(x_t)] \\ &+ [\mathrm{KL}(p_{\theta_C}(x_t)||p_{\theta_C}(x_t + \tilde{r}))],\end{aligned}$$

- 总体算法流程为

Algorithm 1 Pseudo-code of our model adaptation process

Input: Pre-trained prediction model C on the source domain, unlabeled data X_t in the target domain, λ_g, λ_{clu} and λ_w, batch size B;

Output: θ_C for the prediction model C;

Initialize learning rates ζ_G, ζ_D and ζ_C for G, D and C;

for $epoch$ = 1 to N **do**
  Randomly sample x_t of size B from X_t, and random vectors $\{y, z\}$ from the uniform distribution;
  for each mini-batch **do**
    Generate new samples with y and z: $X_g = G(y, z)$
    Update D via $\theta_D \leftarrow \texttt{Adam}(\nabla_{\theta_D}(\sum_{x_t} \log D(x_t) + \sum_{y,z} \log D(1 - G(y, z))), \theta_D, \zeta_D)$.
    Update G via $\theta_G \leftarrow \texttt{Adam}(\nabla_{\theta_G}(\ell_{adv} + \lambda_s \ell_{sem}), \theta_G, \zeta_G)$
    if starting adaptation **then**
      Update C via $\theta_C \leftarrow \texttt{Adam}(\nabla_{\theta_C}(\lambda_g \ell_{gen} + \lambda_w \ell_{wReg} + \lambda_{clu} \ell_{cluReg}), \theta_C, \zeta_C)$
    end if
  end for
end for

experiment

对于每个task, 仅使用source data去训练模型, 而不用于adaption. 对于其他的方法, 可以使用source data, 并且取结果互相比较, 以供参考

Digit and sign datasets 使用 MNIST/USPS/MNIST-M/SVHN/Syn.Digits和2个交通数据集 Syn.Sign/GTSRB做实验. 数字数据集共有10个相同的类, 交通信号数据集有43个类. Syn.Sign和Syn.Digit是合成数据集, 因此更加的realistic

Office-31 标准的domain adaption benchmark. 图片来自Amazon(A) Webcam(W) DSLR(D)三个域, 并且共享31个类别, 分别包含2817, 795, 498个样本. 按照[43, 34]的实验标准来使用

VisDA17 一个大型的数据集, 从synthetic domain到real domain, 总共有12个类别. 由于这个数据集的source data数量很大, 因此我们可以用来说明我们方法的优越性

对于digit and sign数据集, 我们把所有图片都resize为32x32x3. 对于office-31和VisDA17, 使用在ImageNet上pre-trained的ResNet50和ResNet101作为feature extractor.

- result

Method	SVHN→MNIST	MNIST→USPS	USPS→MNIST	MNIST→MNIST-M	Syn.Digits→SVHN	Syn.Sign→GTSRB
Source-Only	76.4±1.5	92.4±1.7	86.1±1.3	54.2±0.9	86.2±0.9	78.3±1.6
DAN [31]	71.1	81.1	-	76.9	88	91.1
AssocDA [16]	97.6	-	-	89.5	91.8	97.6
DANN [11]	73.8	85.1	73.0	77.4	91.1	88.7
UNIT [29]	90.5	95.9	93.5	-	-	-
GenToAdapt [50]	92.4±0.9	95.3±0.7	90.8±1.3	-	-	-
DSN [3]	82.7	91.3	-	83.2	91.2	93.1
PixelDA [2]	-	95.9	-	98.2	-	-
CyCADA [18]	90.4±0.4	95.6±0.2	96.5±0.1	-	-	-
SimDA [45]	-	96.4	95.6	90.5	-	-
MCD [49]	96.2±0.4	94.2±0.7	94.1±0.3	-	-	94.4±0.3
VADA [52]	97.9	-	-	97.7	94.8	98.8
DIRT-T [52]	99.4	-	-	**98.9**	**96.1**	99.5
Our Model	**99.4**±0.1	**97.3**±0.2	**99.3**±0.1	98.5±0.2	95.9±0.2	**99.6**±0.1

Table 1. Classification accuracy (%) on digit and sign dataset. '-' denotes that the results are not reported.

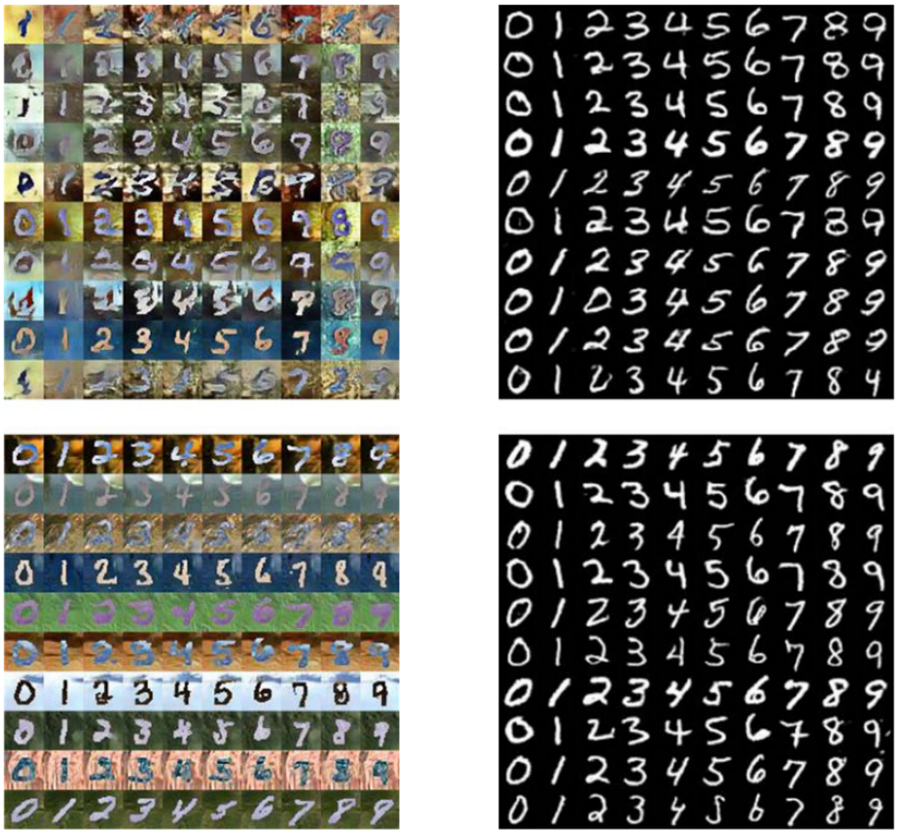
Method	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet50 [17]	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	65.2±0.3	60.7±0.3	76.1
DAN [31]	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
RTN [33]	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
DANN [11]	82.6±0.4	96.9±0.2	99.3±0.2	81.5±0.4	68.4±0.5	67.5±0.5	82.7
ADDA [57]	86.2±0.5	96.2±0.3	98.4±0.3	77.8±0.3	69.5±0.4	68.9±0.5	82.9
JAN [34]	86.0±0.4	96.7±0.3	99.7±0.1	85.1±0.4	69.2±0.4	70.7±0.5	84.6
MADA [43]	90.0±0.2	97.4±0.1	99.6±0.1	87.8±0.2	70.3±0.3	66.4±0.3	85.2
GenToAdapt [50]	89.5±0.5	97.9±0.3	99.8±0.2	87.7±0.5	72.8±0.3	71.4±0.4	86.5
Our Model	**93.7**±0.2	**98.5**±0.1	**99.8**±0.2	**92.7**±0.4	**75.3**±0.5	**77.8**±0.1	**89.6**

Table 2. Classification accuracy (%) on office-31 based on ResNet50 [17].

Method	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	Average
Source-Only	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN [31]	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
MCD [49]	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
SWD [26]	90.8	82.5	81.7	70.5	91.7	69.5	86.3	77.5	87.4	63.6	85.6	29.2	76.4
SimDA [45](ResNet152)	94.3	82.3	73.5	47.2	87.9	49.2	75.1	79.7	85.3	68.5	81.1	50.3	72.9
Self-Ensembling [9] (min aug)	92.9	84.9	71.5	41.2	88.8	92.4	67.5	63.5	84.5	71.8	83.2	48.1	74.2
Our Model	94.8	73.4	68.8	74.8	93.1	95.4	88.6	84.7	89.1	84.7	83.5	48.1	81.6
Our Model †	95.7	78.0	69.0	74.2	94.6	93.0	88.0	87.2	92.2	88.8	85.1	54.3	**83.3**

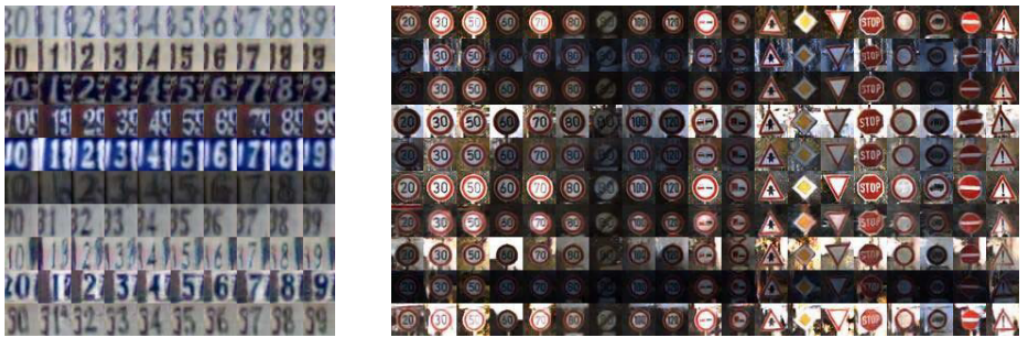
Table 3. Class-wise accuracy (%) on VisDA17 based on ResNet101 [17]. † denotes that we use an enhanced version of ResNet101 which replaces the first 7×7 convolution with three 3×3 convolutions.

- Visualization Analysis



(a) MNIST→MNIST-M (b) SVHN→MNIST

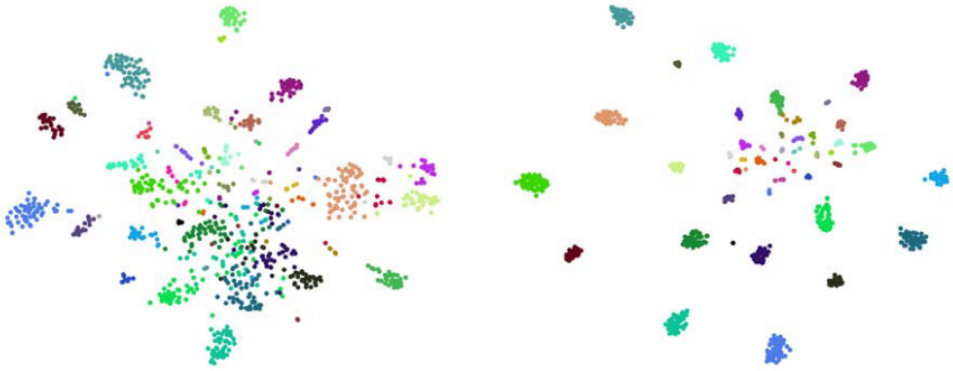
Figure 3. Class conditional generation in (a) MNIST→MNIST-M and (b) SVHN→MNIST. The top row indicates the samples generated with pre-trained source model, and the bottom row refers to the samples generated during the last adaptation stage.



(a) Syn.Digits→SVHN (b) Syn.Sign→GTSRB

Figure 4. Class-conditional generation in (a) Syn.Digits→SVHN and (b) Syn.Sign→GTSRB (shows the first 19 out of 43 classes). Each column has the same class y and the rows share the same noise vector z.

- 使用t-SNE投影分布



(a) Before Adaptation (b) After Adaptation

Figure 5. The t-SNE projection of the last hidden layer of target features (a) before adaptation and (b) after adaptation in the task of Syn.Sign→GTSRB. Different colors represent different classes.

- Ablation Study

  首先移除了 $loss_{gen}$, 这样会损害模型的disciminativity

  移除$loss_{weight}$和$loss_{cluster}$, 相对于source model可以显著提高性能