Multi-Source Domain Adaptation with Mixture of Experts

Jiang Guo, MIT, in EMNLP 2018

paper：https://www.aclweb.org/anthology/D18-1498.pdf

code：https://github.com/jiangfeng1124/transfer

任务定义

领域自适应：将资源丰富的源领域中的知识迁移到资源匮乏的目标领域，提升目标领域的性能。

目标领域只有一个，传统领域自适应只有一个源领域。（Cross-lingual, bilingual）

多源的领域自适应：现实中有多个源领域的数据可以获得，由此进行互补地迁移学习。（multilingual）

例子：

目标领域：kitchen（包含 pans, cookbooks, electronic devices）

源领域：分别对应的源领域 Cookware, Books, Electronics

相关工作

- 无监督领域自适应：目标领域没有标注数据
  - 将不同领域对齐到同一空间，通过训练使得模型在目标领域泛化好
    - 优点：简单
    - 缺点：丢失了不同领域的特性，甚至造成负迁移
    - **ours**：通过MOE捕捉不同领域的特性
- 多源领域自适应：关注不同源领域与目标领域之间的关系
  - 同等看待
  - 有监督地学习相似性度量，或者使用预先定义好的度量方法
    - domain2domain：无监督地学习数据分布的相似性，然后对源领域进行加权，构造伪训练集
    - example2domain：针对目标数据筛选训练数据、有监督的atten
    - **ours**：example2domain是否也能无监督

动机

- domain2domain粒度太粗，能否细粒度地度量example2domain：即point-to-set
  - 计算隐层表示的马氏距离：参数化的度量方式
- 目标领域资源匮乏，能否通过无监督的范式学习
  - meta-training：K个源领域，每次拿一个作为目标领域，其他做源领域
- 同时学习模型和度量方式
  - 对抗训练：使得鉴别器难以区分源领域和目标领域

模型结构

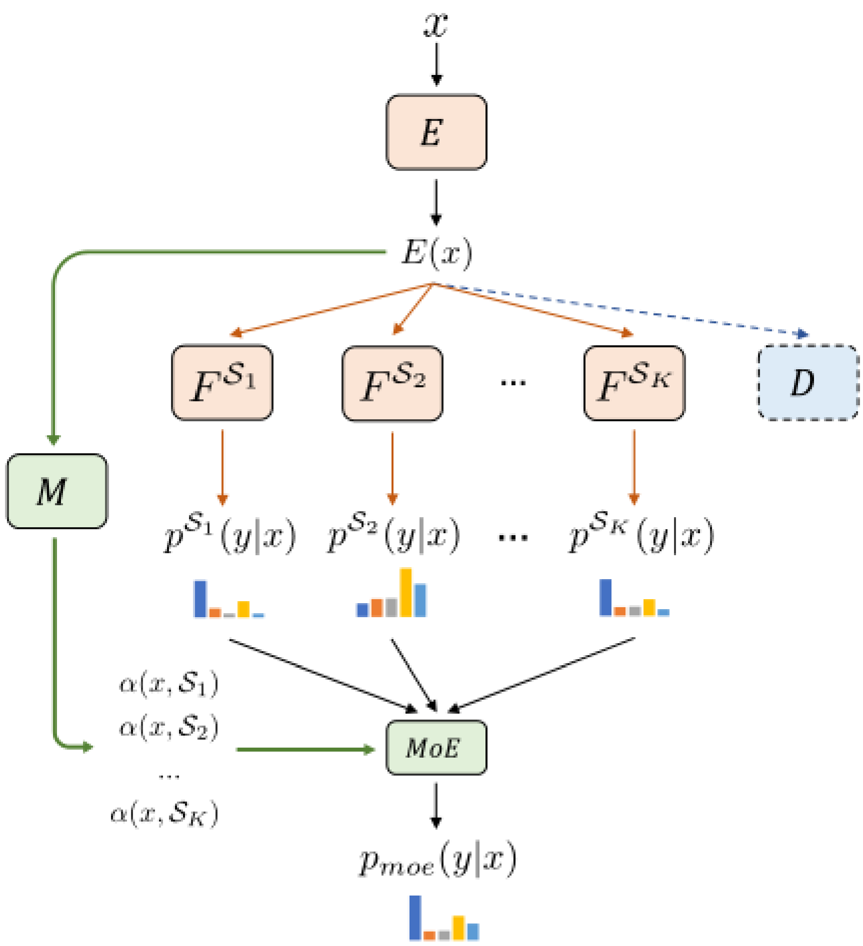

Figure 1: Architecture of the MoE model. E is the encoder which maps an input x to a hidden representation $E(x)$; $F^{\mathcal{S}_i}$ is the classifier on the i^{th} source domain; D is the critic that is only used during adversarial training. M is the metric learning component, which takes the encoding of x and source domains ($\mathcal{S}_{1:K}$) as input and computes α.

MOE 后验概率

- 多个领域训练通过多任务，学习每个领域自己的分类器

$$\mathcal{L}_{mtl} = -\sum_{i=1}^{K} \sum_{j=1}^{|\mathcal{S}_i|} \log p^{\mathcal{S}_i}(y_j^{\mathcal{S}_i} | x_j^{\mathcal{S}_i}) \qquad (4)$$

- 每个目标样本通过MOE学习后验概率

$$
\begin{aligned}
p_{moe}(y|x) &= \sum_{i=1}^{K} \alpha(x, \mathcal{S}_i) \cdot p^{\mathcal{S}_i}(y|x) \\
&= \sum_{i=1}^{K} \alpha(x, \mathcal{S}_i) \cdot \mathtt{softmax}\left(\mathbf{W}^{\mathcal{S}_i} E(x)\right)
\end{aligned}
$$

p^{S_i} 单隐层分类器，α计算权重

- 通过meta-training无监督计算MOE loss

K个Source，一个作为meta-target，其余作为meta-sources，组成K个meta-training tasks。

$$
\begin{aligned}
\mathcal{L}_{moe} &= -\sum_{i=1}^{K} \sum_{j=1}^{|\mathcal{S}_i|} \log p_{moe}(y_j^{\mathcal{S}_i} | x_j^{\mathcal{S}_i}) \\
&= -\sum_{i=1}^{K} \sum_{j=1}^{|\mathcal{S}_i|} \log \sum_{l=1, l \neq i}^{K} \alpha(x, \mathcal{S}_l) \cdot p^{\mathcal{S}_l}(y_j^{\mathcal{S}_i} | x_j^{\mathcal{S}_i})
\end{aligned} \tag{3}
$$

对抗训练

meta-sources的label是meta-target取反，通过交叉熵训练二分类

距离度量

马氏距离计算目标样本到单个源领域的距离

$$
d(x, \mathcal{S}) = \left(\left(E(x) - \mu^{\mathcal{S}} \right)^{\top} \mathbf{M}^{\mathcal{S}} \left(E(x) - \mu^{\mathcal{S}} \right) \right)^{\frac{1}{2}}
$$

置信分数由马氏距离计算得出

$$
e(x, \mathcal{S}_i) \;=\; f\left(d(x, \mathcal{S}_i)\right)
$$

然后Softmax归一化

$$
\alpha(x, \mathcal{S}_i) = \frac{\exp\left(e(x, \mathcal{S}_i)\right)}{\sum_{j=1}^{K} \exp\left(e(x, \mathcal{S}_j)\right)} \tag{2}
$$

熵正则化

$$H\big(\boldsymbol{\alpha}(x,\cdot)\big) = -\sum_{l=1}^{K} \alpha(x,\mathcal{S}_l)\cdot\log\alpha(x,\mathcal{S}_l)$$

$$\mathcal{R}_h = \sum_{i=1}^{K}\sum_{j=1}^{|\mathcal{S}_i|} H\big(\boldsymbol{\alpha}(x_j^{\mathcal{S}_i},\cdot)\big) \tag{6}$$

联合训练

根据权重调整几个loss

$$\begin{aligned}\mathcal{L} = \lambda\cdot\mathcal{L}_{moe} + (1-\lambda)\cdot\mathcal{L}_{mtl} \\ + \gamma\cdot\mathcal{L}_{adv} \\ + \eta\cdot\mathcal{R}_h\end{aligned} \tag{7}$$

训练过程

Algorithm 1 Training Procedure

Input: multi-source domain data $\mathcal{S}=\{\mathcal{S}_i\}_{i=1}^K$, target domain data $\mathcal{T}$
Hyper-parameters: mini-batch size m, coefficients for different losses: λ, γ and η
repeat
  Sample K source mini-batches $\{(\mathbf{x}^{\mathcal{S}_i},\mathbf{y}^{\mathcal{S}_i})\}_{i=1}^K$ from $\mathcal{S}$ and a target mini-batch $\mathbf{x}^{\mathcal{T}}$ from $\mathcal{T}$
  $\mathcal{L}_{mtl}, \mathcal{L}_{moe}, \mathcal{L}_{adv}, \mathcal{R}_h \leftarrow 0$
  for $t=1$ to K **do**
    Set ***meta-target*** as $\mathcal{T}^{meta} \triangleq \tilde{\mathcal{S}}_t \triangleq (\mathbf{x}^{\mathcal{S}_t},\mathbf{y}^{\mathcal{S}_t})$
    Set ***meta-sources*** as $\mathcal{S}^{meta} \triangleq \{\tilde{\mathcal{S}}_i\}_{i=1,i\neq t}^K$, where $\tilde{\mathcal{S}}_i \triangleq (\mathbf{x}^{\mathcal{S}_i},\mathbf{y}^{\mathcal{S}_i})$
    Compute cross-entropy loss over $\mathcal{T}^{meta}$, and add to $\mathcal{L}_{mtl}$
    Compute Mahalanobis metric $\alpha(x,\mathcal{S}')$ for each $x\in\mathcal{T}^{meta}$ and $\mathcal{S}'\in\mathcal{S}^{meta}$ ▷ Eq. (2)
    Compute MoE loss over $(\mathcal{S}^{meta},\mathcal{T}^{meta})$ using $\boldsymbol{\alpha}$, and add to $\mathcal{L}_{moe}$ ▷ Eq. (3)
    Compute entropy of $\boldsymbol{\alpha}(x,\cdot)$ for each $x\in\mathcal{T}^{meta}$, and add to $\mathcal{R}_h$ ▷ Eq. (6)
  end for
  Compute MMD between $\mathbf{x}^{\mathcal{T}}$ and $\cup_{i=1}^K\mathbf{x}^{\mathcal{S}_i}$, and add to $\mathcal{L}_{adv}$ ▷ Eq. (5)
  Update parameters via backpropagating gradients of the total loss $\mathcal{L}$ ▷ Eq. (7)
until converge

实验结果

亚马逊商品评论数据集

- 四个源领域 Books (B), DVDs (D), Electronics (E), and Kitchen appliances (K)
- 每个源领域1000个正例 1000个负例，由Chen12采样得到Ziser17
- 交叉验证的范式进行meta-training

SETTING	NON-ADVERSARIAL			ADVERSARIAL				
	best-SS	uni-MS	MoE	mSDA†	MDAN	best-SS-A	uni-MS-A	MoE-A
D,E,K–B	75.43	78.43	**79.42**	76.98	78.63	80.07	80.25	**80.87**
B,E,K–D	81.23	82.49	**83.35**	78.61	80.65	82.68	83.30	**83.99**
B,D,K–E	85.51	84.79*	**86.62**	81.98	85.34	86.32	85.96*	**86.38**
B,D,E–K	86.83	87.00	**87.96**	84.33	86.26	87.05	87.55	**88.06**
Average	*82.25*	*83.18*	***84.34***	*80.48*	*82.72*	*84.03*	*84.27*	***84.83***

Table 1: Multi-Source domain adaptation accuracy on Amazon dataset of CHEN12. * indicates negative transfer, i.e., the unified multi-source model underperforms the best single-source model. mSDA† is not an adversarial approach, but utilizes unlabeled data from target domain.

SETTING	NON-ADVERSARIAL			ADVERSARIAL		
	best-SS	uni-MS	MoE	best-SS-A	uni-MS-A	MoE-A
D,E,K–B	85.35	87.00	**87.55**	86.85	87.55	**87.85**
B,E,K–D	85.25	86.80	**87.85**	86.00	87.40	**87.65**
B,D,K–E	86.80	88.30	**89.20**	88.90	89.35	**89.50**
B,D,E–K	88.90	89.65	**90.45**	89.95	90.35	**90.45**
Average	*86.58*	*87.94*	***88.76***	*87.93*	*88.66*	***88.86***

Table 2: Multi-Source domain adaptation accuracy on Amazon dataset of ZISER17.

代码实现

训练集

train：3个源领域

unl：目标领域的训练（用于对抗训练）

input：b x 5000

hidden：b x 500

1. MTL 交叉熵

ms_outpus：list x domain_nums 每个domain的batch分别经过对应二分类器，只计算对角线

out: b x 2

item	domian 1 cls	domian 1 cls	domian 1 cls
domian 1 batch	out		
domian 2 batch		*	
domian 3 batch			*

2. KLLoss 均方误差

source_alphas：list x domain_nums

使用hidden计算每个样本与所有domain的**马氏距离**，按列Softmax

size: domains x b

item	example 1	example 2	...	example batch_szie
domian 1	alpha 1,1			
domian 2	alpha 1,2			
domian 3	alpha 1,1			

source_labels：每个样本对应的domain为1，其余为0

domain	example 1	example 2	...	example batch_szie
domian 1	1	1	...	1
domian 2	0	0	...	0
domian 3	0	0	...	0

拟合alpha的分布，相当于attention，这里计算均方误差，不同于交叉熵只关注label为1的项。

3. HLOSS 计算权重的熵

$$H(X) = \sum_{x \in X} -p(x)logp(x)$$

对2的补充，熵越小，越趋近于one-hot。

例：

p(x1)	p(x2)	p(x3)	熵
1	0	0	0
1/2	1/4	1/4	3/2log2
1/3	1/3	1/3	log3

4. MOE NLLLoss

source_alphas：3 x b

ms_outputs：[b x 2, b x 2, b x 2]

遍历每个domain，每个类别乘权重。

5. Domain adversarial network

目标领域训练集的标签为真实标签，源领域训练集的标签是目标领域标签的**反**，使得模型区分源/目标领域。

实现：hidden通过二分类器。

测试

train：3个源领域

test：目标领域的测试

domain编码：计算每个domain的正例，负例，所有数据的均值

计算距离，MOE后验概率

参考文献

Related Works in Semi-supervised Domain Adaptation for Dependency Parsing, Zhenghua Li, SUDA, in ACL19