Unsupervised Domain Adaptation

Outline

- Two domain adaptation models that perform explicit class alignment via cross-domain class prototypes on top of typical feature alignment
- Moving Semantic Transfer Network (MSTN) (Xie et al., ICML 2018)
  - The paper proposes a new module that pseudo-labels batches of unlabeled target examples and then aligns class prototypes of labeled source and newly pseudo-labeled target examples during each iteration
- Progressive Progressive Feature Alignment (PFAN) (Chen et al., CVPR 2019), an incremental improvement upon the MSTN model
  - The model separately pseudo-label all unlabeled examples based on encoded source prototypes the and selectively collect "easy" pseudo-labeled ones to form smaller pseudo-labeled target dataset
  - Then, the model performs class prototypes alignment
- Overall, both models do not provide good performance compared to the implicit joint domain-class alignment (JDCA), but these papers present good ideas on how to solve the class alignment problem (the idea of pseudo-labeling has been applied to many other topics)

Moving Semantic Transfer Network (MSTN): Class Alignment Objective

- New module: for each iteration with batches of labeled source and unlabeled target examples
  - pseudo-label the batch of unlabeled target examples
  - compute the local class prototypes of both labeled source and pseudo-labeled target batches
  - update the global class prototypes based on exponential moving average
  - (new objective) minimize the distance between the cross-domain global class prototypes

---

Algorithm : MSTN's procedure to compute the additional objective of class alignment

---

Require: labeled source examples $\mathcal{D}^s$, and unlabeled target examples $\mathcal{D}^t$

for epoch in num epochs **do**

  initialize global class prototypes of source and target domains $c^s_{k(I)}$ and $c^t_{k(I)}$ to be 0

  for iter i in num iters **do**

    get source and target batches of examples $\mathcal{B}^s_i \in \mathcal{D}^s$ and $\mathcal{B}^t_i \in \mathcal{D}^t$

    pseudo-label the batch of unlabeled target examples

    $\widehat{\mathcal{B}}^t_i = \{(\boldsymbol{x}^t_i, \widehat{y}^t_i)\}$

    $\widehat{y}^t_i = \mathrm{argmax}_k\, h(g(\boldsymbol{x}^t_i))$

    compute local (batch) class prototypes of both domains

    $c_{k(i)} = \frac{1}{|\mathcal{B}_i|} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{B}_i} g(\boldsymbol{x}_i)$

    update global class prototypes of both domains

    $c_{k(I)} = \alpha c_{k(I)} + (1-\alpha) c_{k(i)}$

    additional objective: aligning global class prototypes of both domains

    $L_{\text{ca}} = \sum_k^K \psi(c^s_{k(I)}, \widehat{c}^t_{k(I)})$

  end for

end for

Moving Semantic Transfer Network (MSTN): Notes

- We should use moving average global prototypes instead of local batch prototypes
  - it is possible that some classes are missing in the current batch since the batch is randomly selected
  - if the batch size is small, even one false pseudo-labeled example will lead to the large deviation between pseudo-labeled prototypes and true prototypes

Progressive Feature Alignment (PFAN): Class Alignment Object

- Easy-to-Hard Transfer Strategy (EHTS)
  - pseudo-label all target examples based on the encoded source prototypes and select most "easy" pseudo-labeled ones to form a pseudo-labeled dataset
- Adaptive Prototype Alignment (APA)
  - perform class alignment by computing and aligning the class prototypes of labeled source and pseudo-labeled target examples

Algorithm : PFAN's procedure to compute the additional objective of class alignment

Require: labeled source examples $\mathcal{D}^s$, and unlabeled target examples $\mathcal{D}^t$

for epoch in num epochs **do**
  $\mapsto$ EHTS
  compute the class prototypes of source domain
  $c_k^s = \frac{1}{|\mathcal{D}^s|} \sum_{(\boldsymbol{x}_i^s, y_i^s) \in \mathcal{D}_k^s} g(\boldsymbol{x}_i^s)$
  pseudo-label all unlabeled target examples, and select only "easy" examples that have similarity score above a threshold
  $\widehat{\mathcal{D}}^t = \{(\boldsymbol{x}_i^t, \widehat{y}_i^t)\}$
  $\widehat{y}_i^t = \mathrm{argmax}_k\, \psi(g(\boldsymbol{x}_i^t), c_k^s)$
  and $\max_k \psi(g(\boldsymbol{x}_i^t), c_k^s) > \tau$
  $\mapsto$ APA
  initialize global class prototypes of source and target domains $c_{k(I)}^s$ and $c_{k(I)}^t$ to be 0
  for iter i in num iters **do**
    get source and pseudo-labeled target batches of examples $\mathcal{B}_i^s \in \mathcal{D}^s$ and $\widehat{\mathcal{B}}_i^t \in \widehat{\mathcal{D}}^t$
    compute local (batch) class prototypes of both domains
    $c_{k(i)} = \frac{1}{|\mathcal{B}_i|} \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{B}_i} g(\boldsymbol{x}_i)$
    update global class prototypes of both domains
    $\overline{c}_{k(i)} = \frac{1}{i} \sum_{j=1}^{i} c_{k(j)}$
    $\rho_i = \psi(\overline{c}_{k(i)}, c_{k(I)})$
    $c_{k(I)} = \rho_i^2 c_{k(I)} + (1 - \rho_i^2)\overline{c}_{k(i)}$
    additional objective: aligning global class prototypes of both domains
    $L_{\text{ca}} = \sum_k^K \psi(c_{k(I)}^s, \widehat{c}_{k(I)}^t)$
  end for
end for

Progressive Feature Alignment (PFAN): EHTS

- EHTS: pseudo-label all target examples and select most "easy" pseudo-labeled ones to form a pseudo-labeled dataset
  - compute source prototypes c_k^S for every classes

$$c_k^s = \frac{1}{N_k^s} \sum_{(\boldsymbol{x}_i^s, y_i^s) \in \mathcal{D}_k^s} g(\boldsymbol{x}_i^s)$$

  - cluster and pseudo-label all unlabeled target examples to the corresponding source prototypes using a similarity metric

$$\widehat{y}_i = \underset{k}{\operatorname{argmax}}\, \psi(g(\boldsymbol{x}_i^T), c_k^S)$$

  - rank and select only "easy" pseudo-labeled examples that have similarity scores above an annealing threshold τ
    - ★ after each training epoch, since the model is better at performing representations and likely gathers more "easy" examples
    - ★ in contrast, we want to keep the selection ratio constant, so we gradually increase the threshold to harder the selection

Progressive Feature Alignment (PFAN): APA

- APA: perform class alignment by computing and aligning the class prototypes of labeled source and pseudo-labeled target examples
  - initialize global prototypes $c^s_{k(I)}$ and $c^t_{k(I)}$ of all examples from $\mathcal{D}^s$ and $\widehat{\mathcal{D}}^t$ respectively
  - at each iter i, compute local prototypes $c^s_{k(i)}$ and $c^t_{k(i)}$ of the batches $\mathcal{B}^s_i$ and $\widehat{\mathcal{B}}^t_i$ respectively, then update the global prototypes as follows

$$
\begin{aligned}
\overline{c}_{k(i)} &= \frac{1}{i}\sum_{j=1}^{i} c_{k(j)} \\
\rho_i &= \psi(\overline{c}_{k(i)}, c_{k(I)}) \\
c_{k(I)} &= \rho_i^2 \overline{c}_{k(i)} + (1-\rho_i^2) c_{k(I)}
\end{aligned}
$$

  - (new objective) minimize the distance between cross-domain global class prototypes

$$
L_{\text{apa}}(c^s_{k(I)}, c^t_{k(I)}) = \|c^s_{k(I)} - c^t_{k(I)}\|^2
$$

Progressive Feature Alignment (PFAN): Notes

- In order to prevent models biased over (over-reliance on) source classification, we suggest to gradually reduce and remove the convergence of the source classification by adding a controllable temperature variable into the last softmax output function

$$\widehat{q}_i = \frac{\exp z_i/T}{\sum_j \exp z_j/T}$$

Thank you !