Unsupervised Domain Adaptation via Regularized Conditional Alignment

Cicek et al., ICCV 2019
Overview of domain adversarial adaptation method (DA) and one of its issue

- Domain adversarial adaptation tries to align/map source and target examples into a common representation so that a class predictor can classify source examples can also perform on target examples
- However, it adapts only the domains not the classes e.g., a natural image of a cat can be mapped to a synthesis image of a dog

This paper proposes a new method: joint domain and class adversarial adaptation (JDCA)

- Instead of imposing a binary domain adversarial loss, it imposes a K-way binary adversarial loss (2K classification, the first K are the known source classes, and the second K are the unknown target classes)
- The encoder will try to fool the predictor by extracting invariant features from a synthesis image (source) and from a natural image (target) of examples of a specific class
Issues of Class Misalignment

- Simple scenario, when feature distributions are aligned (i.e. a domain classifier cannot distinguish which domain the extracted features belong to), when target examples are mapped into source examples with correct classes, the class predictor performs well in both domains.

- Realistic scenario, when target examples are mapped into source examples, some target examples from one class are mapped into source examples from a different class.
Issues of Class Misalignment: Ideal Solution

- Issue: when target examples are mapped into source examples, some target examples from one class are mapped into source examples from a different class.

- Ideally, we want a model that can simultaneously align examples from two domains and align examples’ classes.
Intuitive Visualization between DA and JDCA

- DA

- JDCA
Joint Domain and Class Adaptation Model

- Three components
  - Feature encoder $g$
  - Class predictor $h_c$ is trained only source examples, but is used for inference for both source and target examples
  - Joint predictor $h_j$ is trained to jointly align domains and classes of examples
- Note: the model is trained in adversarially using GAN style, not gradient reversal layer, so the model does not have the domain predictor
Joint Domain and Class Adaptation Model

- Class predictor $h_c$ is trained only source examples, but is used for inference for both source and target examples
  \[
  L_{sc}(\theta_g, \theta_{h_c}) = \text{CE}(h_c(g(x^s)), y^s)
  \]

- The encoder $g$ is trained to jointly align domains and classes of examples via adversarial mechanism with the help of the joint predictor
  - The joint predictor will try to distinguish true source and target labels
    \[
    L_{jsc}(\theta_{h_j}) = \text{CE}(h_j(g(x^s)), [y^s, 0])
    \]
    \[
    L_{jtc}(\theta_g, \theta_{h_j}) = \text{CE}(h_j(g(x^t)), [0, \hat{y}^t]),
    \]
    where $\hat{y}^t = \arg\max h_c(g(x^t))$
  - The encoder will try to fool the joint predictor by generating adversarial source and target labels
    \[
    L_{jsa}(\theta_g) = \text{CE}(h_j(g(x^s)), [0, y^s])
    \]
    \[
    L_{jta}(\theta_g) = \text{CE}(h_j(g(x^t)), [\hat{y}^t, 0]),
    \]
    where $\hat{y}^t = \arg\max h_c(g(x^t))$
Once source and target examples are aligned, we train the model solely on target examples (to focus on target representations) using semi-supervised learning model.

- Model can automatically predict pseudo-labels and train on its own.

This paper uses a entropy minimization and virtual adversarial training models for semi-supervised learning.
Thank you!
## Image Classification Benchmark

<table>
<thead>
<tr>
<th>Source dataset</th>
<th>MNIST</th>
<th>SVHN</th>
<th>CIFAR</th>
<th>STL</th>
<th>SYN-DIGITS</th>
<th>SVHN</th>
<th>MNIST-M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target dataset</td>
<td>MNIST-M</td>
<td>SVHN</td>
<td>MNIST-M</td>
<td>STL-CIFAR</td>
<td>SYN-DIGITS-SVHN</td>
<td>MNIST-M</td>
<td></td>
</tr>
<tr>
<td>[10] DANN*</td>
<td>60.6</td>
<td>68.3</td>
<td>78.1</td>
<td>62.7</td>
<td>90.1</td>
<td>94.6</td>
<td></td>
</tr>
<tr>
<td>[11] DRCN</td>
<td>40.05</td>
<td>82.0</td>
<td>66.37</td>
<td>58.86</td>
<td>NR</td>
<td>NR</td>
<td></td>
</tr>
<tr>
<td>[38] kNN-Ad</td>
<td>40.3</td>
<td>78.8</td>
<td>NR</td>
<td>NR</td>
<td>NR</td>
<td>NR</td>
<td>86.7</td>
</tr>
<tr>
<td>[36] ATT</td>
<td>52.8</td>
<td>86.2</td>
<td>NR</td>
<td>NR</td>
<td>92.9</td>
<td>94.2</td>
<td></td>
</tr>
<tr>
<td>[9] Π-model**</td>
<td>33.87</td>
<td>93.33</td>
<td>77.53</td>
<td>71.65</td>
<td>96.01</td>
<td>NR</td>
<td></td>
</tr>
<tr>
<td>[40] VADA</td>
<td>47.5</td>
<td>97.9</td>
<td>80.0</td>
<td>73.5</td>
<td>94.8</td>
<td>97.7</td>
<td></td>
</tr>
<tr>
<td>[40] DIRT-T</td>
<td>54.5</td>
<td>99.4</td>
<td>NR</td>
<td>75.3</td>
<td>96.1</td>
<td>98.9</td>
<td></td>
</tr>
<tr>
<td>[40] VADA + IN</td>
<td>73.3</td>
<td>94.5</td>
<td>78.3</td>
<td>71.4</td>
<td>94.9</td>
<td>95.7</td>
<td></td>
</tr>
<tr>
<td>[40] DIRT-T +IN</td>
<td>76.5</td>
<td>99.4</td>
<td>NR</td>
<td>73.3</td>
<td>96.2</td>
<td>98.7</td>
<td></td>
</tr>
<tr>
<td>[18] Co-DA</td>
<td>81.7</td>
<td>99.0</td>
<td>81.4</td>
<td>76.4</td>
<td>96.4</td>
<td>99.0</td>
<td></td>
</tr>
<tr>
<td>[18] Co-DA + DIRT-T</td>
<td>88.0</td>
<td>99.4</td>
<td>NR</td>
<td>77.6</td>
<td>96.4</td>
<td>99.1</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>89.19</td>
<td>99.33</td>
<td>81.65</td>
<td>77.76</td>
<td>96.22</td>
<td>99.47</td>
<td></td>
</tr>
<tr>
<td>Source-only (baseline)</td>
<td>44.21</td>
<td>70.58</td>
<td>79.41</td>
<td>65.44</td>
<td>85.83</td>
<td>70.28</td>
<td></td>
</tr>
<tr>
<td>Target-only</td>
<td>94.82</td>
<td>99.28</td>
<td>77.02</td>
<td>92.04</td>
<td>96.56</td>
<td>99.87</td>
<td></td>
</tr>
</tbody>
</table>