Maximum Classifier Discrepancy for Unsupervised Domain Adaptation

[CVPR, 2018]

2023. 04. 12.

김성수

Data **M**ining and **Q**uality **A**nalytics

Data Mining
Quality Analytics

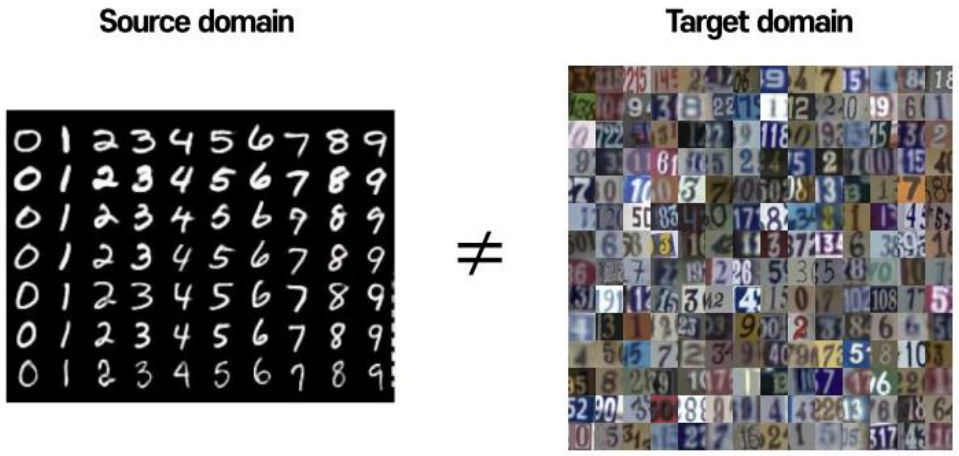
고려대학교
KOREA UNIVERSITY

연구배경

- Task명

❖ Task: Domain Adaptation

- Target Domain에서 데이터의 개수가 부족할 때, 데이터 양이 풍부한 Source Domain의 정보를 활용하여 모델을 학습하는 방법론
  - ✓ Source Domain도 Label을 갖고, Target Domain도 Label을 갖는 상황
  - ✓ 단순히 Source Domain으로 학습한 모델을 활용할 경우, Domain Gap이 존재하여, 일반화 성능이 떨어지는 문제 존재
- Source Domain을 활용하되, Domain Gap을 줄이면서 활용하는 것이 핵심!
  - ✓ **Source Domain의 정보를 효과적으로 활용하여 Target Domain의 성능을 높여보자!**

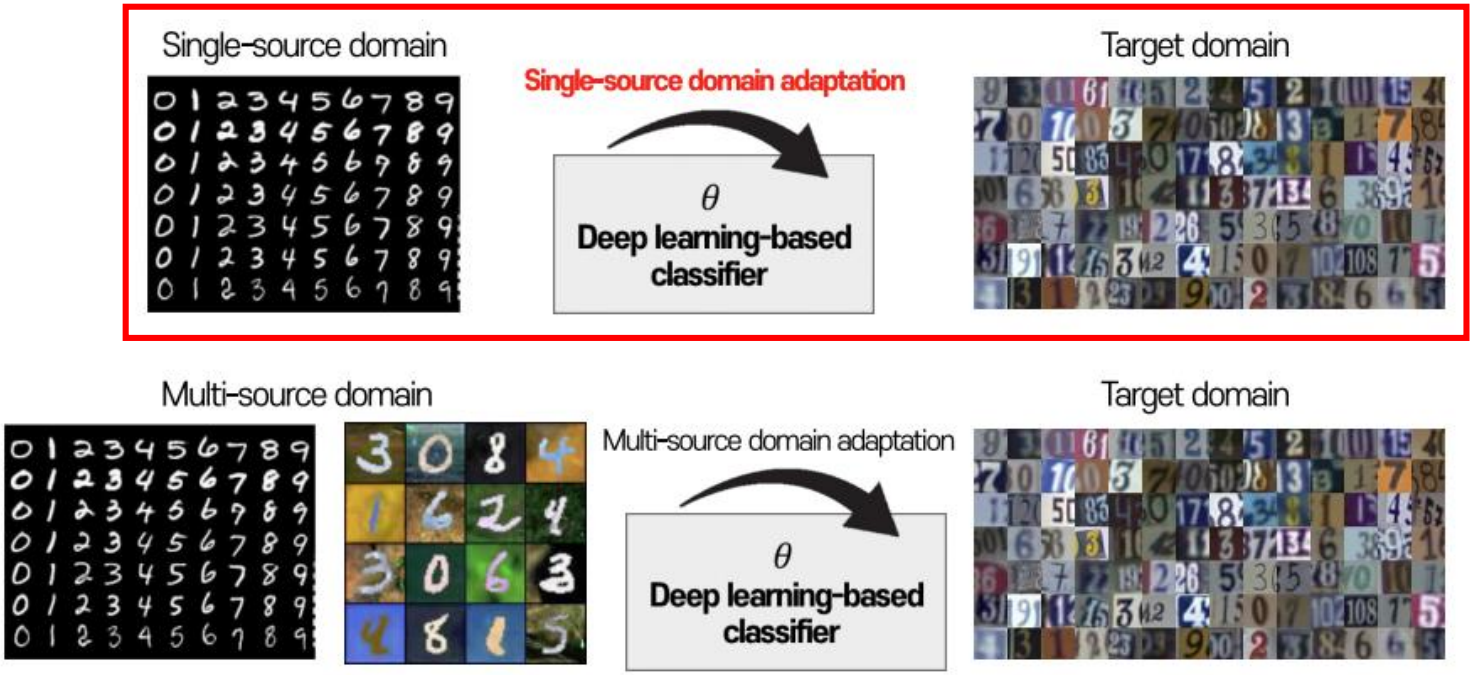


연구배경

- Task명

❖ Task: Single Source – Single Target Domain Adaptation

- 1개 Source Domain 정보를 활용하여 1개 Target 도메인의 성능을 향상시키고자 함.

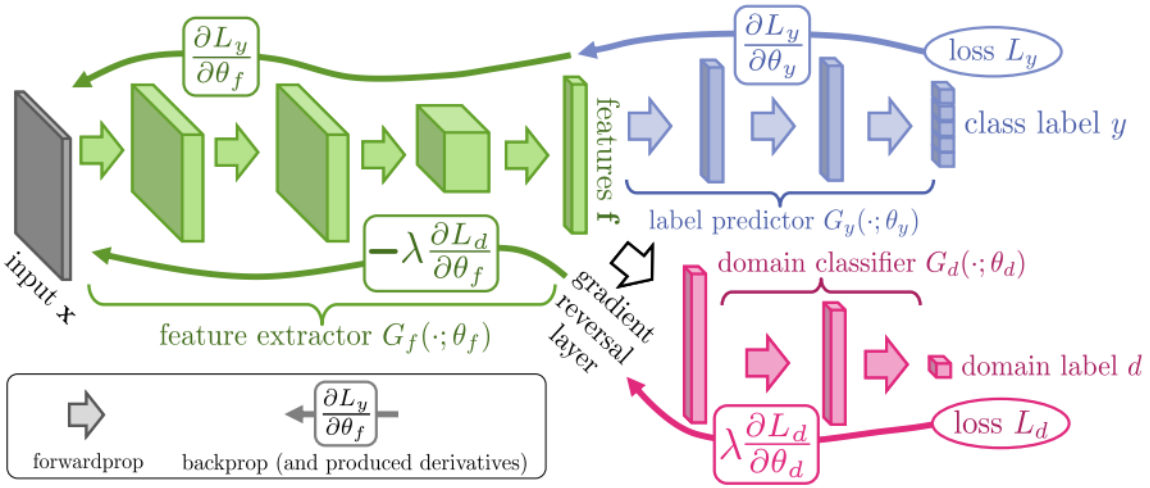


연구배경

- Task명

❖ Task: Unsupervised Domain Adaptation

- Source Domain은 Label을 가지지만, Target Domain은 Label을 갖지 않는 상황
- Idea: Source Domain의 정보를 Target Domain에 최대한 심어주어, Target Label이 없음에도 잘 학습시켜보자.
  - ✓ **<u>Label을 갖는 Source Domain은 훌륭한 Reference가 될 수 있다!</u>**
  - ✓ Ex) DANN (2015, JMLR)



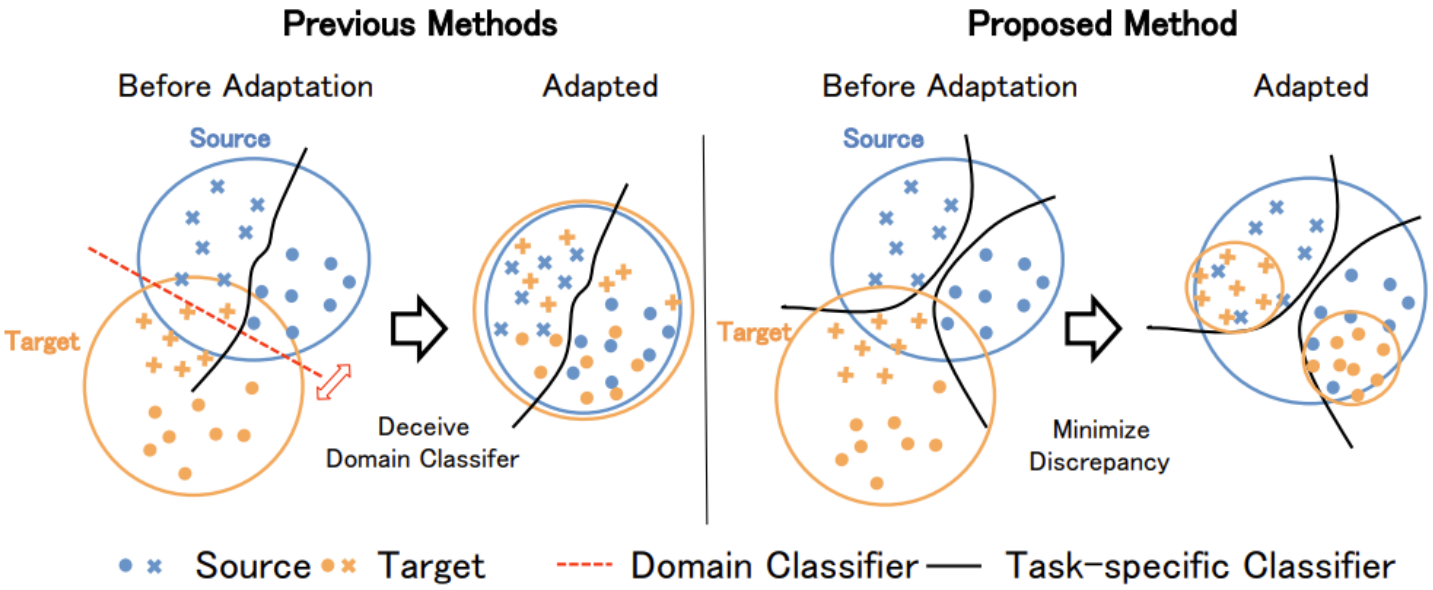
Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

연구배경

- 선행연구의 한계

❖ Limitation of Previous Research

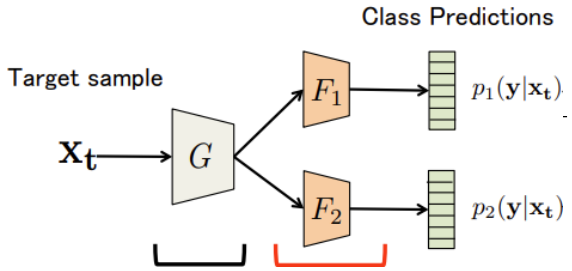
- Source 데이터를 Domain Gap을 줄이지 않고 활용한다면, 일반화 성능이 떨어짐
  - ✓ **Domain Gap 때문**
- 최근에 제안된 DANN은 Source와 Target 도메인은 구분하지만, 각 Class에 대한 구분이 미흡



Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

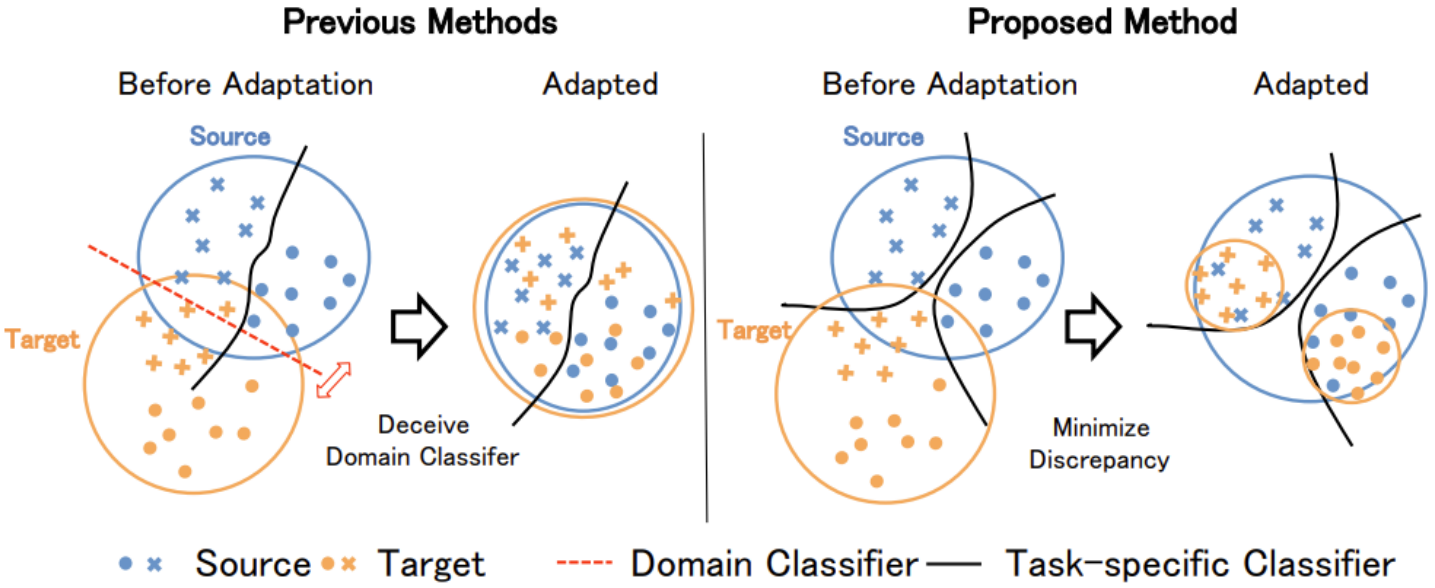
연구배경

- 선행연구의 한계를 극복과정 (Overview of Research)



❖ Overcome the Limitation

- Source/Target에 대한 구분 뿐만 아니라, 각 Class에 대한 구분까지 학습
  - ✓ Adversarial Learning 컨셉은 동일하지만, Class 구분까지 잘 학습
- 두 개의 Classifier를 활용하고, Domain Discrepancy를 정의하여 극복



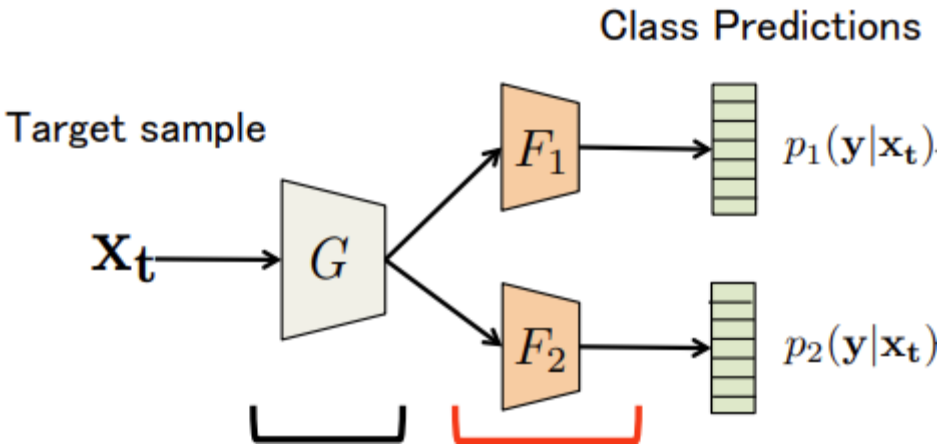
Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

연구배경

- Contribution

❖ Contribution

- Domain 구분 뿐만 아니라, Class 구분까지 잘하는 Adversarial Learning-based Unsupervised Domain Adaptation 제안
  - ✓ 두 Classifier의 Discrepancy로 Class 구분에 대해서도 잘 학습



G: Feature Extractor
F: Classifier

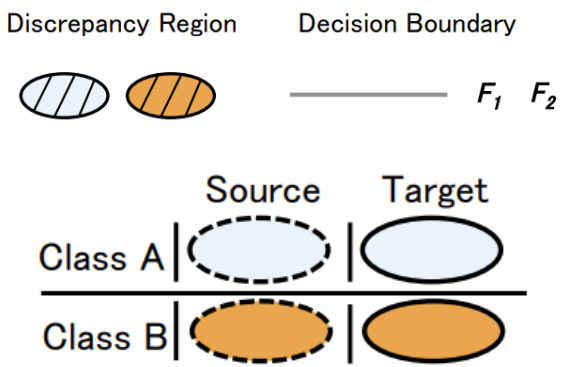
Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

방법론

- Ideation

❖ 방법론 아이디어

- Source의 분포에서 떨어진 데이터를 찾자 (=Boundary 근처 데이터)
  - ✓ 이러한 데이터는 두 Classifier를 통과했을 때, 서로 다른 결과를 갖게 될 것이다. (학습하지 못한 데이터이기에)
  - ✓ 두 Classifier는 서로 다른 가중치로 초기화되기에, 비슷하지만 다른 Weight로 학습된다.
- 가정: 두 Classifier는 Source 데이터에 대한 Class 구분은 잘 할 것이다.
  - ✓ Label을 갖는 Source 데이터로 학습했기 때문
  - ✓ 그러므로 Target 데이터 중, Classifier에서 서로 다른 결과를 갖는 것은 Source 도메인의 범위에서 벗어난 데이터일 것이다.



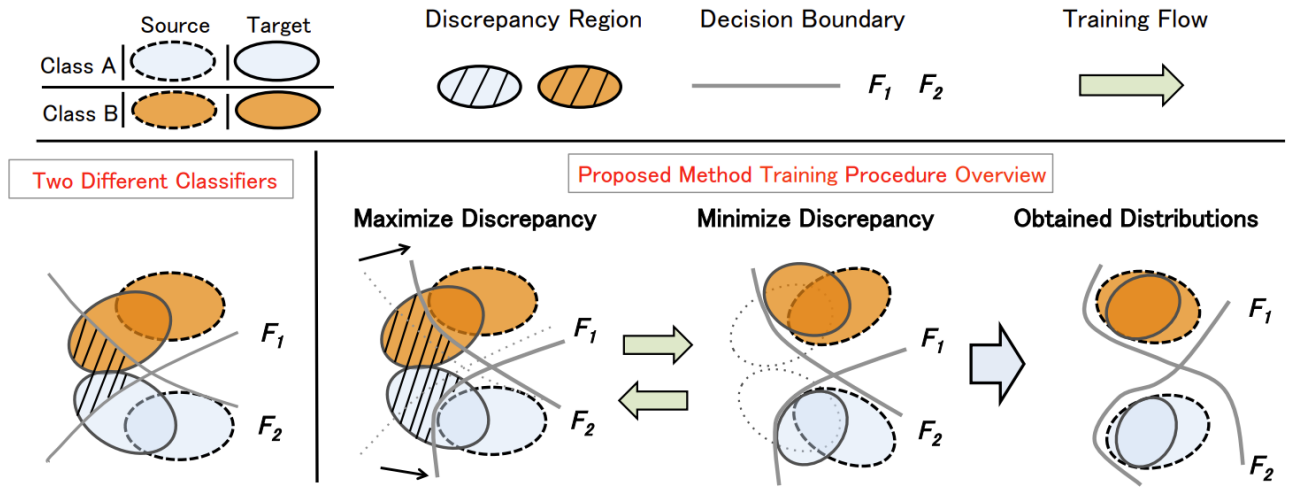
Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

방법론

- Ideation

❖ 방법론 아이디어

- 이처럼 Source의 분포에서 떨어진 데이터를 찾기 위해, Discrepancy를 최대화하도록 학습
  - ✓ 이를 통해 Domain Boundary를 학습 가능
  - ✓ 빗금 친 부분이 커지도록 (두 도메인간 구분이 확실해지도록 학습)
- 두 Classifier를 속이는 Feature Extractor를 학습 → Discrepancy를 최소화하도록 학습
  - ✓ Target Sample이 Source 분포 안으로 들어가도록 학습
- 최종 목표는 Target이 Source 분포 안에 완벽히 들어가도록 학습



Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

방법론

- 세부내용

❖ Details

- Discrepancy Loss: 두 Classifier에서 나온 확률분포의 차이 (L1-norm)
  - ✓ 논문에서는 L2-norm은 효과가 없었다고 언급
- 학습은 크게 3단계로 구성 (1번 iteration 시, 3번의 Update를 진행)

$$d(p_1, p_2) = \frac{1}{K} \sum_{k=1}^{K} |p_{1k} - p_{2k}|$$

Classifier2의 Class k에 대한 확률

Classifier1의 Class k에 대한 확률

```
for batch_idx, data in enumerate(self.datasets):
    img_t = data['T']
    img_s = data['S']
    label_s = data['S_label']
    if img_s.size()[0] < self.batch_size or img_t.size()[0] < self.batch_size:
        break
    img_s = img_s.cuda()
    img_t = img_t.cuda()
    imgs = Variable(torch.cat((img_s, \
                               img_t), 0))
    label_s = Variable(label_s.long().cuda())

    img_s = Variable(img_s)
    img_t = Variable(img_t)
    self.reset_grad()
    feat_s = self.G(img_s)
    output_s1 = self.C1(feat_s)
    output_s2 = self.C2(feat_s)

    loss_s1 = criterion(output_s1, label_s)
    loss_s2 = criterion(output_s2, label_s)
    loss_s = loss_s1 + loss_s2
    loss_s.backward()
    self.opt_g.step()
    self.opt_c1.step()
    self.opt_c2.step()
    self.reset_grad()

    feat_s = self.G(img_s)
    output_s1 = self.C1(feat_s)
    output_s2 = self.C2(feat_s)
    feat_t = self.G(img_t)
    output_t1 = self.C1(feat_t)
    output_t2 = self.C2(feat_t)

    loss_s1 = criterion(output_s1, label_s)
    loss_s2 = criterion(output_s2, label_s)
    loss_s = loss_s1 + loss_s2
    loss_dis = self.discrepancy(output_t1, output_t2)
    loss = loss_s - loss_dis
    loss.backward()
    self.opt_c1.step()
    self.opt_c2.step()
    self.reset_grad()
```

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

방법론

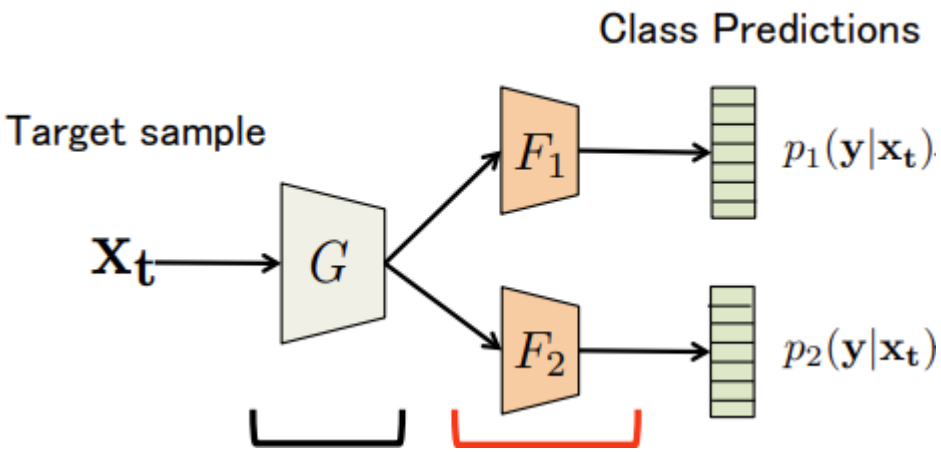
- 세부내용

❖ Details

- Phase1: Source 데이터만으로 Feature Extractor와 Classifier를 모두 학습
- 전체적인 Classifier의 분류능력을 키우자.
- 일반적인 Cross Entropy 활용

$$\min_{G,F_1,F_2} \mathcal{L}(X_s, Y_s)$$

$$\mathcal{L}(X_s, Y_s) = -\mathbb{E}_{(\mathbf{x_s}, y_s)\sim(X_s, Y_s)} \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log p(\mathbf{y}|\mathbf{x}_s)$$



Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

방법론

$$\mathcal{L}(X_s, Y_s) = -\mathbb{E}_{(\mathbf{x_s}, y_s)\sim(X_s, Y_s)} \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log p(\mathbf{y}|\mathbf{x}_s)$$

- 세부내용

$$d(p_1, p_2) = \frac{1}{K} \sum_{k=1}^{K} |p_{1k} - p_{2k}|$$

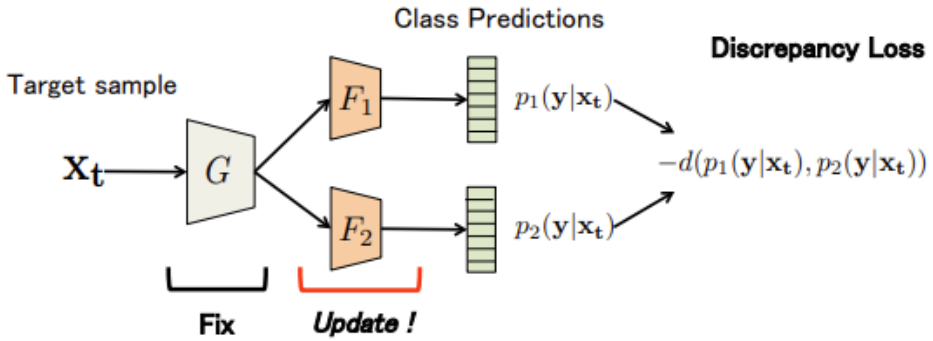
❖ Details

- Phase2: Source 데이터와 Target 데이터를 모두 활용하여 Discrepancy를 Maximize하도록 학습
- Feature Extractor는 Fix, Classifier만 학습
- Source 데이터: Classifier Loss가 낮아지도록
- Target 데이터: 두 Output의 차이가 커지도록

$$\downarrow \min_{F_1, F_2} \mathcal{L}(X_s, Y_s) - \mathcal{L}_{\text{adv}}(X_t).$$

$$\downarrow \qquad \uparrow$$

$$\mathcal{L}_{\text{adv}}(X_t) = \mathbb{E}_{\mathbf{x_t}\sim X_t}[d(p_1(\mathbf{y}|\mathbf{x_t}), p_2(\mathbf{y}|\mathbf{x_t}))]$$



Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

방법론

$$d(p_1, p_2) = \frac{1}{K}\sum_{k=1}^{K}|p_{1k} - p_{2k}|$$

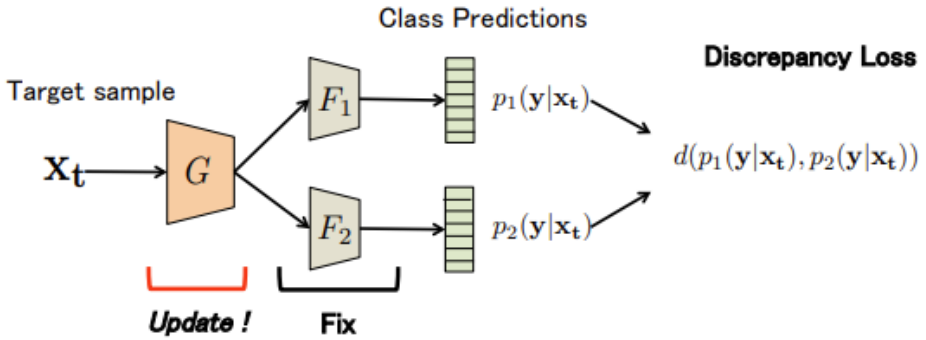
- 세부내용

❖ Details

- Phase3: Target 데이터만 활용하여 Discrepancy를 최소화 하도록 학습
- Classifier는 Fix, Feature Extractor만 학습
- Phase3는 $n(n \geq 2)$번 update할 경우, 좋은 성능을 보임

$$\min_{G} \mathcal{L}_{\text{adv}}(X_t).$$

$$\mathcal{L}_{\text{adv}}(X_t) = \mathbb{E}_{\mathbf{x_t} \sim X_t}[d(p_1(\mathbf{y}|\mathbf{x_t}), p_2(\mathbf{y}|\mathbf{x_t}))]$$



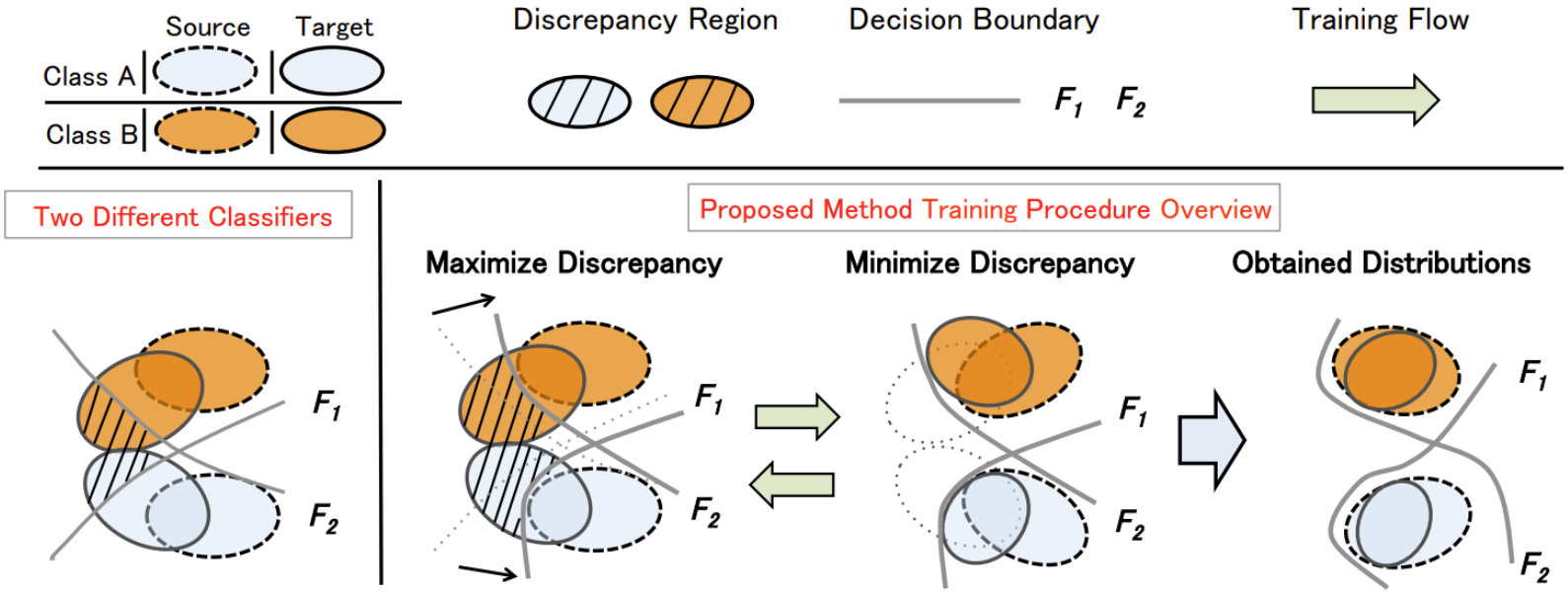
Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

방법론

- 세부내용

❖ Details

- Phase2가 학습될 때, 빗금 친 부분이 커지도록 학습 (F1과 F2의 위치가 이동)
- Phase3가 학습될 때, Target의 원이 Source의 원으로 들어가도록 학습 (원이 이동)



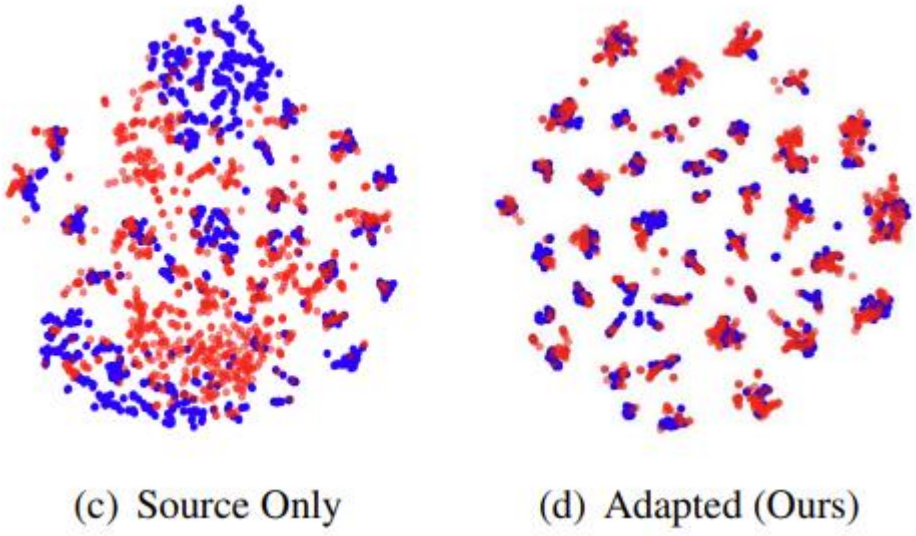
실험결과

- Result

❖ 실험결과

- 본 방법론 적용 시, t-SNE에서 Target이 Source의 분포를 잘 따라감을 확인
- Phase3에서 n이 커질수록 좋은 성능
- 정성적으로 보았을 때도, 좋은 결과를 보임

METHOD	SVHN to MNIST	SYNSIG to GTSRB	MNIST to USPS	MNIST* to USPS*	USPS to MNIST
Source Only	67.1	85.1	76.7	79.4	63.4
Distribution Matching based Methods					
MMD † [21]	71.1	91.1	-	81.1	-
DANN † [7]	71.1	88.7	77.1±1.8	85.1	73.0±0.2
DSN † [4]	82.7	93.1	91.3	-	-
ADDA [39]	76.0±1.8	-	89.4±0.2	-	90.1±0.8
CoGAN [19]	-	-	91.2±0.8	-	89.1±0.8
PixelDA [3]	-	-	-	95.9	-
Ours ($n = 2$)	94.2±2.6	93.5±0.4	92.1±0.8	93.1±1.9	90.0±1.4
Ours ($n = 3$)	95.9±0.5	94.0±0.4	93.8±0.8	95.6±0.9	91.8±0.9
Ours ($n = 4$)	**96.2**±0.4	**94.4**±0.3	**94.2**±0.7	**96.5**±0.3	**94.1**±0.3
Other Methods					
ATDA † [32]	86.2	96.2	-	-	-
ASSC [11]	95.7±1.5	82.8±1.3	-	-	-
DRCN [9]	82.0±0.1	-	91.8±0.09	-	73.7±0.04



(c) Source Only (d) Adapted (Ours)

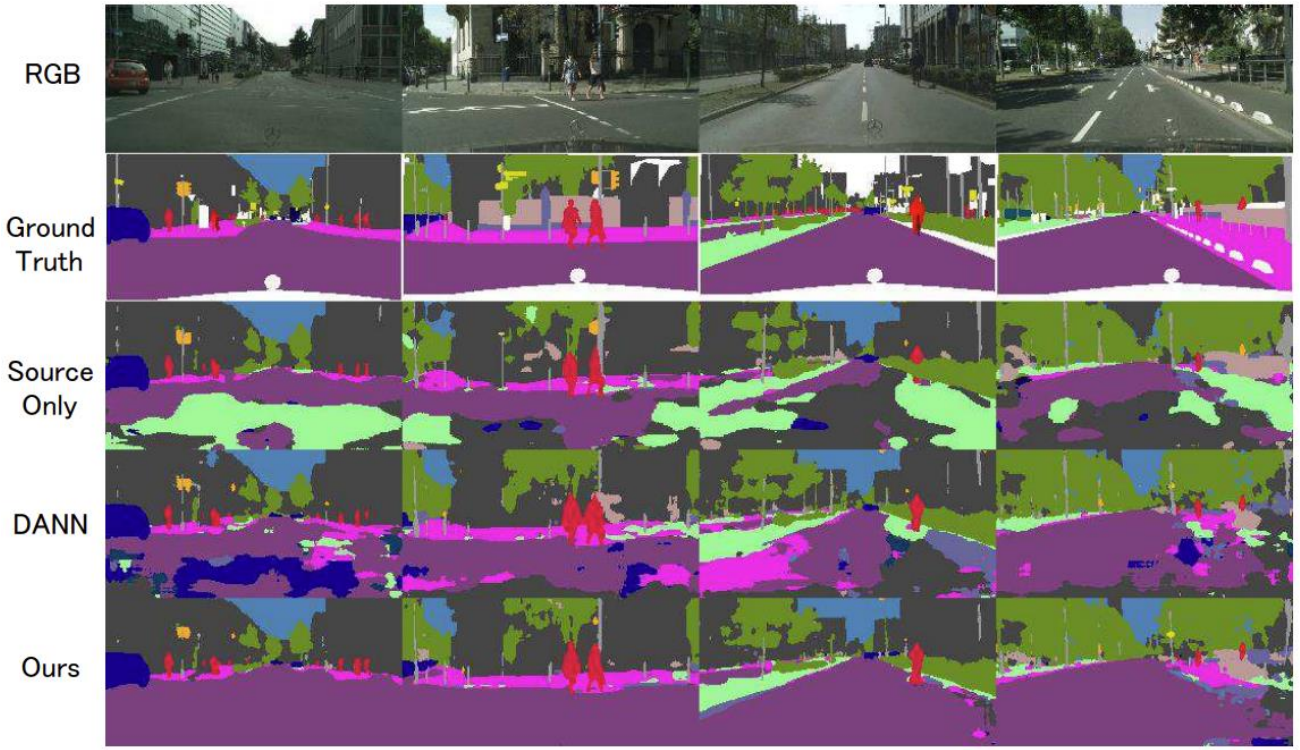
Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.

실험결과

- Result

❖ 실험결과

- 본 방법론 적용 시, t-SNE에서 Target이 Source의 분포를 잘 따라감을 확인
- Phase3에서 n이 커질수록 좋은 성능
- 정성적으로 보았을 때도, 좋은 결과를 보임



Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... & Lempitsky, V. (2016). Domain-adversarial training of neural networks. The journal of machine learning research, 17(1), 2096-2030.