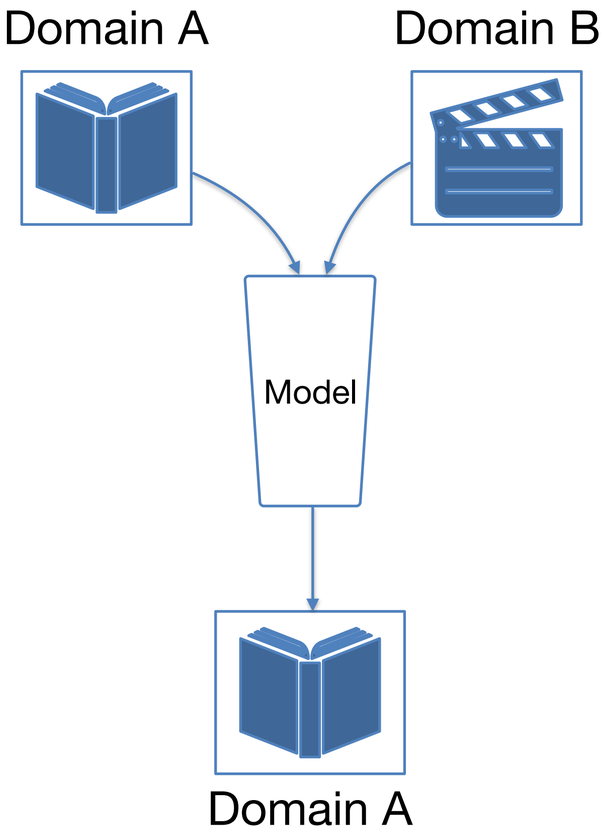
Domain Adaptation

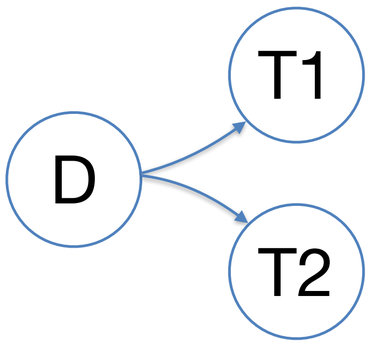
KyungJae Cho
Researcher, VUNO Inc.

Domain Adaptation (abstract)

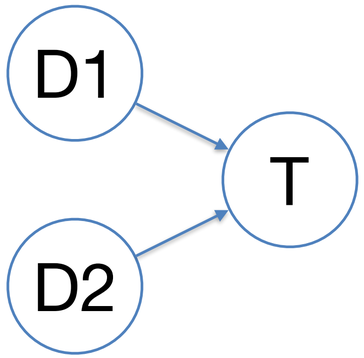
- 어떤 두 개의 domain들이 있으면 한 쪽을 다른 쪽으로 조정하거나 맞추려는 (adapt) 것
- Example
  - Domain A: 영화에 대한 리뷰가 긍정적인지 부정적인지 판별하는 도메인
  - Domain B: 책에 대한 리뷰를 판별하는 도메인
- 다른 두 도메인을 한 쪽의 도메인으로 맞추면, 두 도메인에 해당되는 데이터를 모두 사용하여 성능을 개선 시킬 수 있음



- Multi-task와는 다른 개념



Multi task learning



Domain Adaptation

Domain Adaptation (problem definition)

We consider classification tasks where X is the input space and $Y = \{0, 1, \ldots, L-1\}$ is the set of L possible labels. Moreover, we have two different distributions over $X \times Y$, called the *source domain* $\mathcal{D}_{\mathrm{S}}$ and the *target domain* $\mathcal{D}_{\mathrm{T}}$. An *unsupervised domain adaptation* learning algorithm is then provided with a *labeled source sample* S drawn *i.i.d.* from $\mathcal{D}_{\mathrm{S}}$, and an *unlabeled target sample* T drawn *i.i.d.* from $\mathcal{D}_{\mathrm{T}}^{X}$, where $\mathcal{D}_{\mathrm{T}}^{X}$ is the marginal distribution of $\mathcal{D}_{\mathrm{T}}$ over X.

$$S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n} \sim (\mathcal{D}_{\mathrm{S}})^n ; \quad T = \{\mathbf{x}_i\}_{i=n+1}^{N} \sim (\mathcal{D}_{\mathrm{T}}^{X})^{n'},$$

with $N = n + n'$ being the total number of samples. The goal of the learning algorithm is to build a classifier $\eta : X \to Y$ with a low *target risk*

$$R_{\mathcal{D}_{\mathrm{T}}}(\eta) = \Pr_{(\mathbf{x},y)\sim\mathcal{D}_{\mathrm{T}}} \Big(\eta(\mathbf{x}) \neq y\Big),$$

while having no information about the labels of $\mathcal{D}_{\mathrm{T}}$.

- Source domain에 해당되는 data와 target domain에 해당 되는 data를 기반으로 classifier를 학습하여 **low target risk**를 갖도록 학습하는 것이 목표
- **전제는 target domain은 label이 없음, 결국 target domain의 data은 model을 generalize하는데 사용됨 (regularizer)**
- Low target risk?
  - Target domain의 label을 잘 맞추는 것

Domain Adaptation (main idea)

- **(1) Discriminativeness**는 유지하면서 **(2) domain-invariance**를 고려하여 모델을 학습

- Classification의 역할을 잘하도록 유지하되 input으로 들어가는 **data가 어떤 domain에서 왔는지를 구별하지 못하게** domain discriminator를 약화하는 방향으로 학습시키겠다는 것

Domain Divergence

- Domain divergence (e.g., H-divergence)?
  - source와 target 분포간의 distance를 구하는 measure

Binary classifier들의 집합

Definition 1 (Ben-David et al., 2006, 2010; Kifer et al., 2004) *Given two domain distributions* $\mathcal{D}_{\mathrm{S}}^X$ *and* $\mathcal{D}_{\mathrm{T}}^X$ *over* X*, and a hypothesis class* $\mathcal{H}$*, the* $\mathcal{H}$-divergence *between* $\mathcal{D}_{\mathrm{S}}^X$ *and* $\mathcal{D}_{\mathrm{T}}^X$ *is*

H-divergence

$$d_{\mathcal{H}}(\mathcal{D}_{\mathrm{S}}^X, \mathcal{D}_{\mathrm{T}}^X) = 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_{\mathbf{x} \sim \mathcal{D}_{\mathrm{S}}^X} [\eta(\mathbf{x}) = 1] - \Pr_{\mathbf{x} \sim \mathcal{D}_{\mathrm{T}}^X} [\eta(\mathbf{x}) = 1] \right|.$$

모델이 S 도메인과 T 도메인의 데이터에 대해서 동일한 label을 부여한다면 domain divergence가 낮음

That is, the $\mathcal{H}$-divergence relies on the capacity of the hypothesis class $\mathcal{H}$ to distinguish between examples generated by $\mathcal{D}_{\mathrm{S}}^X$ from examples generated by $\mathcal{D}_{\mathrm{T}}^X$. Ben-David et al. (2006, 2010) proved that, for a symmetric hypothesis class $\mathcal{H}$, one can compute the *empirical* $\mathcal{H}$*-divergence* between two samples $S \sim (\mathcal{D}_{\mathrm{S}}^X)^n$ and $T \sim (\mathcal{D}_{\mathrm{T}}^X)^{n'}$ by computing

$$\hat{d}_{\mathcal{H}}(S, T) = 2 \left(1 - \min_{\eta \in \mathcal{H}} \left[\frac{1}{n} \sum_{i=1}^{n} I[\eta(\mathbf{x}_i) = 0] + \frac{1}{n'} \sum_{i=n+1}^{N} I[\eta(\mathbf{x}_i) = 1] \right] \right), \quad (1)$$

where $I[a]$ is the indicator function which is 1 if predicate a is true, and 0 otherwise.

- H-divergence
  - 특정 space (H)에 있는 Binary classifier들 중에서 두 도메인을 가장 잘 구별하는 값 (supremum)
- 두 도메인의 데이터들에 대해서 동일한 label을 부여한다면 도메인을 잘 구별하지 못한다는 것을 의미함

Domain Adaptation theory

The work of Ben-David et al. (2006, 2010) also showed that the $\mathcal{H}$-divergence $d_{\mathcal{H}}(\mathcal{D}_{\mathrm{S}}^{X}, \mathcal{D}_{\mathrm{T}}^{X})$ is upper bounded by its empirical estimate $\hat{d}_{\mathcal{H}}(S, T)$ plus a constant complexity term that depends on the *VC dimension* of $\mathcal{H}$ and the size of samples S and T. By combining this result with a similar bound on the source risk, the following theorem is obtained.

- H 공간의 크기에 따라서, H-divergence의 값이 달라지기때문에 upper bound를 계산하여, 값을 추정해야함
- H-divergence <= Empirical risk (classification error) + Model complexity
- 최종적으로, Domain adaptation의 목적인, low target risk를 구하기위한 수식을 아래와 같이 이끌어낼수있음

Theorem 2 (Ben-David et al., 2006) *Let $\mathcal{H}$ be a hypothesis class of VC dimension d. With probability $1-\delta$ over the choice of samples $S \sim (\mathcal{D}_{\mathrm{S}})^n$ and $T \sim (\mathcal{D}_{\mathrm{T}}^{X})^n$, for every $\eta \in \mathcal{H}$:*

Target risk →

$$R_{\mathcal{D}_{\mathrm{T}}}(\eta) \leq R_S(\eta) + \sqrt{\frac{4}{n}\left(d \log \frac{2e\,n}{d} + \log \frac{4}{\delta}\right)} + \hat{d}_{\mathcal{H}}(S,T) + 4\sqrt{\frac{1}{n}\left(d \log \frac{2n}{d} + \log \frac{4}{\delta}\right)} + \beta\,,$$

with $\beta \geq \inf_{\eta^* \in \mathcal{H}} \left[R_{\mathcal{D}_{\mathrm{S}}}(\eta^*) + R_{\mathcal{D}_{\mathrm{T}}}(\eta^*)\right]$, *and*

$$R_S(\eta) = \frac{1}{n}\sum_{i=1}^{m} I\left[\eta(\mathbf{x}_i) \neq y_i\right]$$

Low target risk

H-divergence 추정 값

Theorem 2 (Ben-David et al., 2006) *Let $\mathcal{H}$ be a hypothesis class of VC dimension d. With probability $1-\delta$ over the choice of samples $S \sim (\mathcal{D}_{\mathrm{S}})^n$ and $T \sim (\mathcal{D}_{\mathrm{T}}^X)^n$, for every $\eta \in \mathcal{H}$:*

$$R_{\mathcal{D}_{\mathrm{T}}}(\eta) \leq R_S(\eta) + \sqrt{\frac{4}{n}\left(d \log \frac{2e\,n}{d} + \log \frac{4}{\delta}\right)} + \hat{d}_{\mathcal{H}}(S,T) + 4\sqrt{\frac{1}{n}\left(d \log \frac{2n}{d} + \log \frac{4}{\delta}\right)} + \beta\,,$$

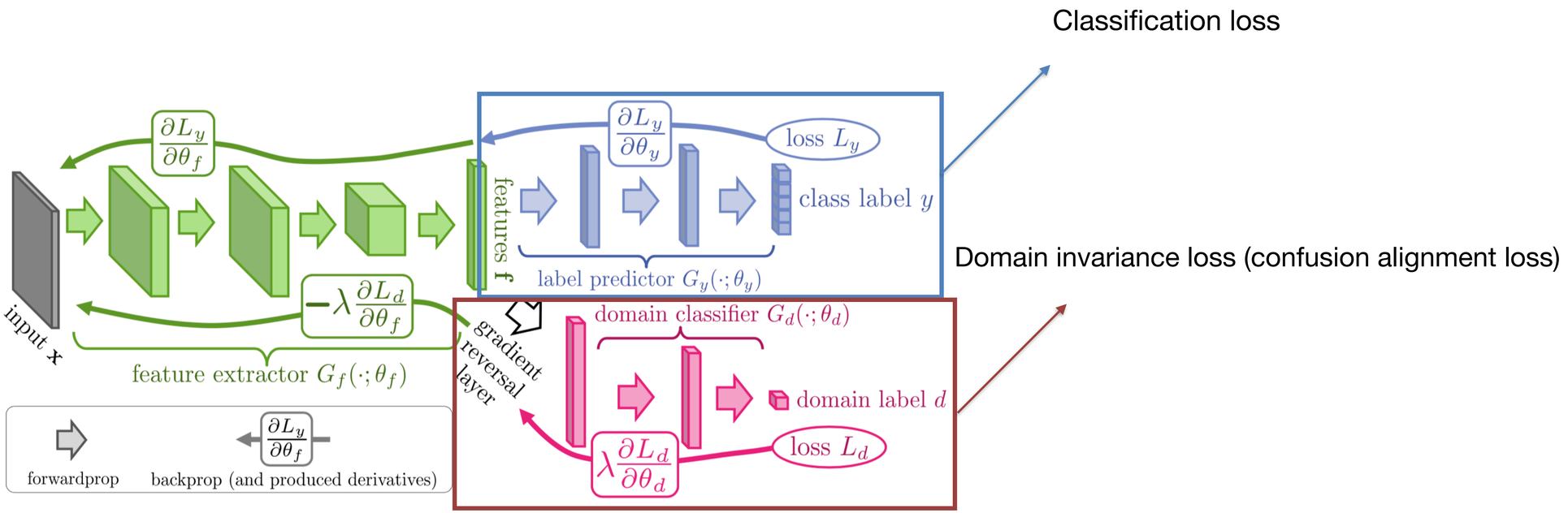
with $\beta \geq \inf_{\eta^* \in \mathcal{H}} \left[R_{\mathcal{D}_{\mathrm{S}}}(\eta^*) + R_{\mathcal{D}_{\mathrm{T}}}(\eta^*)\right]$, *and*

Low target risk를 위해서는 결국 upper bound가 최소가 되도록하면 됨

$$R_S(\eta) = \frac{1}{n}\sum_{i=1}^{m} I\left[\eta(\mathbf{x}_i) \neq y_i\right]$$

- Low target risk를 위해서는 아래 요소들을 최소화시켜줘야함
  - Source risk (R_s)
  - H-divergence (d_H(S,T) + model complexity)
  - Beta
  - VC dimension d (H 공간의 크기를 의미함)
- 위 요소들을 최소화 시켜야함
  - Source risk를 최소화하기 위해서는, source domain의 classification 성능을 높이고 **(low source risk, discriminativeness)**,
  - Source 도메인과 target 도메인 invariant하게 모델을 학습시켜야함 **(low H-divergence, domain-invariance)**
  - Beta는 H의 공간에 따라서 값이 달라지는데, **H의 공간을 너무 크게 잡으면 d 값이 높아져서**, target risk가 높아짐 **(적당한 complexitiy의 모델을 선택해야함)**

Domain-Adversarial Neural Network (DANN)



- **(1) Discriminativeness**는 유지하면서 **(2) domain-invariance**를 고려하기위한 모델 (end-to-end)
  - Discriminativeness -> classification loss를 최소화
  - Domain-invariance -> Domain invariance loss를 최대화
- Gradient reversal layer의 필요성
  - Domain-invariant한 모델을 학습하기위해서는 model이 도메인의 레이블을 못맞추도록 feature extractor를 학습해야함
  - 즉, 일반적인 cross entropy loss를 최소화하는 방식과는 다르게 cross entropy loss를 최대화해야함 (?)

DANN

- Classification loss
  - W,b,V,c 의 parameter로 이루어진 model이 있을 때, classification loss는 아래와 같음

$$\min_{\mathbf{W},\mathbf{b},\mathbf{V},\mathbf{c}} \left[\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_y^i(\mathbf{W},\mathbf{b},\mathbf{V},\mathbf{c}) + \lambda \cdot R(\mathbf{W},\mathbf{b}) \right],$$

- Domain-invariance loss (regularizer로 활용됨)

$$R(\mathbf{W},\mathbf{b}) = \max_{\mathbf{u},z} \left[-\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z) - \frac{1}{n'} \sum_{i=n+1}^{N} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z) \right],$$

- Complete optimization objective

$$E(\mathbf{W},\mathbf{V},\mathbf{b},\mathbf{c},\mathbf{u},z) \tag{9}$$
$$= \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_y^i(\mathbf{W},\mathbf{b},\mathbf{V},\mathbf{c}) - \lambda \left(\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z) + \frac{1}{n'} \sum_{i=n+1}^{N} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z) \right),$$

Training (adversarial)

$$E(\mathbf{W}, \mathbf{V}, \mathbf{b}, \mathbf{c}, \mathbf{u}, z) \tag{9}$$
$$= \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_y^i(\mathbf{W}, \mathbf{b}, \mathbf{V}, \mathbf{c}) - \lambda\left(\frac{1}{n}\sum_{i=1}^{n}\mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z) + \frac{1}{n'}\sum_{i=n+1}^{N}\mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z)\right),$$

(Adversarial training)
Classification loss <-> Domain invariance loss

$$(\hat{\mathbf{W}}, \hat{\mathbf{V}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}) = \underset{\mathbf{W},\mathbf{V},\mathbf{b},\mathbf{c}}{\operatorname{argmin}} E(\mathbf{W}, \mathbf{V}, \mathbf{b}, \mathbf{c}, \hat{\mathbf{u}}, \hat{z})$$

Classification loss 최소화

$$(\hat{\mathbf{u}}, \hat{z}) = \underset{\mathbf{u},z}{\operatorname{argmax}} E(\hat{\mathbf{W}}, \hat{\mathbf{V}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}, \mathbf{u}, z)$$

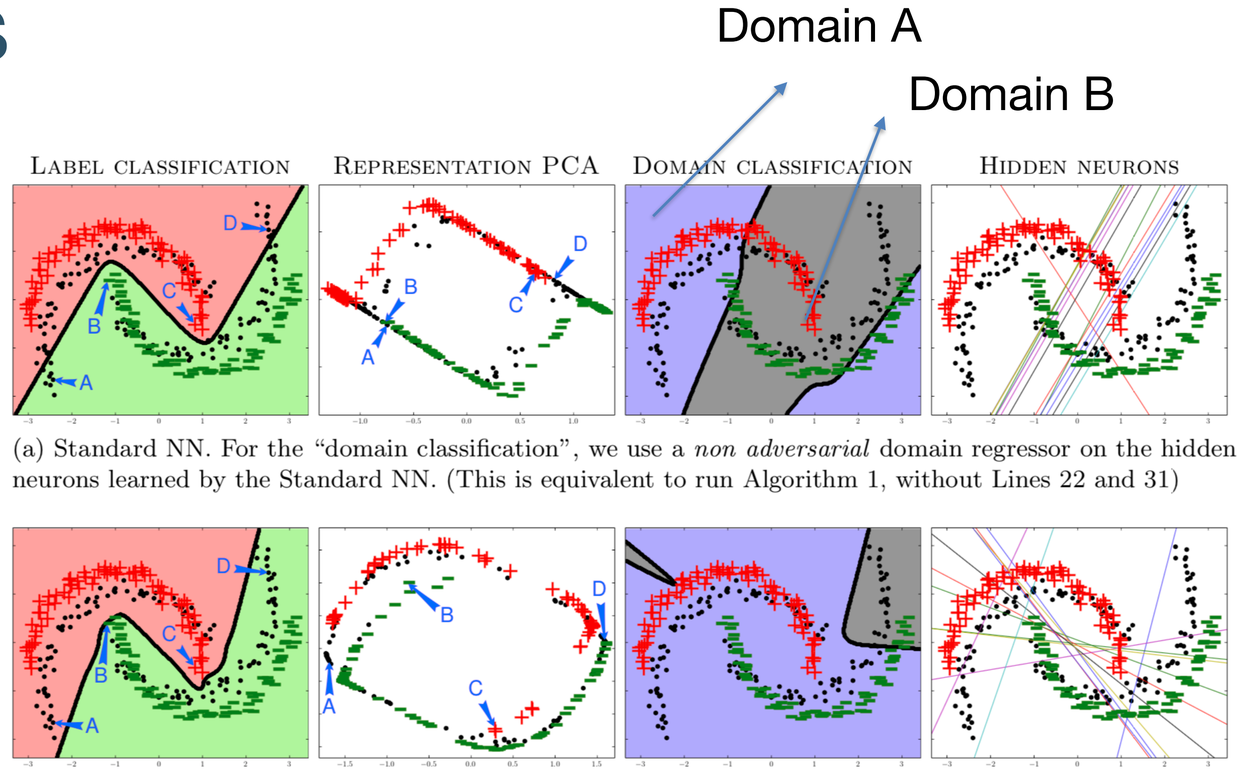
Domain invariance 최대화

Domain invariance 최대화 방법 -> cross entropy loss에 -lambda 값을 곱하여 backpropagate

Experiments

w/o domain adaptation

w/ domain adaptation



(a) Standard NN. For the "domain classification", we use a *non adversarial* domain regressor on the hidden neurons learned by the Standard NN. (This is equivalent to run Algorithm 1, without Lines 22 and 31)

(b) DANN (Algorithm 1)

Figure 2: The *inter-twinning moons* toy problem. Examples from the source sample are represented as a "+" (label 1) and a "−" (label 0), while examples from the unlabeled target sample are represented as black dots. See text for the figure discussion.

Experiments

Figure 6: Examples of domain pairs used in the experiments. See Section 5.2.4 for details.

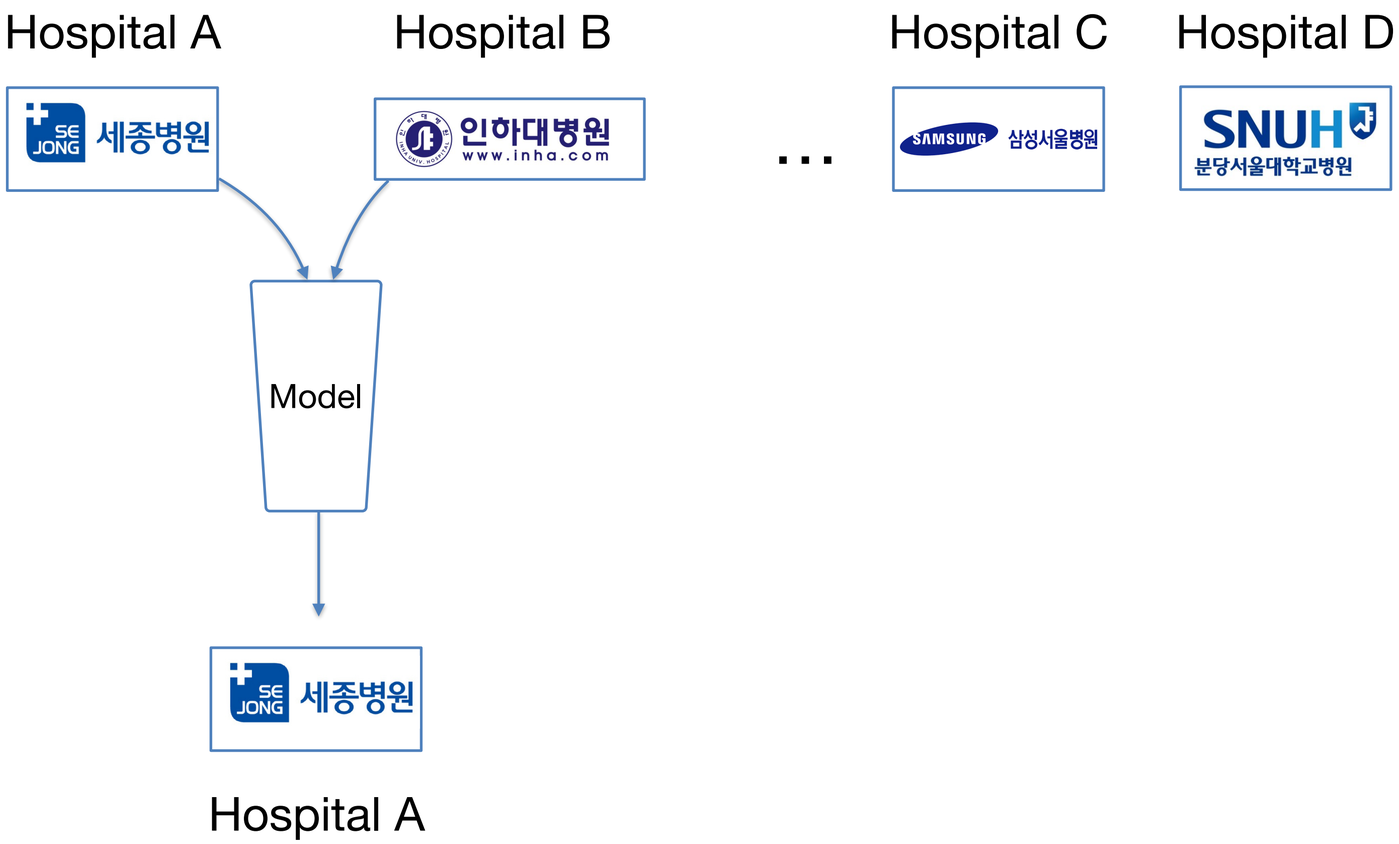
Method	Source	MNIST	Syn Numbers	SVHN	Syn Signs
	Target	MNIST-M	SVHN	MNIST	GTSRB
Source only		.5225	.8674	.5490	.7900
SA (Fernando et al., 2013)		.5690 (4.1%)	.8644 (−5.5%)	.5932 (9.9%)	.8165 (12.7%)
DANN		**.7666** (52.9%)	**.9109** (79.7%)	**.7385** (42.6%)	**.8865** (46.4%)
Train on target		.9596	.9220	.9942	.9980

Method	Source	Amazon	DSLR	Webcam
	Target	Webcam	Webcam	DSLR
GFK(PLS, PCA) (Gong et al., 2012)		.197	.497	.6631
SA* (Fernando et al., 2013)		.450	.648	.699
DLID (Chopra et al., 2013)		.519	.782	.899
DDC (Tzeng et al., 2014)		.618	.950	.985
DAN (Long and Wang, 2015)		.685	.960	.990
Source only		.642	.961	.978
DANN		**.730**	**.964**	**.992**

Table 3: Accuracy evaluation of different DA approaches on the standard Office (Saenko et al., 2010) data set. All methods (except SA) are evaluated in the "fully-transductive" protocol (some results are reproduced from Long and Wang, 2015). Our method (last row) outperforms competitors setting the new state-of-the-art.

Domain Adaptation 적용 방안 (DEWS)

"Unified Deep Supervised Domain Adaptation and Generalization"



Putting the world’s medical data to work

hello@vuno.co