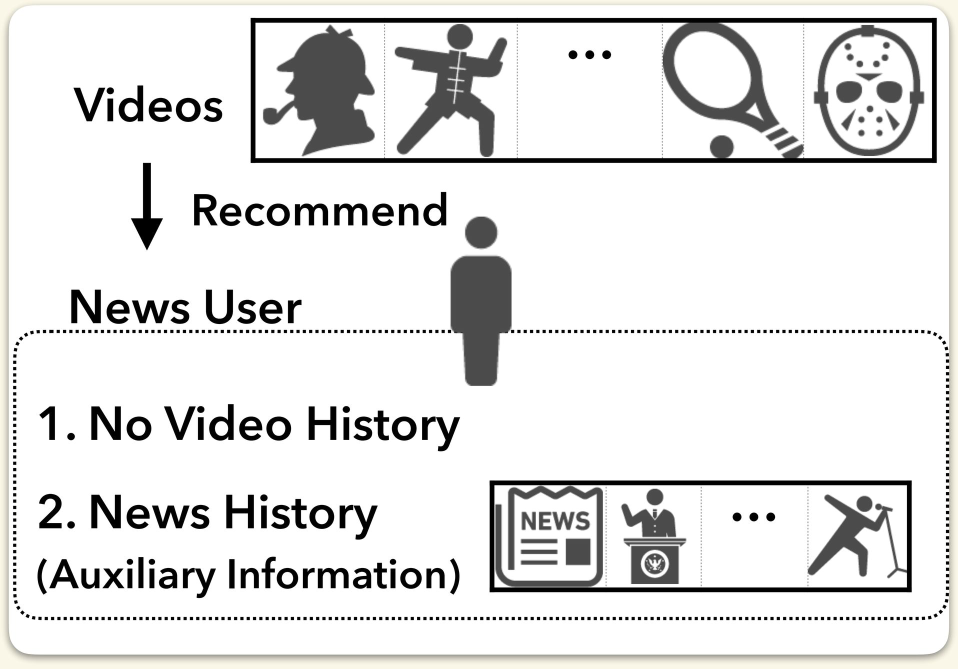
Cross-Domain Recommendation via Deep Domain Adaptation

Heishiro Kanagawa[1*] Hayato Kobayashi[2,4]
Nobuyuki Shimizu[2] Yukihiro Tagami[2] Taiji Suzuki[3,4]
[1] Gatsby Unit, UCL, [2] Yahoo Japan Corporation
[3] University of Tokyo, [4] RIKEN AIP
*Work performed during the author's internship at Yahoo! JAPAN

1. Motivation: Recommending Videos to Users in News Service

Task:

- Given: Video & News services
- Goal: design a Recommender System (RS) that suggests **videos** to users who
  - A. Have **never** used **Video** service before
  - B. But used **News** service
- Constraint:
  - **Few/No users shared across services**



Use case:

- News = popular & having a large user base
- Video = less known (e.g. relatively new service)

→ Making quality recommendations attract new users in News service who have never used Video service

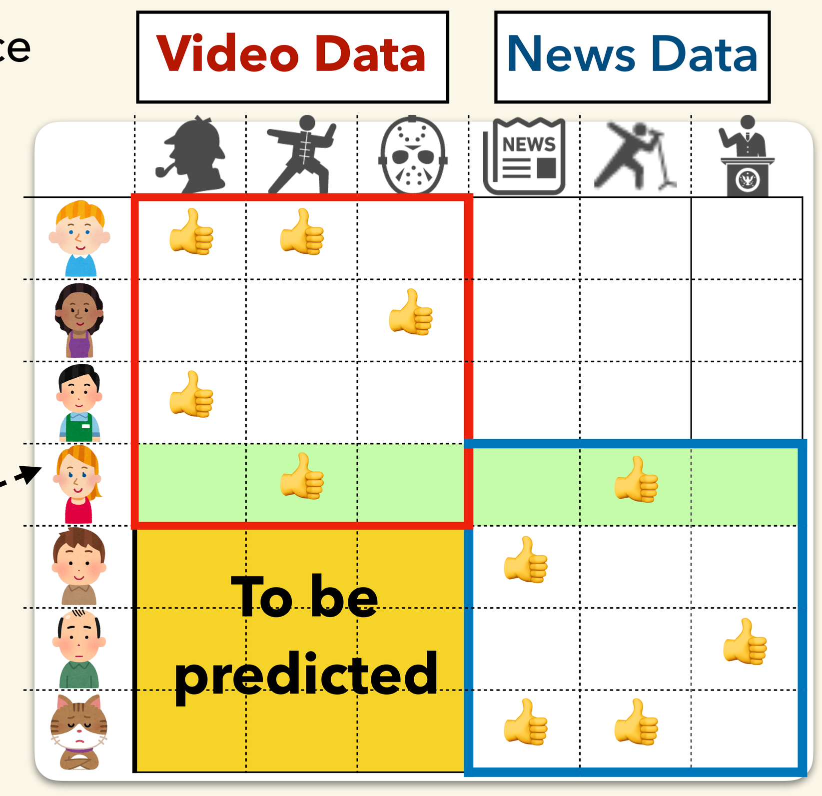
Challenge: Conventional RSs don't work

- Learning from Video users

→ Optimised for video users (input = video histories)
→ News users don't have video histories

- Learning from News users ≠ possible (no labels)
- Learning from **shared** users ≠ feasible

→ There are few/no shared users
→ Not enough training examples



Few Shared Users

Q. How should we utilise knowledge of Video service and transfer it to News users?

→ Our Approach:
Adapt Video RS to News users with **extreme classification + domain adaptation**
→ Contribution:
Propose a method that works with commonly available forms of content information

2. Problem Formulation

Recommendation as Extreme Classification:

Given:

- Video data (source) $D_S = \{(x_i^S, y_i^S)\}_{i=1}^{N_S} \overset{\text{i.i.d.}}{\sim} P_S(X, Y)$
- News data (target) $D_T = \{x_i^T\}_{i=1}^{N_T} \overset{\text{i.i.d}}{\sim} P_T(X)$

$x_i^S \in \mathbb{R}^d$: vector representing video history, $x_i^T \in \mathbb{R}^d$: vector representing news history
$y_i \in \{1, \ldots, K\}$: video label

Goal = Construct a classifier $\eta : x^T \mapsto y \in \{1, \ldots, K\}$
(predict a video corresponding to a news user)
with a low expected error: $\Pr_{(X,Y)\sim P_T(X,Y)}[\eta(X) \neq Y]$

Note: Data domains are distinct $P_S(X, Y) \neq P_T(X, Y)$
→ Supervised ML + Training on D_S won't work (error ≠ low)
→ Correction via domain adaptation

3. Unsupervised Domain Adaptation

We use Domain Separation Network (DSN) [1]

DSN achieves domain adaptation with:

- Shared encoder: extract predictive features shared across domains
- Private encoder: extract features private to each data domain
- Shared decoder: reconstruct from private + shared features
- Objective: $L_{\text{DSN}} = L_{\text{class}} + \alpha L_{\text{reconst}} + \beta L_{\text{diff}} + \gamma L_{\text{sim}}$

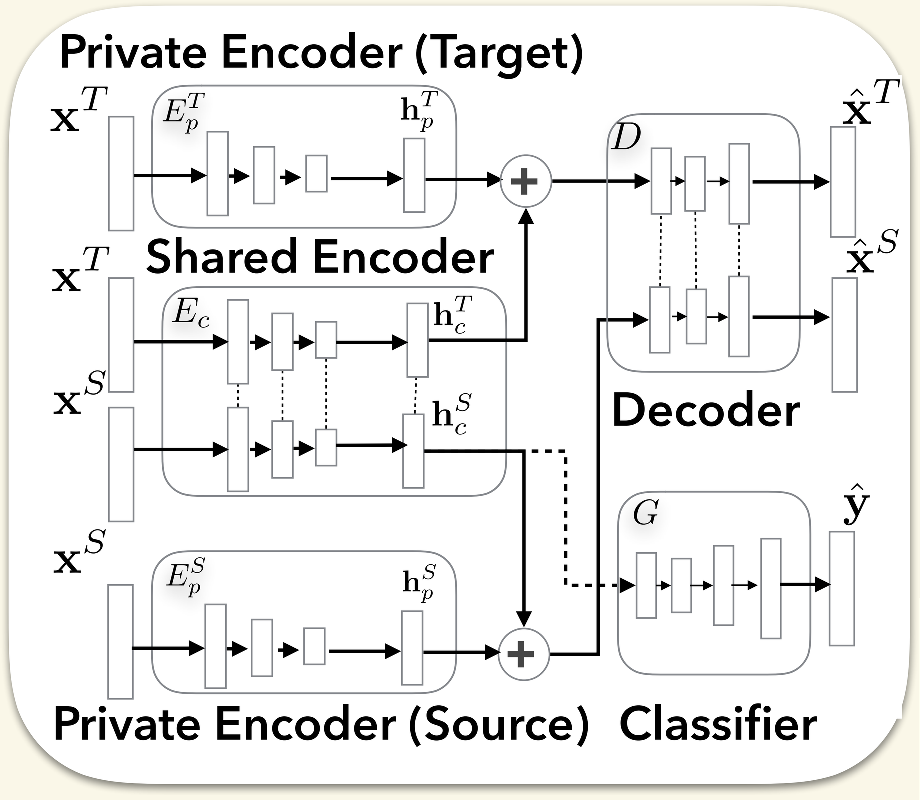


Fig: Architecture of DSN [1]

$$L_{\text{class}} = -\sum_{i=1}^{N_S} \mathbf{y}_i \cdot \log \hat{\mathbf{y}}_i$$

$$L_{\text{reconst}} = \sum_{u\in\{S,T\}} \sum_{i=1}^{N_u} \| \mathbf{x}_i^u - \hat{\mathbf{x}}_i^u \|^2$$

$$L_{\text{diff}} = \sum_{u\in\{S,T\}} \sum_{i,j=1}^{N_u} \underbrace{|\mathbf{h}_{c,i}^u \cdot \mathbf{h}_{p,j}^u|^2}_{\text{orthogonality}}$$

$$L_{\text{sim}} = -\sum_i \mathbf{d}_i \cdot \log \hat{\mathbf{d}}_i$$

L_{sim}: domain adversarial loss
- $\mathbf{d}_i$: Domain label
- $\hat{\mathbf{d}}_i$: Domain classifier output (input = $\mathbf{h_c}$)

→ Maximise w.r.t. encoder
Minimise w.r.t. classifier

4. Experiment

Data = Video/News services of Yahoo! JAPAN

- 2 week-long browsing logs
- Training data = of size 11m for each service
  - No shared users in this data
- Validation data (33k) & Test data (38k):
  - Constructed from logs of shared users
  - Instance = (news history, video label) pair

Items have textual attributes:

- Video: title, cast, category, short description
- News: title, category

Note: only news articles in **entertainment** categories were used

Data representation:

- User history = bag of Items
  - Treat as a document composed of item's textual attributes
- Represent history with TF-IDF:
  - For each domain, form a vocabulary set according to TF-IDF value (computed from histories)
  - Combining two vocabulary sets
    → common vocabulary set of size 50k
    → Input dimension $d = 50{,}000$

Construct a DSN:

- Fully-connected layers (hidden layers):
  - Encoder: (256–128–128– 64)
  - Decoder: (128–128–256)
  - Classifier: (256–256–256–64)
- ADAM optimiser with initial learning rate 10^{-3}
- Hyperparemeters of the objective L_{DSN}

$$\alpha = 10^{-3}, \beta = 10^{-2}, \gamma = 10^{2}$$

Evaluation Metric = DCG (ranking quality measure)

$$\text{DCG}@M = \frac{1}{\log(m+1)} \sum_{m=1}^{M} I[\hat{y}_m = y]$$

Compare with baseline Models:

- NN:
  - Same neural network **trained only on** Video data
  - Compared to investigate the effectiveness of domain adaptation
  - Considered as strong single-domain content-based method
- Cross-domain Matrix Factorisation (CdMF) [2]:
  - SOTA Cross-domain Bayesian matrix factorisation
  - Trained on binary matrices
  - Do not use content information
- POP: suggest items in descending popularity order
  - Non-personalised method
  - Compared to see personalisation performance

Result: Performance Comparison in DCG

- 80% of training/validation/test was subsampled → 1 trial
- Table entry = Mean DCG ± Std (across 5 trials)
- DSN (DCG/CEL) = chosen by DCG/Cross Entropy Loss on validation data

Method	DCG@1	DCG@50	DCG@100
DSN (DCG)	**0.062 ± 0.021**	**0.287 ± 0.015**	**0.295 ± 0.015**
DSN (CEL)	0.041 ± 0.021	0.258 ± 0.023	0.266 ± 0.023
NN (DCG)	0.042 ± 0.021	0.274 ± 0.010	0.280 ± 0.011
NN (CEL)	0.028 ± 0.030	0.247 ± 0.025	0.256 ± 0.024
CdMF	0.001 ± 0.000	0.014 ± 0.000	0.064 ± 0.000
POP	0.040 ± 0.001	0.279 ± 0.002	0.287 ± 0.001

- DSN (DCG) = best performance & DSN (DCG) > NN (DCG)
- NN/DSN (CEL) underperformed POP
- CdMF: worst performance → explicit ratings required; our datasets are binary matrices → CdMF could not process implicit feedback properly

5. Discussions and Future Work

Discussion: Poor Performance of NN/DSN (CEL)

- Worse than POP, does not capture popularity
- Top-1 item prediction is too hard
- CEL does not give useful signal
- DCG better captures quality of predictions (given in the form of probability distribution)

Future Work:

- Replacing the training loss with a ranking loss (e.g. DCG)
- Combining item side information (info unique to RSs) using zero-/few-shot learning techniques
  → ease the difficulty of extreme classification

References:

1. Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D. & Erhan, D. (2016). Domain Separation Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon & R. Garnett (ed.),*Advances in Neural Information Processing Systems 29* (pp. 343-351).
2. Iwata, T. & Koh, T.. (2015). Cross-domain recommendation without shared users or items by sharing latent vector distributions. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, in PMLR* 38:379-387