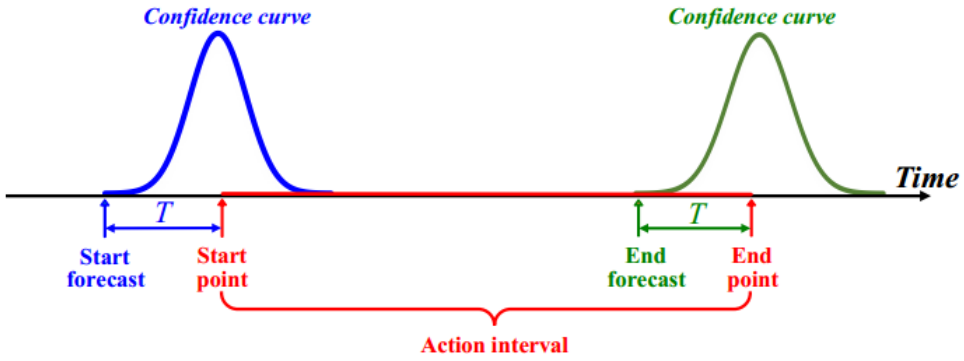
Vehicle Interaction Learning

Xiaosong Jia

08/06/2019

Additional Prior Information

- Starting < Ending
- Naïve Algorithm: Confidence > threshold and Farthest Local Minima -> Changing_Lane_Time Acc > 90%
- Other Trying:



1. σ = 5

```
[0.056 0.089 0.135 0.198 0.278 0.375 0.487 0.607 0.726 0.835 0.923 0.98
 1.    0.98  0.923 0.835 0.726 0.607 0.487 0.375 0.278 0.198 0.135 0.089
 0.056]
```

-> σ = 1.2 worse

```
[1.929e-22 5.670e-19 8.324e-16 6.102e-13 2.234e-10 4.083e-08 3.727e-06
 1.699e-04 3.866e-03 4.394e-02 2.494e-01 7.066e-01 1.000e+00 7.066e-01
 2.494e-01 4.394e-02 3.866e-03 1.699e-04 3.727e-06 4.083e-08 2.234e-10
 6.102e-13 8.324e-16 5.670e-19 1.929e-22]
```

Additional Prior Information

- Other trying 2:
- Inverse Order Loss:

Start_Confidence = [0.3, 0.6, 0.9, 0.3]

End_Confidence = [0.7, 0.3, 1.0, 0.2]

Start_Max_After = [0.9, 0.9, 0.9, 0.3]

End_Max_Before = [0.7, 0.7, 1.0, 1.0]

Results:

1. 6X slower
2. Inverse Order Loss ~0 after average among batch and length

Inver Order Loss = max(0, Start_Confidence+ End_Max_Before -1) + max(0, End_Confidence + Start_Max_After -1)

Additional Prior Information

- Other possible way?

1. Tune σ

2. Proposal Confidence Scorer -> Train by IoU

3. Loss Function

4. Model Structure

About IoU

- IoU = $\frac{A \cap B}{A \cup B}$
- As for 15 frames (fps=5):

IoU = 0.6 -> 25 frames max-len or 3 frames offset (12/18 = 0.667)

IoU = 0.7 -> 21 frames max-len or 2 frames offset (13/17 = 0.765)

IoU = 0.8 -> 18 frames max-len or 1 frames offset (14/16=0.875)

IoU = 0.9 -> 16 frames max-len or 0 frames offset

Best Results

- Train on NGSIM(50s):

Validation Set: Best_IoU6_Acc 0.848 Best_IoU7_Acc 0.736 Best_IoU8_Acc 0.535 Best_IoU9_Acc 0.201, Best_Change_Lane_Acc 0.972, Best_Traj_Cls_Acc 0.895

Transfer on HighD(50s):

Exchange xy: IOU6 Acc 0.741, IOU7 Acc 0.576, IOU8 Acc 0.316, IOU9 Acc 0.087, Change_Lane_Acc 0.919, Traj_Cls_Acc 0.762

no_exchange xy:IOU6 Acc 0.630, IOU7 Acc 0.457, IOU8 Acc 0.244, IOU9 Acc 0.065, Change_Lane_Acc 0.870, Traj_Cls_Acc 0.750

Best Results

- Train on HighD(50s):

Validation Set: Best_IoU6_Acc 0.966 Best_IoU7_Acc 0.899 Best_IoU8_Acc 0.711 Best_IoU9_Acc 0.355, Best_Change_Lane_Acc 0.993, Best_Traj_Cls_Acc 0.994

Transfer on NGSIM (10s):

Exchange xy: IOU6 Acc 0.635, IOU7 Acc 0.489, IOU8 Acc 0.304, IOU9 Acc 0.085, Change_Lane_Acc 0.873, Traj_Cls_Acc 0.842

no_exchange xy:IOU6 Acc 0.647, IOU7 Acc 0.488, IOU8 Acc 0.268, IOU9 Acc 0.075, Change_Lane_Acc 0.865, Traj_Cls_Acc 0.792

Some Visualization

- Train on NGSIM:

Training Set: Samples:5339, False Positive:414, False Negative:14

Validation Set: Samples:1310, False Positive:129, False Negative:27

Transfer on HighD: Samples:12514, False Positive:2958 False Negative:26

About Trajectory Max Length

Train Set	Length	Validation Res			Transfer (exchange xy)			Transfer (not exchange xy)		
		IoU 6 Acc	Change Lane Acc	Traj Cls Acc	IoU 6 Acc	Change Lane Acc	Traj Cls Acc	IoU 6 Acc	Change Lane Acc	Traj Cls Acc
NGSIM	10s	91.7%	98.5%	90.8%	57.0%	85.7%	86.3%	57.0%	78.7%	**89.0%**
NGSIM	20s	86.1%	97.8%	90.5%	52.1%	90.9%	80.4%	65.9%	90.3%	82.1%
NGSIM	50s	84.8%	97.2%	89.5%	**74.1%**	91.9%	76.2%	63.0%	87.0%	75.0%
HighD	10s	96.3%	99.3%	99.3%	60.8%	90.3%	80.8%	64.6%	93.1%	78.7%
HighD	20s	97.0%	99.3%	99.4%	68.4%	89.2%	80.3%	57.1%	86.9%	75.8%
HighD	50s	96.6%	99.3%	99.4%	36.4%	75.3%	70.0%	37.1%	69.7%	75.2%

About Scaling

- Scale locally by Trajectory (max) then scale globally by all frames(mean std) -> validation: worse, transfer: IoU (slightly better), traj_cls (worse)
- Scale locally by Trajectory (max) -> worse
- Scale globally by maximum of all values then rotation -> not converge (deeper model does not help as well)
- Rotation then scale(mean-std) by batch -> not converge (deeper model does not help as well)
- Use same scale factor for car1 and car2 (similar results)

NGSIM vs HighD (max_norm)

	NGSIM	HighD
X1_mean	-4.37e-02	-1.62e-02
Y1_mean	4.74e-01	4.12e-02
X1_scale_mean	1.83e+01	3.37e+02
Y1_scale_mean	1.41e+03	4.24e+00
X2_mean	-2.25e-01	-2.10e-02
Y2_mean	4.22e-01	-1.30e-02
X2_scale_mean	1.83e+01	3.37e+02
Y2_scale_mean	1..41e+03	4.24e+00

Data Augmentation

- xy augmentation: (x1, y1, x2, y2) -> (y1, x1, y2, x2)
- vehicle augmentation: (x1, y1, x2, y2) -> (x2, y2, x1, y1)
- Random rotation

4x samples + randomness -> Nearly No Overfitting (Maybe we could use deeper model)

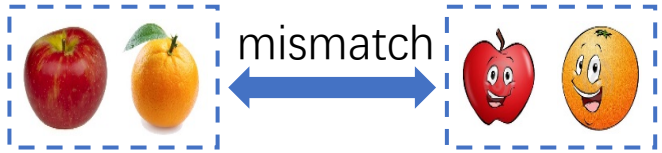
- Slow (50s data) -> more than 25 hour to converge
- Best Performance on Validation Set

NGSIM: Best_IoU6_Acc 0.961 Best_IoU7_Acc 0.894 Best_IoU8_Acc 0.660 Best_IoU9_Acc 0.215, Best_Change_Lane_Acc 0.996, Best_Traj_Cls_Acc 0.905

- Worse transferring (HighD 50s) but similar results on transferring (HighD 10s)

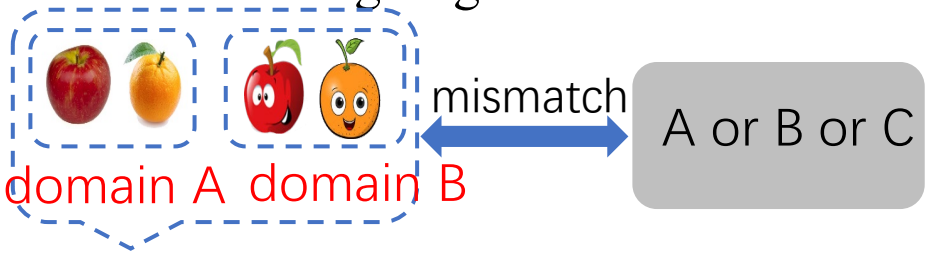
About Final Task

- Domain adaptation: unlabeled target domain is seen in the training stage.

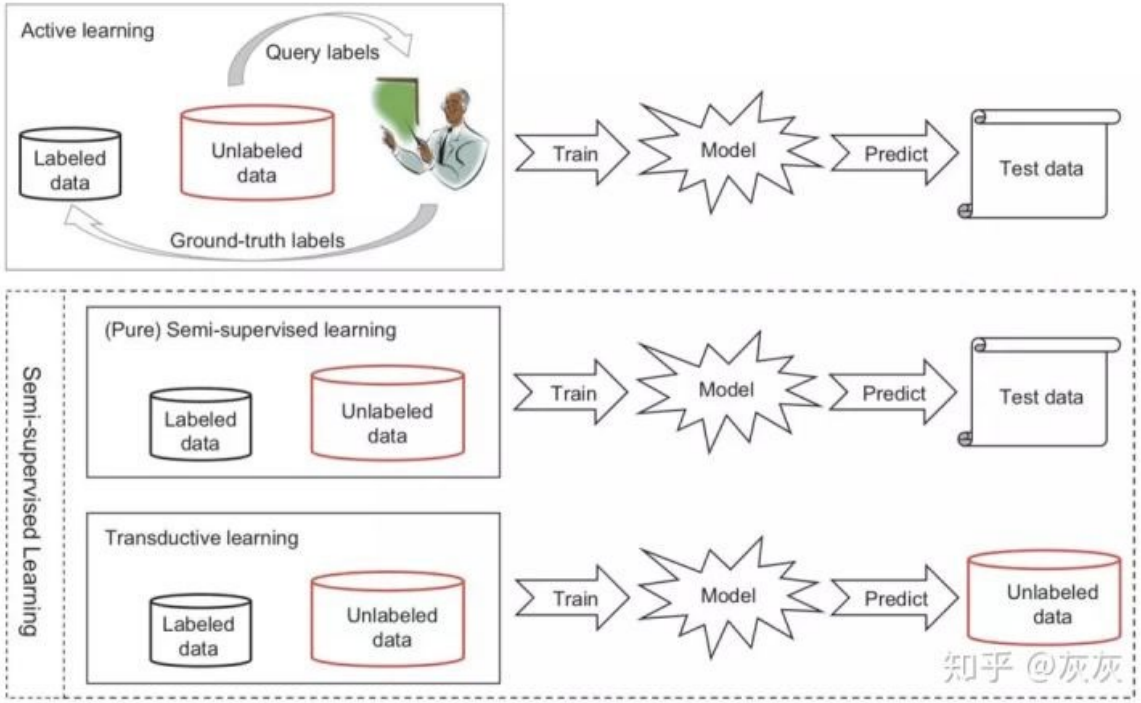


source domain target domain

- Domain generalization: unlabeled target domain is unseen in the training stage.



multiple latent domains target domain

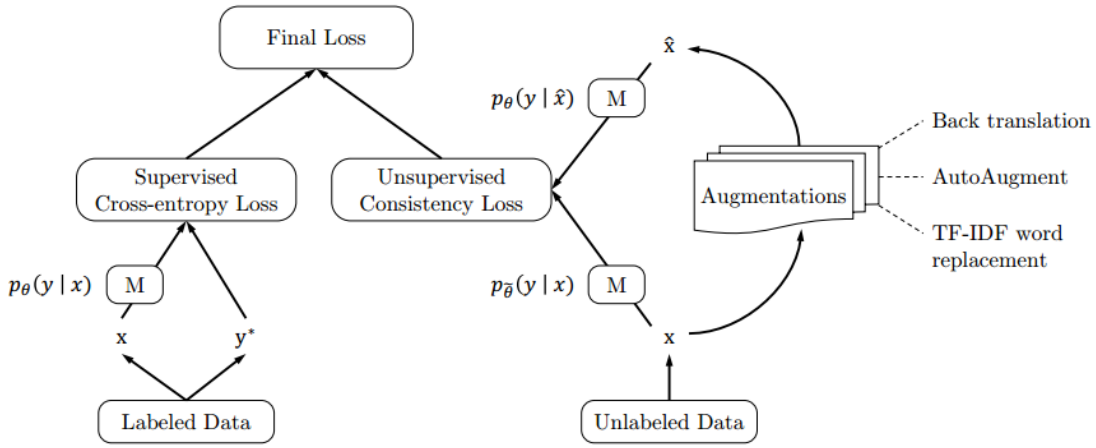


About Semi-supervised learning

- UDA (2019-07 Google, CMU)

$$\min_{\theta} \mathcal{J} = \mathop{\mathbb{E}}_{x, y^* \in L} \left[p_\theta(y^* \mid x) \right] + \lambda \mathcal{J}_{\text{UDA}}(\theta)$$

$$\min_{\theta} \mathcal{J}_{\text{UDA}}(\theta) = \mathop{\mathbb{E}}_{x \in U} \mathop{\mathbb{E}}_{\hat{x} \sim q(\hat{x}|x)} \left[\mathcal{D}_{\text{KL}} \left(p_{\tilde{\theta}}(y \mid x) \,\|\, p_\theta(y \mid \hat{x}) \right) \right]$$



Fully supervised baseline						
Datasets (# Sup examples)		IMDb (25k)	Yelp-2 (560k)	Yelp-5 (650k)	Amazon-2 (3.6m)	Amazon-5 (3m)	DBpedia (560k)
Pre-BERT SOTA		*4.32*	2.16	29.98	3.32	34.81	0.70
$\text{BERT}_{\text{LARGE}}$		4.51	*1.89*	*29.32*	*2.63*	*34.17*	*0.64*
Semi-supervised setting							
Initialization	**UDA**	IMDb (20)	Yelp-2 (20)	Yelp-5 (2.5k)	Amazon-2 (20)	Amazon-5 (2.5k)	DBpedia (140)
Random	✗	43.27	40.25	50.80	45.39	55.70	41.14
	✓	25.23	8.33	41.35	16.16	44.19	7.24
$\text{BERT}_{\text{BASE}}$	✗	27.56	13.60	41.00	26.75	44.09	2.58
	✓	5.45	2.61	33.80	3.96	38.40	1.33
$\text{BERT}_{\text{LARGE}}$	✗	11.72	10.55	38.90	15.54	42.30	1.68
	✓	4.78	2.50	33.54	3.93	37.80	1.09
$\text{BERT}_{\text{FINETUNE}}$	✗	6.50	2.94	32.39	12.17	37.32	-
	✓	**4.20**	**2.05**	**32.08**	**3.50**	**37.12**	-