Rethinking Distributional Matching Based Domain Adaptation

(2020)

Bo Li et al.

Summary

Contributions

In this paper the authors start by addressing some problems with popular domain adaptation (DA) works. Mainly, taking into account realistic domain shifts, the main assumptions such methods are based on fail. The authors specify three scenarios:

- **Label Distribution Shift (LDS).** When the target and source distributions are matched in the representation space, the label distributions on target and source will likely be disjoint, resulting in negative transfer and unbounded target error. For example, if the source domain has 50% of each class, A and B, while the target has 30 - 70%, if we align the distributions in the representation space, and with 100% accuracy on source, we can only obtain 80% accuracy on target. See figure bellow left.

- **Intermediate Layer Distribution Shift (ILDS).** Another line of work align the representation conditioned on the outputs to help overcome LDS and the negative transfer, where the representations are aligned per class or mode. However, in this case where the intra-class features are difference across domains. For example, if a given class with two subclasses, and the portion of these subclasses is different in source and target, we will have in a misalignment of the representations even if the main class label distribution is the same across domains. See figure bellow right.

- **Target with Outliers (TwO).** Recent methods try to relax the the distribution alignment by not requiring the latent-space distributions to match exactly, but only up to a given degree to give good results. However, we might need relax such requirement too much to get the correct matching, resulting in a poor discriminative target representations, additionally, there might be some samples from target lying outside the source support, and in this case the target risk becomes vacuous.

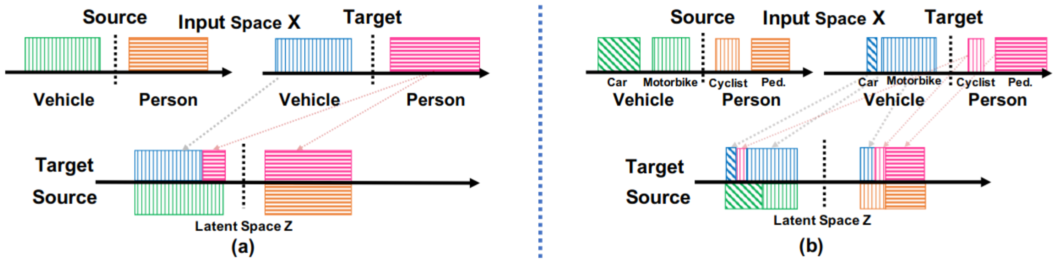


Figure 1: Failure cases of DM methods RDS. Colors denote categories, fill patterns denote sub-types, ped. denotes *pedestrians*.

To solve such issues, the authors propose a new method called InstaPBM consisting of: (1) Mutual Information Predictive Behavior Matching; (2) Contrastive Predictive Behavior Matching; (3) Mix-up

Predictive Behavior Matching. And test it on newly proposed benchmarks by the authors to recreate the three scenarios discussed above.

Method

Notations:

- $\mathcal{X}$, $\mathcal{Z}$ and $\mathcal{Y}$ denote the input, representation and output spaces respectively.
- Feature extractor $g : \mathcal{X} \mapsto \mathcal{Z}$ parametrized by ϕ.
- Classifier $h : \mathcal{Z} \mapsto \mathcal{Y}$ parametrized by ψ.
- The whole model as a composite of the feature extractor and the classifier: $f = g \circ h : \mathcal{X} \mapsto \mathcal{Y}$ parametrized by θ.
- We sample pairs of (input, label) from source S: $(x, y) \sim \mathcal{P}_S(X, Y)$, and only the inputs from target T: $x \sim \mathcal{P}_T(X)$.
- Distributional matching (DM) learns a shared representation space with g so that the representations across domain are aligned: $\mathcal{P}_u(Y \mid X) = \mathcal{P}_u(Y \mid Z = g(X)), u \in \{S, T\}$.

Instance-based Predictive Behavior Matching The proposed loss function consists of four main components:

1- Mutual Information Predictive Behavior Matching. In the normal supervised case, as the training progress, the model learns a correlation between the inputs and the outputs, resulting in an increase of the mutual information (MI) between the outputs and the inputs $I(X;Y)$. In the target domain, with no labels, the authors propose to maximize the equivalent formulation of the MI: $I_T(X;Y) = H_T(Y) - H_T(Y \mid X)$. And by replacing each entropy term by its definition, we obtain the MI based loss:

$$\mathcal{L}_M(T;\theta) = \mathbb{E}_{y \sim \mathcal{P}_\theta(Y)}\left[\log p_\theta(y)\right] - \mathbb{E}_{x \sim \mathcal{P}_T(X)}\left[\sum_y p_\theta(y \mid x) \log p_\theta(y \mid x)\right]$$

To approximate the label distribution $p_\theta(y)$, the authors propose to use a moving average $q(y)$ of $p_\theta(y \mid x), x \sim \mathcal{P}_T(x)$ as an approximation of $p_\theta(y)$.

2- Contrastive Predictive Behavior Matching. The second term enforces a consistency over a set of semantic preserving transformations $\mathcal{T}_{sp}$, where the model is trained to output similar labels of a given image x and its augmented version $t(x)$, while predicting different labels for two different examples x and x'. This can be achieved by minimizing the KL divergence between the outputs of x and $t(x)$ and maximizing for x and x'. The maximization is done over both target and source, while the maximization is only done over source to be able to sample instance x' with different classes that x. The loss is formulated as follows:

$$\begin{aligned}\mathcal{L}_C(S,T;\theta) = {} & \mathbb{E}_{x \sim \mathcal{P}_{S,T}, t \sim \mathcal{T}}\left[D_{KL}\left(p_\theta(y \mid x) \| p_\theta(y \mid t(x))\right)\right] \\ & - \lambda_{\text{con}} \mathbb{E}_{(x,y) \sim \mathcal{P}_S, (x',y') \sim \mathcal{P}_S, y \neq y'}\left[D_{KL}\left(p_\theta(y \mid x) \| p_\theta\left(y' \mid x'\right)\right)\right]\end{aligned}$$

3- Mix-up Predictive Behavior Matching. The thirds term uses a mixup operation: $t\left(x, x', \beta\right) = \beta x + (1-\beta)x' \sim \mathcal{T}_\beta, \beta \in [0,1]$ that creates an interpolation between two inputs to encourages the model to behave similarly on linear interpolations of samples in both domains. In this case, we sample a source sample x and then a target sample x', the model is then trained to have similar prediction, between the outputs of the mixed input $t(x, x')$, and the mixed outputs $y_{si} = \beta y + (1-\beta)y'$:

$$\mathcal{L}_U(S,T;\phi,\psi) = \mathbb{E}_{(x,x') \sim \mathcal{P}_{S,T}, t \sim \mathcal{T}_{si}}\left[D_{KL}\left(p_\theta\left(y \mid t\left(x, x', \beta\right)\right) \| q\left(y_{si} \mid x, x'\right)\right)\right]$$

4- Self-Supervision. The last term added to the loss function consists of self-supervised loss, where the model is trained to prediction the transformation that was applied to the input, the auhors choose to use Rotation Prediction,Vertical Flip Prediction and Patch Location Prediction. This loss is:

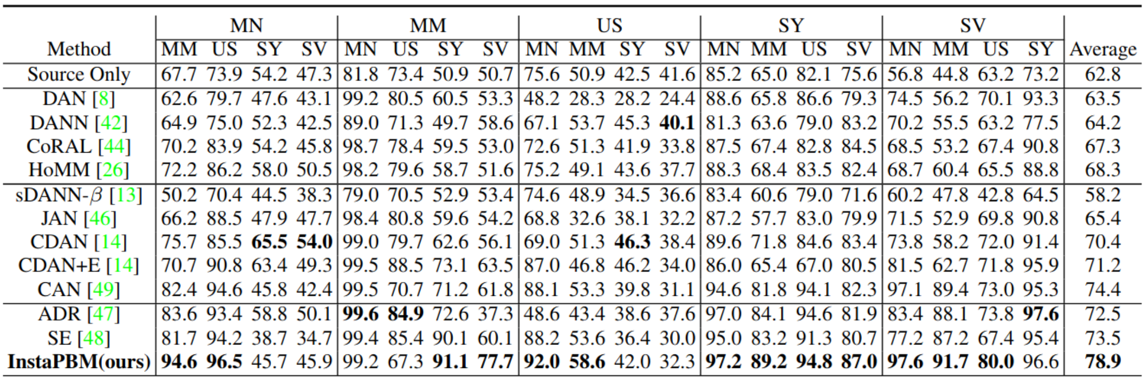
$$\mathcal{L}_S(S,T;\phi,\omega)=\mathbb{E}_{x\sim\mathcal{P}_{S,T},t\sim\mathcal{T}_s t}\left[L_t\left(h_t\circ g\left(t\left(x_{S,T}\right)\right),\tilde{y}_t\right)\right]$$

Total Loss:

$$\min_{\theta,\phi,\omega}\lambda_M\mathcal{L}_M(S,T;\theta)+\lambda_C\mathcal{L}_C(S,T;\theta)+\lambda_U\mathcal{L}_U(S,T;\phi,\psi)+\lambda_S\mathcal{L}_S(S,T;\phi,\omega)$$

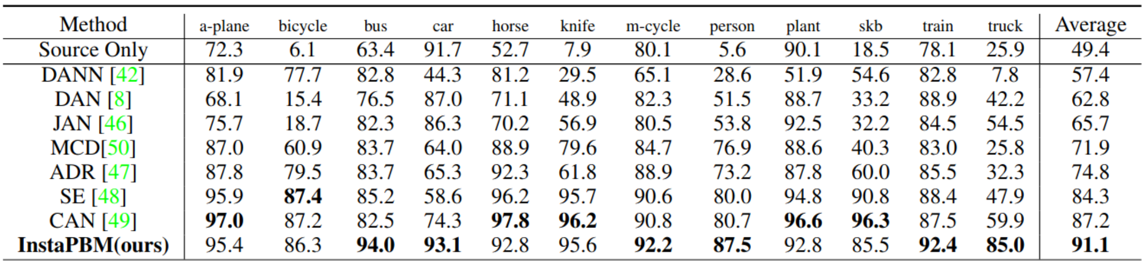
Results

Table 1: Performance comparison on **Digits5** dataset. Best accuracy in bold.



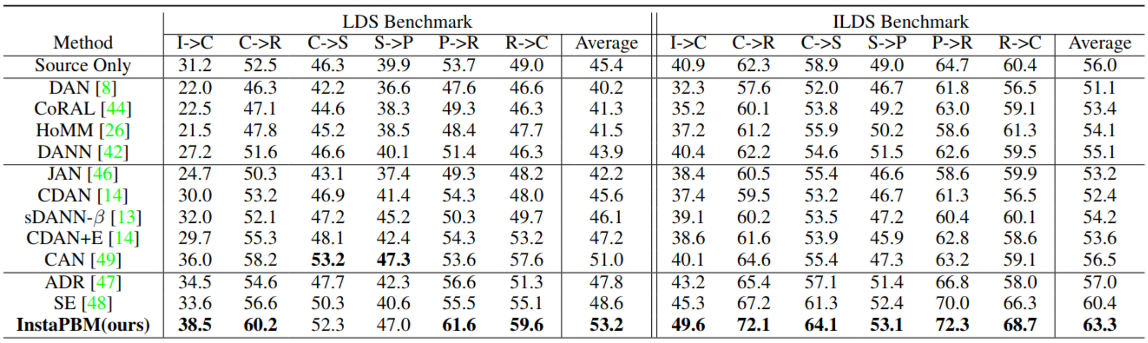
Method	MN				MM				US				SY				SV				Average
	MM	US	SY	SV	MN	US	SY	SV	MN	MM	SY	SV	MN	MM	US	SV	MN	MM	US	SY	
Source Only	67.7	73.9	54.2	47.3	81.8	73.4	50.9	50.7	75.6	50.9	42.5	41.6	85.2	65.0	82.1	75.6	56.8	44.8	63.2	73.2	62.8
DAN [8]	62.6	79.7	47.6	43.1	99.2	80.5	60.5	53.3	48.2	28.3	28.2	24.4	88.6	65.8	86.6	79.3	74.5	56.2	70.1	93.3	63.5
DANN [42]	64.9	75.0	52.3	42.5	89.0	71.3	49.7	58.6	67.1	53.7	45.3	**40.1**	81.3	63.6	79.0	83.2	70.2	55.5	63.2	77.5	64.2
CoRAL [44]	70.2	83.9	54.2	45.8	98.7	78.4	59.5	53.0	72.6	51.3	41.9	33.8	87.5	67.4	82.8	84.5	68.5	53.2	67.4	90.8	67.3
HoMM [26]	72.2	86.2	58.0	50.5	98.2	79.6	58.7	51.6	75.2	49.1	43.6	37.7	88.3	68.4	83.5	82.4	68.7	60.4	65.5	88.8	68.3
sDANN-β [13]	50.2	70.4	44.5	38.3	79.0	70.5	52.9	53.4	74.6	48.9	34.5	36.6	83.4	60.6	79.0	71.6	60.2	47.8	42.8	64.5	58.2
JAN [46]	66.2	88.5	47.9	47.7	98.4	80.8	59.6	54.2	68.8	32.6	38.1	32.2	87.2	57.7	83.0	79.9	71.5	52.9	69.8	90.8	65.4
CDAN [14]	75.7	85.5	**65.5**	**54.0**	99.0	79.7	62.6	56.1	69.0	51.3	**46.3**	38.4	89.6	71.8	84.6	83.4	73.8	58.2	72.0	91.4	70.4
CDAN+E [14]	70.7	90.8	63.4	49.3	99.5	88.5	73.1	63.5	87.0	46.8	46.2	34.0	86.0	65.4	67.0	80.5	81.5	62.7	71.8	95.9	71.2
CAN [49]	82.4	94.6	45.8	42.4	99.5	70.7	71.2	61.8	88.1	53.3	39.8	31.1	94.6	81.8	94.1	82.3	97.1	89.4	73.0	95.3	74.4
ADR [47]	83.6	93.4	58.8	50.1	**99.6**	**84.9**	72.6	37.3	48.6	43.4	38.6	37.6	97.0	84.1	94.6	81.9	83.4	88.1	73.8	**97.6**	72.5
SE [48]	81.7	94.2	38.7	34.7	99.4	85.4	90.1	60.1	88.2	53.6	36.4	30.0	95.0	83.2	91.3	80.7	77.2	87.2	67.4	95.4	73.5
InstaPBM(ours)	**94.6**	**96.5**	45.7	45.9	99.2	67.3	**91.1**	**77.7**	**92.0**	**58.6**	42.0	32.3	**97.2**	**89.2**	**94.8**	**87.0**	**97.6**	**91.7**	**80.0**	96.6	**78.9**

Table 2: Performance comparison on **VisDA2017** dataset. Best accuracy in bold.



Method	a-plane	bicycle	bus	car	horse	knife	m-cycle	person	plant	skb	train	truck	Average
Source Only	72.3	6.1	63.4	91.7	52.7	7.9	80.1	5.6	90.1	18.5	78.1	25.9	49.4
DANN [42]	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN [8]	68.1	15.4	76.5	87.0	71.1	48.9	82.3	51.5	88.7	33.2	88.9	42.2	62.8
JAN [46]	75.7	18.7	82.3	86.3	70.2	56.9	80.5	53.8	92.5	32.2	84.5	54.5	65.7
MCD[50]	87.0	60.9	83.7	64.0	88.9	79.6	84.7	76.9	88.6	40.3	83.0	25.8	71.9
ADR [47]	87.8	79.5	83.7	65.3	92.3	61.8	88.9	73.2	87.8	60.0	85.5	32.3	74.8
SE [48]	95.9	**87.4**	85.2	58.6	96.2	95.7	90.6	80.0	94.8	90.8	88.4	47.9	84.3
CAN [49]	**97.0**	87.2	82.5	74.3	**97.8**	**96.2**	90.8	80.7	**96.6**	**96.3**	87.5	59.9	87.2
InstaPBM(ours)	95.4	86.3	**94.0**	**93.1**	92.8	95.6	**92.2**	**87.5**	92.8	85.5	**92.4**	**85.0**	**91.1**

Table 3: Performance comparison on **DomainNet-LDS** and **DomainNet-ILDS** Benchmarks.



Method	LDS Benchmark							ILDS Benchmark						
	I->C	C->R	C->S	S->P	P->R	R->C	Average	I->C	C->R	C->S	S->P	P->R	R->C	Average
Source Only	31.2	52.5	46.3	39.9	53.7	49.0	45.4	40.9	62.3	58.9	49.0	64.7	60.4	56.0
DAN [8]	22.0	46.3	42.2	36.6	47.6	46.6	40.2	32.3	57.6	52.0	46.7	61.8	56.5	51.1
CoRAL [44]	22.5	47.1	44.6	38.3	49.3	46.3	41.3	35.2	60.1	53.8	49.2	63.0	59.1	53.4
HoMM [26]	21.5	47.8	45.2	38.5	48.4	47.7	41.5	37.2	61.2	55.9	50.2	58.6	61.3	54.1
DANN [42]	27.2	51.6	46.6	40.1	51.4	46.3	43.9	40.4	62.2	54.6	51.5	62.6	59.5	55.1
JAN [46]	24.7	50.3	43.1	37.4	49.3	48.2	42.2	38.4	60.5	55.4	46.6	58.6	59.9	53.2
CDAN [14]	30.0	53.2	46.9	41.4	54.3	48.0	45.6	37.4	59.5	53.2	46.7	61.3	56.5	52.4
sDANN-β [13]	32.0	52.1	47.2	45.2	50.3	49.7	46.1	39.1	60.2	53.5	47.2	60.4	60.1	54.2
CDAN+E [14]	29.7	55.3	48.1	42.4	54.3	53.2	47.2	38.6	61.6	53.9	45.9	62.8	58.6	53.6
CAN [49]	36.0	58.2	**53.2**	**47.3**	53.6	57.6	51.0	40.1	64.6	55.4	47.3	63.2	59.1	56.5
ADR [47]	34.5	54.6	47.7	42.3	56.6	51.3	47.8	43.2	65.4	57.1	51.4	66.8	58.0	57.0
SE [48]	33.6	56.6	50.3	40.6	55.5	55.1	48.6	45.3	67.2	61.3	52.4	70.0	66.3	60.4
InstaPBM(ours)	**38.5**	**60.2**	52.3	47.0	**61.6**	**59.6**	**53.2**	**49.6**	**72.1**	**64.1**	**53.1**	**72.3**	**68.7**	**63.3**

Table 4: Performance comparison on **ID-TwO** based on ResNet-101. Best accuracy in bold.

Method	a-plane	bicycle	clock	dog	f-pan	lion	n-lace	potato	sock	b-ball	bucket	cup	d-bell	laptop	lipstick	panda	shark	w-bottle	Average
Source Only	52.8	69.5	67.9	61.2	51.6	56.3	94.0	30.2	53.7	46.8	57.5	59.9	61.3	57.3	55.0	47.8	45.7	59.8	57.1
DAN [8]	49.3	61.5	68.1	63.0	56.9	51.1	83.8	37.6	51.3	42.1	53.3	60.9	57.2	62.1	51.8	42.0	41.6	57.3	55.1
DANN [42]	56.7	82.0	76.3	62.7	51.3	54.4	88.1	35.7	55.7	43.7	72.1	57.0	63.3	57.5	54.6	43.6	44.4	59.8	58.8
HoMM [26]	54.0	75.9	68.1	59.8	53.2	57.9	93.1	36.2	55.4	54.1	73.1	58.4	68.1	67.7	54.8	46.5	57.2	62.1	60.9
CoRAL [44]	53.7	91.4	89.3	70.6	62.4	53.7	69.8	42.9	57.3	40.0	63.3	58.1	62.3	60.0	58.0	52.4	54.1	61.4	61.1
sDANN-β [13]	58.4	72.7	78.4	42.6	64.2	50.1	81.4	38.4	50.5	51.0	69.4	48.1	46.4	56.1	59.5	49.2	49.5	76.0	57.8
JAN [46]	53.7	69.7	76.5	62.6	67.2	55.1	85.2	40.4	56.5	51.1	71.9	50.1	64.4	65.5	56.5	44.7	47.5	68.0	60.4
CDAN [14]	55.5	87.0	73.2	62.5	61.6	54.9	83.3	**59.6**	58.6	53.5	74.2	70.0	69.3	69.2	56.8	44.6	**77.4**	74.0	65.8
CDAN+E [14]	55.8	76.5	77.4	70.7	61.6	58.4	89.6	47.6	61.4	61.1	75.4	67.1	81.9	71.1	55.3	46.1	52.9	78.6	66.0
CAN [49]	60.7	93.2	88.8	**74.5**	77.3	59.8	87.7	52.1	79.3	84.2	83.2	72.0	81.4	70.6	58.2	59.5	62.9	83.5	73.8
ADR [47]	57.7	85.8	82.1	73.5	73.0	62.7	87.5	52.6	73.6	68.2	79.4	67.5	79.6	77.3	53.1	55.1	52.1	77.3	69.9
SE [48]	58.3	83.9	84.9	71.3	73.5	65.0	86.7	53.1	77.6	74.0	81.5	**71.4**	77.8	74.3	56.3	57.7	55.3	81.4	71.3
InstaPBM(ours)	**62.7**	**96.3**	**91.3**	72.7	**76.3**	**66.3**	**91.5**	55.4	**86.2**	**86.8**	**87.5**	67.0	**84.9**	**92.7**	**61.8**	**61.8**	59.4	**92.2**	**77.4**

Table 5: Ablation study on different components in InstaPBM across all benchmarks.

	Conventional Benchmarks		RDS Benchmarks					
Method	VisDA2017	Digits5	V-LDS	D-LDS	DN-LDS	DN-ILDS	ID-TwO	Average
Baseline	49.4	62.8	55.1	55.8	45.4	56.0	57.1	54.5
+MIM	83.5	72.1	69.8	64.3	52.9	61.1	73.8	68.2
+CPBM_RA	77.9	67.5	62.2	59.5	50.7	57.7	69.5	63.6
+CPBM_NI	71.2	65.7	63.3	57.1	48.1	56.7	65.1	61.0
+CPBM_ALL	79.2	70.2	64.5	60.4	50.9	59.0	69.9	64.9
+MuPBM	75.3	67.1	59.5	58.9	49.2	58.7	64.3	61.8
+TPBM_ROT	62.3	64.3	56.1	57.2	45.7	56.1	60.5	57.5
+TPBM_QDR	64.6	66.2	58.2	59.3	46.1	56.7	61.3	58.9
+TPBM_FLIP	59.8	62.6	55.9	54.3	45.5	56.2	58.3	56.1
+TPBM_ALL	65.5	67.3	58.5	60.0	46.5	57.7	61.8	59.6
+InstaPBM	91.1	78.9	74.7	66.2	53.2	63.3	77.4	72.1

Baseline denotes model trained on source without adaptation. **+MIM** denotes using our mutual information maximization objective. **+MuPBM** represents Mix-up Predictive Behavior Matching. **+CPBM_*** denotes applying Contrastive Predictive Behavior Matching by Random Augmentation(RA), Noise Injection(NI) and both(ALL) specifically. **+TPBM_*** denotes implementing Task-oriented Predictive Behavior Matching by using different self-supervised tasks(i.s. Rotation Prediction(ROT), Vertical Flip Prediction(FLIP) and Patch Location Prediction(QDR)). **V-LDS** denotes **VisDA2017-LDS**, **D-LDS** denotes **Digits5-LDS**, **DN-LDS** and **DN-ILDS** denote **DomainNet-LDS** and **DomainNet-ILDS** respectively. **InstaPBM** denotes the integration of MIM, MuPBM, CPBM_ALL and TPBM_ALL.