Optimal transport for domain adaptation

Nicolas Courty, Remi Flamary, Devis Tuia, and Alain Rakotomamonjy

Presented by Mengying Yan @ ML in practice reading group

Introduction

- Domain adaptation challenge: divergence between the data probability distribution functions of the domains
- Main idea: transform data to make their distributions closer

Proposed framework:

- Use optimal transport to estimate the transportation map between the two distributions – align PDFs
- Use regularization terms for the optimal transport problem that exploits labels from the source domain

N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal Transport for Domain Adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, Sep. 2017, doi: 10.1109/TPAMI.2016.2615921.

Why optimal transport?

- Defines a transformation for each sample
- Can be used for computing distances (e.g. Wasserstein) and those distances can be evaluated directly on empirical distributions
- Can be used even when the supports of the distributions don’t overlap

Domain adaptation as a transportation problem

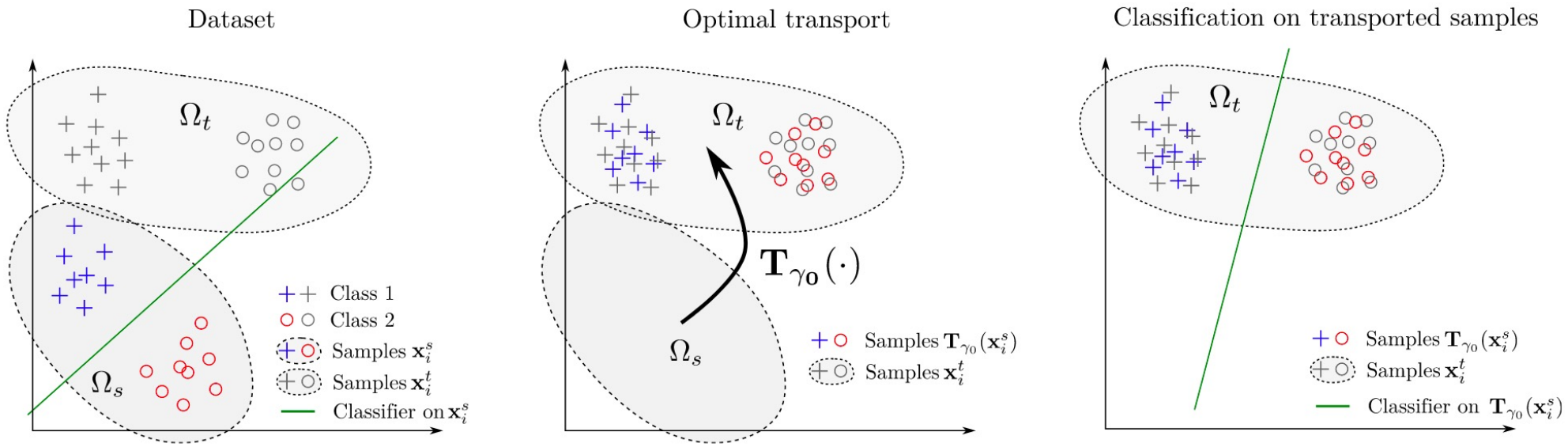


Fig. 1. Illustration of the proposed approach for domain adaptation. (left) Dataset for training, i.e., source domain, and testing, i.e., target domain. Note that a classifier estimated on the training examples clearly does not fit the target data. (middle) A data dependent transportation map $\mathbf{T}_{\gamma 0}$ is estimated and used to transport the training samples onto the target domain. Note that this transformation is usually non linear. (right) The transported labeled samples are used for estimating a classifier in the target domain.

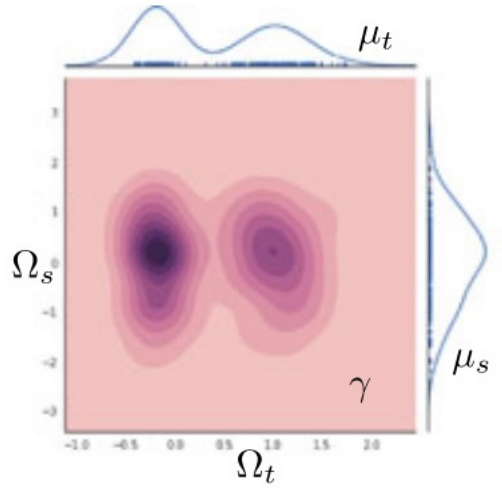
1. Estimate marginal distribution μ_s and μ_t from X_s and X_t
2. Find a transport map T from μ_s to μ_t
3. Use T to transport labeled samples X_s and train a classifier using them

Assumptions

- The domain drift is due to an unknown, possibly nonlinear map of the input space $\boldsymbol{T}: \Omega_s \rightarrow \Omega_t$
- The transformation preserves the conditional distribution $P_s(y|x^s) = P_t(y|\boldsymbol{T}(x^s))$
  - Label information is preserved by the transformation,

Discrete optimal transport

- Empirical distributions $\mu_s = \sum_{i=1}^{n_s} p_i^s \delta_{\mathbf{x}_i^s}, \quad \mu_t = \sum_{i=1}^{n_t} p_i^t \delta_{\mathbf{x}_i^t}$



- Probabilistic couplings (joint distribution)

$$\mathcal{B} = \left\{ \gamma \in (\mathbb{R}^+)^{\mathbf{n_s} \times \mathbf{n_t}} | \gamma \mathbf{1}_{\mathbf{n_t}} = \mu_{\mathbf{s}}, \gamma^{\mathbf{T}} \mathbf{1}_{\mathbf{n_s}} = \mu_{\mathbf{t}} \right\}$$

- Optimal transportation plan:

$$\gamma_0 = \underset{\gamma \in \mathcal{B}}{\operatorname{argmin}} \quad \langle \gamma, \mathbf{C} \rangle_F$$

$$\gamma_0 = \begin{array}{ll} \arg\min_{\gamma} & \int_{\Omega_s \times \Omega_t} c(\mathbf{x}, \mathbf{y}) \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} d\mathbf{y}, \\ \text{s.t.} & \int_{\Omega_t} \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{y} = \mu_s, \\ & \int_{\Omega_s} \gamma(\mathbf{x}, \mathbf{y}) d\mathbf{x} = \mu_t, \end{array}$$

- Cost function $c : \Omega_s \times \Omega_t \rightarrow \mathbb{R}^+$
- Chose l2 Euclidean di $C(i, j) = ||\mathbf{x}_i^s - \mathbf{x}_j^t||_2^2$.
- The optimal cost is the same as W1 Wasserstein distance

Regularized optimal transport

- Optimization problem (for domain adaptation)

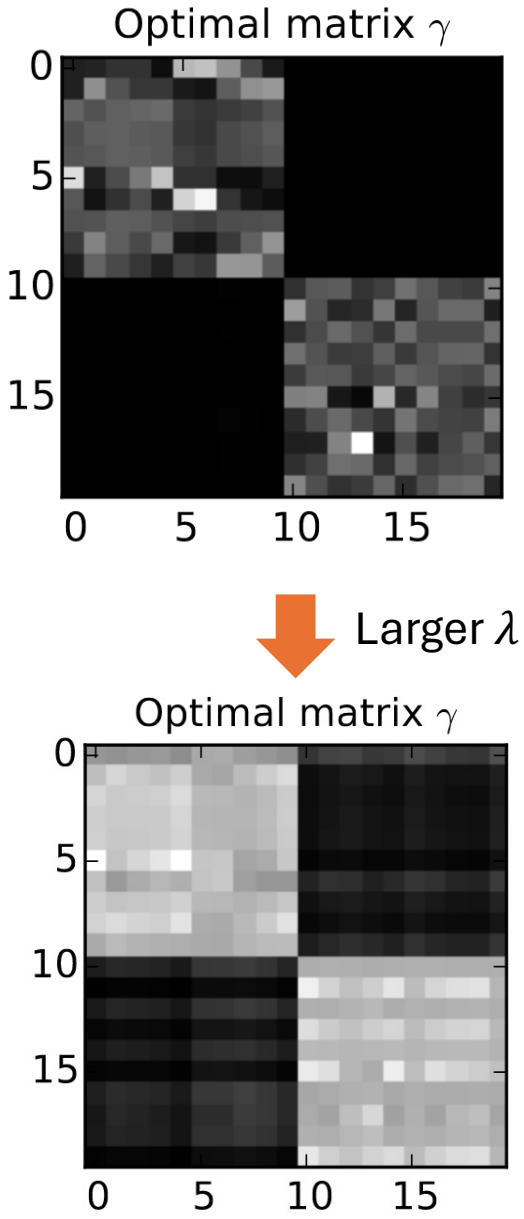
$$\min_{\gamma \in \mathcal{B}} \quad \langle \gamma, \mathbf{C} \rangle_F + \lambda \Omega_s(\gamma) + \eta \Omega_c(\gamma)$$

- Entropic regularization
- Class-based regularization (use label info in source domain)
  - Group Lasso
  - Laplacian
- Semi-supervised regularization

Entropic Regularization

$$\Omega_s(\boldsymbol{\gamma}) = \sum_{i,j} \boldsymbol{\gamma}(i,j) \log \boldsymbol{\gamma}(i,j)$$

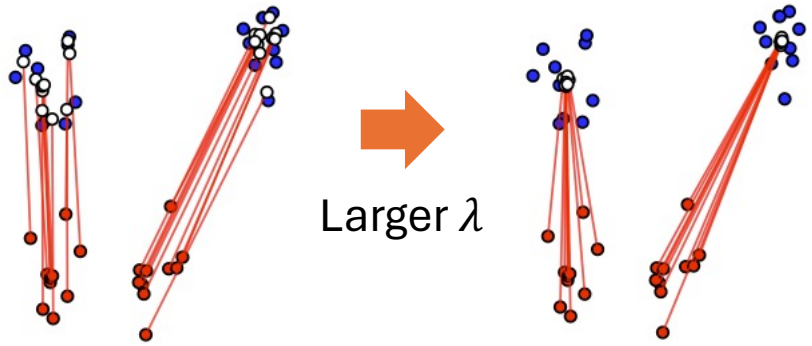
- To avoid overfitting
- Want to decrease sparsity by increasing entropy
- Equivalent to KL divergence between γ and uniform joint distribution $\gamma_u = \frac{1}{n_s n_t} : \mathrm{KL}(\gamma || \gamma_u)$
- Efficient Sinkhorn Knopp algorithm



M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transportation,” in Proc. Advances Neural Inf. Process. Syst., 2013, pp. 2292–2300.

Group Lasso Regularization

$$\Omega_c(\gamma) = \sum_j \sum_{cl} ||\gamma(\mathcal{I}_{cl}, j)||_2$$

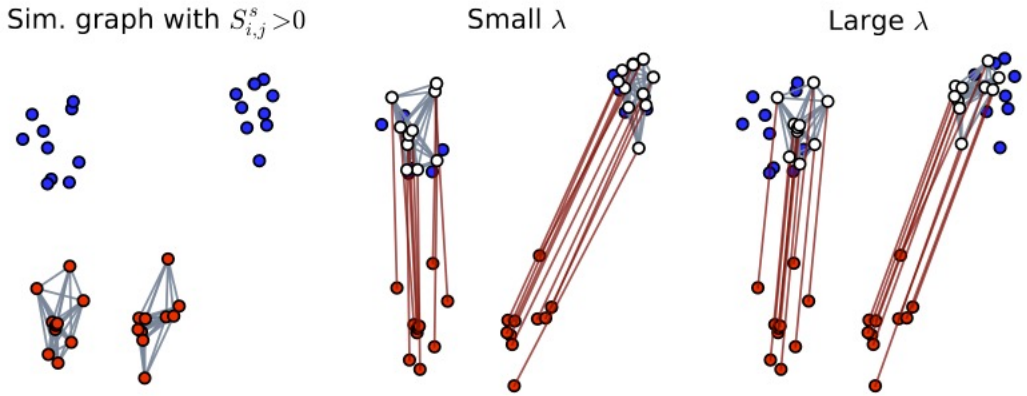


- Want each target sample to receive masses only from source samples that have the same label
- I_{cl} contains the indices of rows in γ related to source domain samples of class cl
- Convex – so that generalized conditional gradient can be used
- Assumption: distribution of labels $P_s(y) = P_t(y)$

N. Courty, R. Flamary, and D. Tuia, “Domain adaptation with regularized optimal transport,” in Proc. Eur. Conf. Mach. Learn. Principles Practice Knowl. Discovery Databases, 2014, pp. 1–16.

Laplacian regularization

$$\Omega_c(\gamma) = \frac{1}{N_s^2} \sum_{i,j} S_s(i,j) \|\hat{\mathbf{x}}_i^s - \hat{\mathbf{x}}_j^s\|_2^2$$



- Want similar samples in the source domain to also be similar after transportation
- Graph-based
- $S_s(i,j)$ positive symmetric similarity matrix for source
- In practice, prune S_s using class information in source

$S_s(i,j) = 0$ if $\mathrm{y}_i^s \neq y_j^s$

M. Carreira-Perpinan and W. Wang, “LASS: A simple assignment model with Laplacian smoothing,” in Proc. 28th AAAI Conf. Artif. Intell., 2014, pp. 1715–1721.
S. Ferradans, N. Papadakis, J. Rabin, G. Peyre, and J.-F. Aujol, “Regularized discrete optimal transport,” in Proc. Scale Space Variational Methods Comput. Vis., 2013, pp. 428–439.

Semi-supervised regularization

$$\Omega_{semi}(\gamma) = \langle \gamma, \mathbf{M} \rangle$$

- Few labeled samples are available in the target domain
- Cost matrix $M(i,j) = 0$ if $\mathrm{y}_i^s = y_j^t$, otherwise infinity

Generalized conditional gradient (GCG)

- Our optimization problem $\min_{\gamma \in \mathcal{B}} \quad \langle \gamma, \mathbf{C} \rangle_F + \lambda \Omega_s(\gamma) + \eta \Omega_c(\gamma)$
- General case: $\min_{\gamma \in \mathcal{B}} \quad f(\gamma) + g(\gamma)$
  - F differentiable, g convex

$$f(\gamma) = \langle \gamma, \mathbf{C} \rangle_F + \eta \Omega_c(\gamma)$$

$$g(\gamma) = \lambda \Omega_s(\gamma)$$

Algorithm 1. Generalized Conditional Gradient

1: Initialize $k = 0$ and $\gamma^0 \in \mathcal{P}$
2: **repeat**
3: With $\mathbf{G} \in \nabla f(\gamma^k)$, solve

$$\gamma^\star = \operatorname*{argmin}_{\gamma \in \mathcal{B}} \quad \langle \gamma, \mathbf{G} \rangle_F + g(\gamma)$$

4: Find the optimal step α^k

$$\alpha^k = \operatorname*{argmin}_{0 \le \alpha \le 1} f(\gamma^k + \alpha \Delta\gamma) + g(\gamma^k + \alpha \Delta\gamma)$$

with $\Delta\gamma = \gamma^* - \gamma^k$
5: $\gamma^{k+1} \leftarrow \gamma^k + \alpha^k \Delta\gamma$, set $k \leftarrow k + 1$
6: **until** Convergence

K. Bredies, D. A. Lorenz, and P. Maass, “A generalized conditional gradient method and its connection to an iterative shrinkage method,” Comput. Optimization Appl., vol. 42, no. 2, pp. 173193, 2009

Implementation

- For this paper: https://remi.flamary.com/soft/soft-transp.html

We provide regularized optimal transport solvers of the form:

$$\min_{\gamma \in \mathcal{P}} \quad \langle \mathbf{M}, \boldsymbol{\gamma} \rangle_F \quad + \quad \Omega(\boldsymbol{\gamma})$$

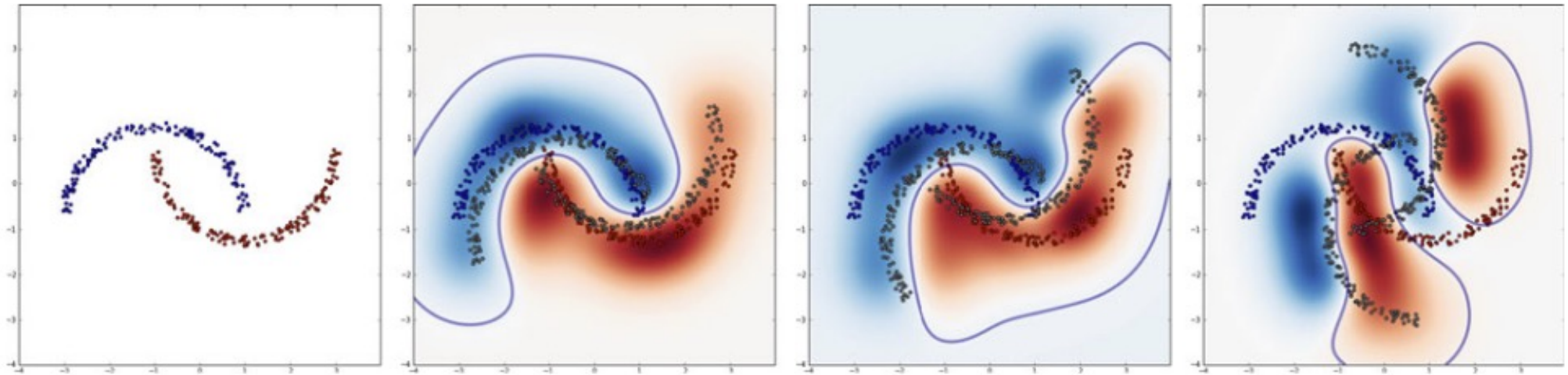
where $\mathcal{P} = \{\boldsymbol{\gamma}\mathbf{1} = \boldsymbol{\mu}_t, \boldsymbol{\gamma}^\top \mathbf{1} = \boldsymbol{\mu}_s, \boldsymbol{\gamma} \geq 0\}$ is the convex set of matrices satisfying marginal $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_t$ and $\mathbf{M}$ is a transportation cost matrix.

The regularization term $\Omega(\boldsymbol{\gamma})$ can be :

- Classic LP transport: $\Omega(\boldsymbol{\gamma}) = 0$
- Sinkhorn regularization : $\Omega(\boldsymbol{\gamma}) = \frac{1}{\lambda} \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j}$
- Sinkhorn + Class regularization : $\Omega(\boldsymbol{\gamma}) = \frac{1}{\lambda} \sum_{i,j} \gamma_{i,j} \log \gamma_{i,j} + \eta \sum_j \sum_c ||\gamma_{\mathcal{I}_c,j}||_q^p$

- Python Package POT: https://pythonot.github.io
- Source code: https://github.com/PythonOT/POT
  - Actively maintained and updated
  - Detailed comments for functions

Simulation: two moons



(a) source domain (b) rotation=20° (c) rotation=40° (d) rotation=90°

Simulation: two moons -- results

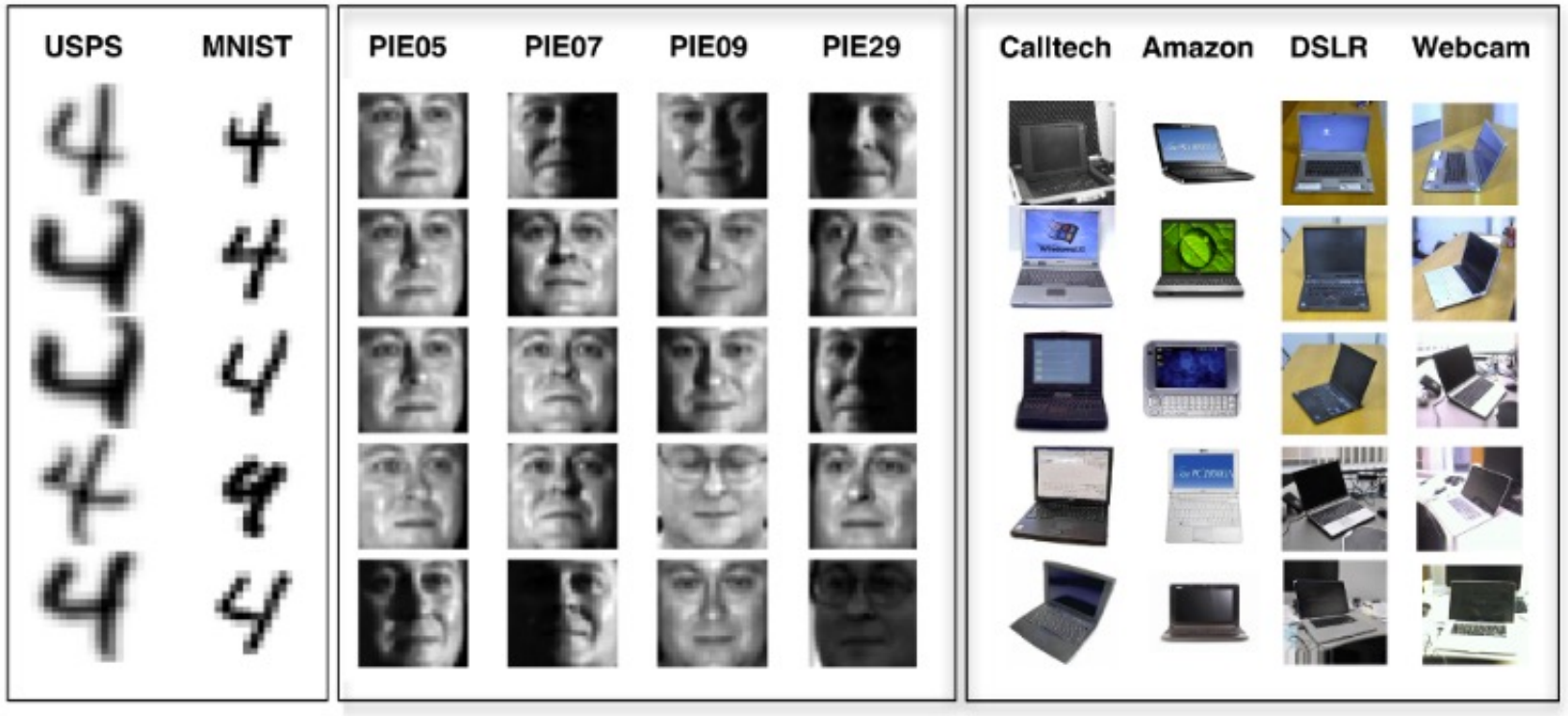
TABLE 1
Mean Error Rate over 10 Realizations for the
Two Moons Simulated Example

Target rotation angle	10°	20°	30°	40°	50°	70°	90°
SVM (no adapt.)	0	0.104	0.24	0.312	0.4	0.764	0.828
DASVM [9]	0	0	0.259	0.284	0.334	0.747	0.82
PBDA [22]	0	0.094	0.103	0.225	0.412	0.626	0.687
OT-exact	0	0.028	0.065	0.109	0.206	0.394	0.507
OT-IT	0	0.007	0.054	0.102	0.221	0.398	0.508
OT-GL	0	0	0	0.013	0.196	0.378	0.508
OT-Laplace	0	0	0.004	0.062	0.201	0.402	0.524

1. OT-exact: non-regularized OT
2. OT-IT: Entropic regularization
3. OT-GL: Group-lasso + entropic regularization
4. OT-Lap: Laplacian + entropic regularization

Align distributions and then use SVM as a classifier

Visual application



Visual application – results1

TABLE 3
Overall Recognition Accuracies in Percent Obtained over All Domains Pairs Using the SURF Features

Domains	1NN	PCA	GFK	TSL	JDA	OT-exact	OT-IT	OT-Laplace	OT-LpL1	OT-GL
U→M	39.00	37.83	44.16	40.66	54.52	50.67	53.66	57.42	**60.15**	57.85
M→U	58.33	48.05	60.96	53.79	60.09	49.26	64.73	64.72	68.07	**69.96**
mean	48.66	42.94	52.56	47.22	57.30	49.96	59.20	61.07	**64.11**	63.90
P1→P2	23.79	32.61	22.83	34.29	**67.15**	52.27	57.73	58.92	59.28	59.41
P1→P3	23.50	38.96	23.24	33.53	56.96	51.36	57.43	57.62	58.49	**58.73**
P1→P4	15.69	30.82	16.73	26.85	40.44	40.53	47.21	47.54	47.29	**48.36**
P2→P1	24.27	35.69	24.18	33.73	**63.73**	56.05	60.21	62.74	62.61	61.91
P2→P3	44.45	40.87	44.03	38.35	**68.42**	59.15	63.24	64.29	62.71	64.36
P2→P4	25.86	29.83	25.49	26.21	49.85	46.73	51.48	**53.52**	50.42	52.68
P3→P1	20.95	32.01	20.79	39.79	**60.88**	54.24	57.50	57.87	58.96	57.91
P3→P2	40.17	38.09	40.70	39.17	65.07	59.08	63.61	**65.75**	64.04	64.67
P3→P4	26.16	36.65	25.91	36.88	52.44	48.25	52.33	**54.02**	52.81	52.83
P4→P1	18.14	29.82	20.11	40.81	**46.91**	43.21	45.15	45.67	46.51	45.73
P4→P2	24.37	29.47	23.34	37.50	**55.12**	46.76	50.71	52.50	50.90	51.31
P4→P3	27.30	39.74	26.42	46.14	**53.33**	48.05	52.10	52.71	51.37	52.60
mean	26.22	34.55	26.15	36.10	**56.69**	50.47	54.89	56.10	55.45	55.88
C→A	20.54	35.17	35.29	45.25	40.73	30.54	37.75	38.96	**48.21**	44.17
C→W	18.94	28.48	31.72	37.35	33.44	23.77	31.32	31.13	38.61	**38.94**
C→D	19.62	33.75	35.62	39.25	39.75	26.62	34.50	36.88	39.62	**44.50**
A→C	22.25	32.78	32.87	**38.46**	33.99	29.43	31.65	33.12	35.99	34.57
A→W	23.51	29.34	32.05	35.70	36.03	25.56	30.40	30.33	35.63	**37.02**
A→D	20.38	26.88	30.12	32.62	32.62	25.50	27.88	27.75	36.38	**38.88**
W→C	19.29	26.95	27.75	29.02	31.81	25.87	31.63	31.37	33.44	**35.98**
W→A	23.19	28.92	33.35	34.94	31.48	27.40	37.79	37.17	37.33	**39.35**
W→D	53.62	79.75	79.25	80.50	**84.25**	76.50	80.00	80.62	81.38	84.00
D→C	23.97	29.72	29.50	31.03	29.84	27.30	29.88	31.10	31.65	**32.38**
D→A	27.10	30.67	32.98	36.67	32.85	29.08	32.77	33.06	37.06	**37.17**
D→W	51.26	71.79	69.67	77.48	80.00	65.70	72.52	76.16	74.97	**81.06**
mean	28.47	37.98	39.21	42.97	44.34	36.69	42.30	43.20	46.42	**47.70**

Unsupervised DA
Classifier: one-nearest neighbor (1NN)

- γ and η are validated on validation target set (0.001; 0.01; 0.1; 1; 10; 100; 1000)
- S_s: nearest neighbor graph (pruned by classes)

Visual application – results2

TABLE 4
Results of Adaptation by Optimal Transport
Using DeCAF Features

	Layer 6				Layer 7			
Domains	DeCAF	JDA	OT-IT	OT-GL	DeCAF	JDA	OT-IT	OT-GL
C→A	79.25	88.04	88.69	**92.08**	85.27	89.63	91.56	**92.15**
C→W	48.61	79.60	75.17	**84.17**	65.23	79.80	82.19	**83.84**
C→D	62.75	84.12	83.38	**87.25**	75.38	85.00	85.00	**85.38**
A→C	64.66	81.28	81.65	**85.51**	72.80	82.59	84.22	**87.16**
A→W	51.39	80.33	78.94	**83.05**	63.64	83.05	81.52	**84.50**
A→D	60.38	**86.25**	85.88	85.00	75.25	85.50	**86.62**	85.25
W→C	58.17	**81.97**	74.80	81.45	69.17	79.84	81.74	**83.71**
W→A	61.15	90.19	80.96	**90.62**	72.96	90.94	88.31	**91.98**
W→D	97.50	**98.88**	95.62	96.25	98.50	**98.88**	98.38	91.38
D→C	52.13	81.13	77.71	**84.11**	65.23	81.21	82.02	**84.93**
D→A	60.71	91.31	87.15	**92.31**	75.46	91.92	92.15	**92.92**
D→W	85.70	**97.48**	93.77	96.29	92.25	**97.02**	96.62	94.17
mean	65.20	86.72	83.64	**88.18**	75.93	87.11	87.53	**88.11**

DeCAF: activation feature extracted from 6th and 7th layers of a CNN (imageNet)

Visual application – results3 semi-supervised

TABLE 5
Results of Semi-Supervised Adaptation with Optimal Transport Using the SURF Features

	Unsupervised + labels		Semi-supervised		
Domains	OT-IT	OT-GL	OT-IT	OT-GL	MMDT [28]
C→A	37.0 ± 0.5	41.4 ± 0.5	46.9 ± 3.4	47.9 ± 3.1	**49.4 ± 0.8**
C→W	28.5 ± 0.7	37.4 ± 1.1	64.8 ± 3.0	**65.0 ± 3.1**	63.8 ± 1.1
C→D	35.1 ± 1.7	44.0 ± 1.9	59.3 ± 2.5	**61.0 ± 2.1**	56.5 ± 0.9
A→C	32.3 ± 0.1	36.7 ± 0.2	36.0 ± 1.3	**37.1 ± 1.1**	36.4 ± 0.8
A→W	29.5 ± 0.8	37.8 ± 1.1	63.7 ± 2.4	**64.6 ± 1.9**	**64.6 ± 1.2**
A→D	36.9 ± 1.5	46.2 ± 2.0	57.6 ± 2.5	**59.1 ± 2.3**	56.7 ± 1.3
W→C	35.8 ± 0.2	36.5 ± 0.2	38.4 ± 1.5	**38.8 ± 1.2**	32.2 ± 0.8
W→A	39.6 ± 0.3	41.9 ± 0.4	47.2 ± 2.5	47.3 ± 2.5	**47.7± 0.9**
W→D	77.1 ± 1.8	**80.2 ± 1.6**	79.0 ± 2.8	79.4 ± 2.8	67.0 ± 1.1
D→C	32.7 ± 0.3	34.7 ± 0.3	35.5 ± 2.1	**36.8 ± 1.5**	34.1 ± 1.5
D→A	34.7 ± 0.3	37.7 ± 0.3	45.8 ± 2.6	46.3 ± 2.5	**46.9 ± 1.0**
D→W	81.9 ± 0.6	**84.5 ± 0.4**	83.9 ± 1.4	84.0 ± 1.5	74.1 ± 0.8
mean	41.8	46.6	54.8	**55.6**	52.5

- SURF: transform each image into a 800 bins histogram
- Three labels per class
- Unsupervised + labels (unsupervised alignment + learning with labels)
- Semi-supervised (target labels are used for alignment)

Recommendations

Well written and well organized. Good first paper to read

Discussion

- This paper focuses on using regularization to preserve label information whereas Chapel et al. (2020) use partial transport (transport only part of the mass)
- Classifier
- Labeling mechanism