Population-based Ensemble Classifier Induction for Domain Adaptation – Nguyen et. al

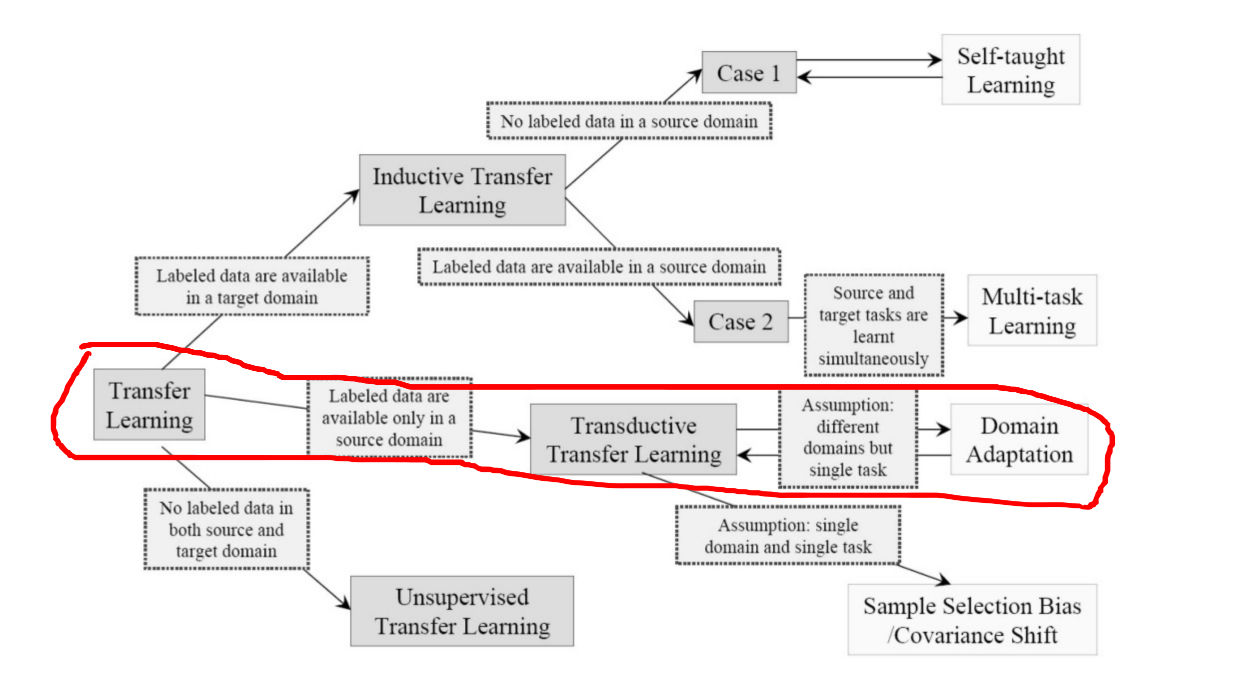
Presented By: Ritam Guha (MSU ID: guharita)

Date: 29th March, 2021

Introduction to Domain Adaptation

- In a standard supervised learning environment, the training data and target data come from the same domain.
- Domain is defined in terms of feature space and domain distribution.
- Is this a Problem??
- What do we do when target data is from a different distribution than the train data?? Simplest Intuitive Solution: Re-train the model. It has lots of problems: High cost, unavailability of labeled data.
- Find a similar domain (source domain) and use it to improve the learning experience in the target domain – Transfer Learning.
- Domain Adaptation: A special case of transfer learning where the two domains a terms of feature space.

Transfer Learning Summary



Transfer Learning

Labeled data are available in a target domain

Inductive Transfer Learning

No labeled data in a source domain

Case 1

Self-taught Learning

Labeled data are available in a source domain

Case 2

Source and target tasks are learnt simultaneously

Multi-task Learning

Labeled data are available only in a source domain

Transductive Transfer Learning

Assumption: different domains but single task

Domain Adaptation

Assumption: single domain and single task

Sample Selection Bias /Covariance Shift

No labeled data in both source and target domain

Unsupervised Transfer Learning

*Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. IEEE Transactions on Knowledge and Da
22, 10 (2010), 1345ś1359.

Solving Domain Adaptation

- The main goal of domain adaptation is to make the target domain as similar to the source domain as possible.
- Two approaches: find a common feature space which reduce divergence between the two domains, instance reweighting based on similarity with the target domain.
- Two classes of algorithms:
  - Transfer Subspace Learning (TSL): First uses feature extraction, then learns the classifier.
  - Transfer Classifier Induction (TCI): Merges the two steps and tries to find an adaptive classifier
- Transfer Classifier Induction works better than Transfer Subspace Learning and that is the focus of this paper.
- Problems with TCI:
  - Uses gradient-based optimization. May get stuck to a local optimum.
  - Outputs single classifier which can easily get over-fitted to the source domain.

Transfer Classifier Induction: MEDA

- The paper tries to improve the performance of an existing transfer classifier induction approach by importing some concepts from Evolutionary Computation (EC).
- The underlying approach used by the paper is: Manifold Embedded Distribution Alignment (MEDA).
- The steps used in MEDA are:
  - Convert the source and target data to a manifold space.
    - $X = [X_s , X_t]$, $Z = (G)^{0.5}X$ [G is Geodesic Flow Kernel conversion]
  - The objective function:

$$F = argmin_f \underbrace{\sum_{i=1}^{n}(y_i - f(z_i))^2}_{\text{reduces difference between target and generated labels}} + \underbrace{\mu||f||^2}_{\text{regularization term}} + \lambda \underbrace{D_f(D_s, D_t)}_{\text{reduce diiference in domain distribution}} + \rho \underbrace{R_f(D_s, D_t)}_{\text{preserve geometrical properties among distributions}}$$

  - The transformed objective function:

$$F = ||(Y - \beta^T K)A||_F^2 + \mu \times tr(\beta^T K \beta) + tr(\beta^T K(\lambda M + \rho L)K\beta)$$

  - Setting the derivative of the objective function with respect to β=0: (Here Y is actual + pseudo targets

$$\beta^* = ((A + \lambda M + \rho K) + \mu I)^{-1} AY^T$$

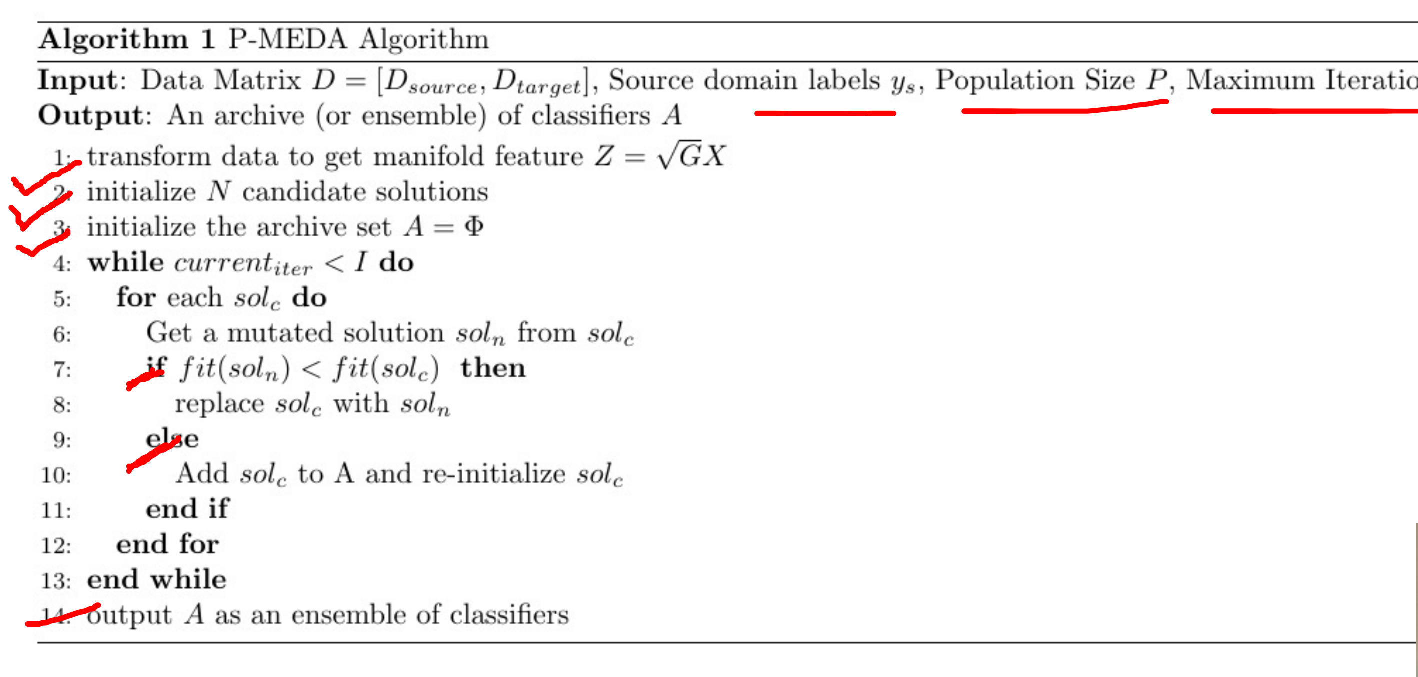
Evolutionary Transfer Classifier Induction: P-MEDA

- P-MEDA uses a population of solutions, instead of a single solution. This addition along with some other EC concepts solve both the issues faced by MEDA.
- Defining the Evolutionary Search Process:
  - **Solution Representation:** β is a matrix which defines a classifier. So, for this approach, the matrix is flattened and turned into a vector which can represent a single solution to the problem.
  - **Fitness function:** Fitness function for the search process is same as the objective function in the MEDA approach. Every candidate solution is converted back to the β matrix and it is used in the equation to get the fitness score. Once calculated, the pseudo target labels and fitness are recorded for each candidate solution.
  - **Solution Update:** A candidate solution is mutated using the same process used in MEDA. But instead of directly replacing it without any comparison, P-MEDA first compares the new solution and it only replaces the parent if it is better.

    If the mutated solution becomes worse than the parent, the parent is added to an Arc optimum) and it is re-initialized.

    Re-initialization can be done randomly or using information from the archive.

P-MEDA Algorithm



Algorithm 1 P-MEDA Algorithm

Input: Data Matrix $D = [D_{source}, D_{target}]$, Source domain labels y_s, Population Size P, Maximum Iterations I

Output: An archive (or ensemble) of classifiers A

```
transform data to get manifold feature $Z = \sqrt{G}X$
initialize $N$ candidate solutions
initialize the archive set $A = \Phi$
while $current_{iter} < I$ do
    for each $sol_c$ do
        Get a mutated solution $sol_n$ from $sol_c$
        if $fit(sol_n) < fit(sol_c)$ then
            replace $sol_c$ with $sol_n$
        else
            Add $sol_c$ to A and re-initialize $sol_c$
        end if
    end for
end while
output $A$ as an ensemble of classifiers
```

Experimental Data

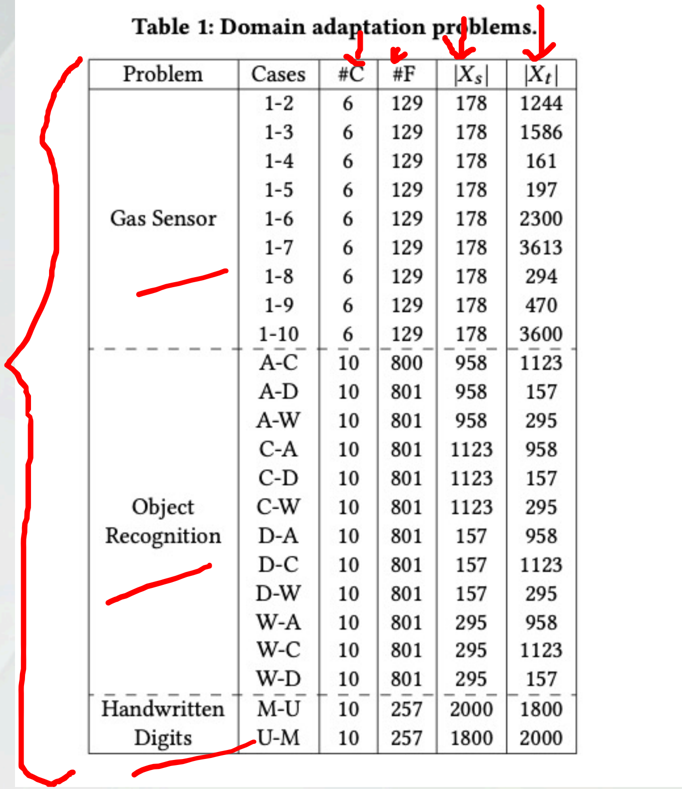


Table 1: Domain adaptation problems.

Problem	Cases	#C	#F	$\|X_s\|$	$\|X_t\|$
Gas Sensor	1-2	6	129	178	1244
	1-3	6	129	178	1586
	1-4	6	129	178	161
	1-5	6	129	178	197
	1-6	6	129	178	2300
	1-7	6	129	178	3613
	1-8	6	129	178	294
	1-9	6	129	178	470
	1-10	6	129	178	3600
Object Recognition	A-C	10	800	958	1123
	A-D	10	801	958	157
	A-W	10	801	958	295
	C-A	10	801	1123	958
	C-D	10	801	1123	157
	C-W	10	801	1123	295
	D-A	10	801	157	958
	D-C	10	801	157	1123
	D-W	10	801	157	295
	W-A	10	801	295	958
	W-C	10	801	295	1123
	W-D	10	801	295	157
Handwritten Digits	M-U	10	257	2000	1800
	U-M	10	257	1800	2000

Experimental Comparison

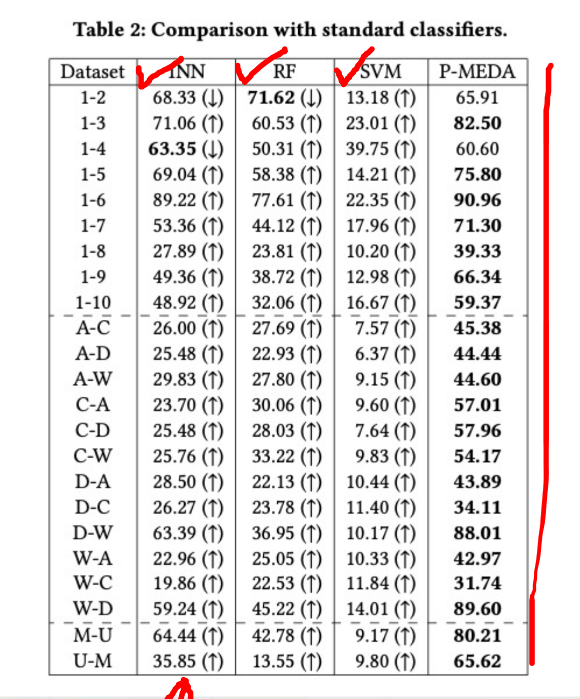


Table 2: Comparison with standard classifiers.

Dataset	1NN	RF	SVM	P-MEDA
1-2	68.33 (↓)	**71.62** (↓)	13.18 (↑)	65.91
1-3	71.06 (↑)	60.53 (↑)	23.01 (↑)	**82.50**
1-4	**63.35** (↓)	50.31 (↑)	39.75 (↑)	60.60
1-5	69.04 (↑)	58.38 (↑)	14.21 (↑)	**75.80**
1-6	89.22 (↑)	77.61 (↑)	22.35 (↑)	**90.96**
1-7	53.36 (↑)	44.12 (↑)	17.96 (↑)	**71.30**
1-8	27.89 (↑)	23.81 (↑)	10.20 (↑)	**39.33**
1-9	49.36 (↑)	38.72 (↑)	12.98 (↑)	**66.34**
1-10	48.92 (↑)	32.06 (↑)	16.67 (↑)	**59.37**
A-C	26.00 (↑)	27.69 (↑)	7.57 (↑)	**45.38**
A-D	25.48 (↑)	22.93 (↑)	6.37 (↑)	**44.44**
A-W	29.83 (↑)	27.80 (↑)	9.15 (↑)	**44.60**
C-A	23.70 (↑)	30.06 (↑)	9.60 (↑)	**57.01**
C-D	25.48 (↑)	28.03 (↑)	7.64 (↑)	**57.96**
C-W	25.76 (↑)	33.22 (↑)	9.83 (↑)	**54.17**
D-A	28.50 (↑)	22.13 (↑)	10.44 (↑)	**43.89**
D-C	26.27 (↑)	23.78 (↑)	11.40 (↑)	**34.11**
D-W	63.39 (↑)	36.95 (↑)	10.17 (↑)	**88.01**
W-A	22.96 (↑)	25.05 (↑)	10.33 (↑)	**42.97**
W-C	19.86 (↑)	22.53 (↑)	11.84 (↑)	**31.74**
W-D	59.24 (↑)	45.22 (↑)	14.01 (↑)	**89.60**
M-U	64.44 (↑)	42.78 (↑)	9.17 (↑)	**80.21**
U-M	35.85 (↑)	13.55 (↑)	9.80 (↑)	**65.62**

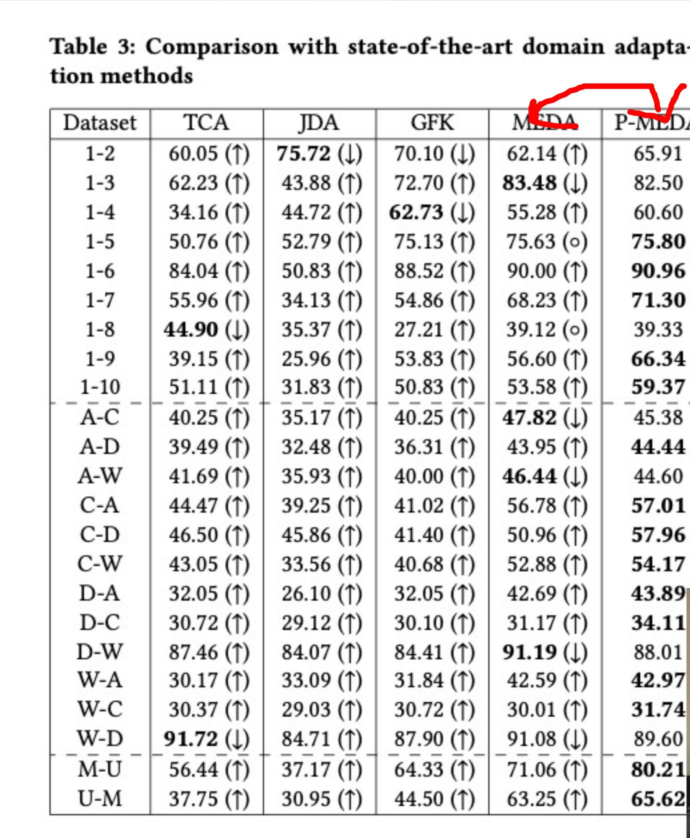
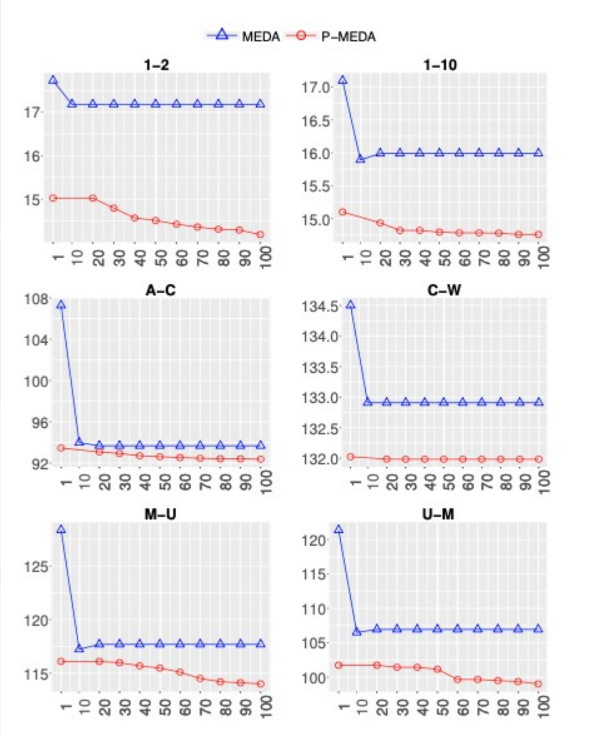


Table 3: Comparison with state-of-the-art domain adaptation methods

Dataset	TCA	JDA	GFK	MEDA	P-MEDA
1-2	60.05 (↑)	**75.72** (↓)	70.10 (↓)	62.14 (↑)	65.91
1-3	62.23 (↑)	43.88 (↑)	72.70 (↑)	**83.48** (↓)	82.50
1-4	34.16 (↑)	44.72 (↑)	**62.73** (↓)	55.28 (↑)	60.60
1-5	50.76 (↑)	52.79 (↑)	75.13 (↑)	75.63 (○)	**75.80**
1-6	84.04 (↑)	50.83 (↑)	88.52 (↑)	90.00 (↑)	**90.96**
1-7	55.96 (↑)	34.13 (↑)	54.86 (↑)	68.23 (↑)	**71.30**
1-8	**44.90** (↓)	35.37 (↑)	27.21 (↑)	39.12 (○)	39.33
1-9	39.15 (↑)	25.96 (↑)	53.83 (↑)	56.60 (↑)	**66.34**
1-10	51.11 (↑)	31.83 (↑)	50.83 (↑)	53.58 (↑)	**59.37**
A-C	40.25 (↑)	35.17 (↑)	40.25 (↑)	**47.82** (↓)	45.38
A-D	39.49 (↑)	32.48 (↑)	36.31 (↑)	43.95 (↑)	**44.44**
A-W	41.69 (↑)	35.93 (↑)	40.00 (↑)	**46.44** (↓)	44.60
C-A	44.47 (↑)	39.25 (↑)	41.02 (↑)	56.78 (↑)	**57.01**
C-D	46.50 (↑)	45.86 (↑)	41.40 (↑)	50.96 (↑)	**57.96**
C-W	43.05 (↑)	33.56 (↑)	40.68 (↑)	52.88 (↑)	**54.17**
D-A	32.05 (↑)	26.10 (↑)	32.05 (↑)	42.69 (↑)	**43.89**
D-C	30.72 (↑)	29.12 (↑)	30.10 (↑)	31.17 (↑)	**34.11**
D-W	87.46 (↑)	84.07 (↑)	84.41 (↑)	**91.19** (↓)	88.01
W-A	30.17 (↑)	33.09 (↑)	31.84 (↑)	42.59 (↑)	**42.97**
W-C	30.37 (↑)	29.03 (↑)	30.72 (↑)	30.01 (↑)	**31.74**
W-D	**91.72** (↓)	84.71 (↑)	87.90 (↑)	91.08 (↓)	89.60
M-U	56.44 (↑)	37.17 (↑)	64.33 (↑)	71.06 (↑)	**80.21**
U-M	37.75 (↑)	30.95 (↑)	44.50 (↑)	63.25 (↑)	**65.62**

Experimental Outcome: MEDA vs P-MEDA



Conclusion

- Addition of EC concepts improve the quality of solutions produced by MEDA. It gets rid of the two problems faced by Transfer Classifier Induction techniques.
- Although the paper focuses on improving MEDA, it is a generalized approach. So, it is applicable in any such framework.
- The main key is the generalization power. Multiple classifier make the process more generalized.

Thank You

Questions on D2L Discussion