CSE 848 Paper Summary
Population-based Ensemble Classifier Induction for Domain Adaptation - Nguyen et. al

Submitted by: Ritam Guha (MSU ID: guharita)

Date: March 22, 2021

1 Introduction

Most of the standard supervised learning are domain specific in nature which means that a classifier trained on a dataset from a particular domain is only applicable to a test data which is from the same domain i.e. the same feature space and the same domain distribution. But these assumptions are not applicable in real-life scenarios. If the target data comes from a different domain, the models need to be re-trained on data from the new domain which makes the process time-consuming and expensive when labeled data is not available for the new domain. The goal of transfer learning is to take help of the labeled data from a domain (source domain) to improve the performance on a similar domain (target domain). Domain Adaption is a subclass of transfer learning task when the source and target domains share the same feature space. The problem of domain adaptation can be solved by a class of approaches which tend to learn a subset of the features which reduces the divergence between data from both the domains. This class of procedures is known as Transfer Classifier Induction.

Primary Problems: Modern transfer classifier induction algorithms face two important problems which make them inapplicable in many situations:

- Most algorithms use gradient-based optimization approach which makes them vulnerable to getting stuck at local optima if the objective function is a multi-modal function.
- They focus on finding a single classifier which can get easily over-fitted to the data from the source domain. But for domain adaption, generalization is very imprortant because the source and target domains are different.

2 Background

The paper uses concepts from Evolutionary Computation (EC) over an existing transfer classifier induction technique known as Manifold Embedded Distribution Alignment (MEDA) [1] to get rid of the problems. MEDA first uses a transformation matrix G (found through Geodesic Flow Kernel [2]) to convert the original feature space into another feature space which avoids feature distortion: $Z = \sqrt{G}X$ where X is the collection of source and target domain samples and Z becomes the transformed samples. The objective function then becomes:

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

The components of Equation 1 altogether try to reduce the error of the predictions and at the same time attempts to make the two distributions (source and target) as close as possible. After using appropriate conversions, the objective function can be represented as:

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

Setting its derivative wrt β to 0 gives us:

$$\beta^* = ((A + \lambda M + \rho K) + \mu I)^{-1} A Y^T \tag{3}$$

Here A, K, I are fixed and M depends on Y. So, finally F is only dependent on Y which is the comabined labels of both the domains. As the data from the target domain are not labeled, a KNN classifier is initially used to label those data samples. The optimization starts with these labels, generates a β^* which in turn gives new labels and the loop continues until we find some convergence.

3 Proposed Solution

The proposed approach called Population-based MEDA (or P-MEDA) uses a population of classifier candidates to circumvent local optima and also to find an ensemble of classifiers which makes it more generalized. For any EC approach, three important requirements are solution representation, fitness function and evolving solutions. These three factors designed for P-MEDA are discussed below:

Solution Representation: From Equation 2, we can see that the fitness function is dependent on the matrix β which is basically a representation of the components of a classifier. In P-MEDA approach, the matrix is flattened and represented as a vector of values. Each such β represents a candidate solution (a classifier) for the problem.

Fitness Function: The candidate solutions are first converted into the matrix for β and the Maximum Mean Discrepancy (MMD) matrix [3] M is computed from β which is then used to generated labels for the target domain (called pseudo targte labels). These values are then plugged into Equation 2 to get the fitness score for a particular candidate solution. These fitness scores and pseudo target values are recorded in the candidate solutions.

Evolving Solutions: In P-MEDA, for each solutions sol_c, a new solution sol_n is generated using Equation 3. But sol_n replaces sol_c only if its fitness is better than that of sol_c. Else, it is assumed that sol_c is a local optimum and it is added to an archive A which contains the ensemble of the classifiers. When this happends, sol_c should be initialized with something in order to continue the searching process. Two methods are proposed for the initialization:

- Randomly allocate values to the vectorized representation of β which initializes sol_c to a completely new location.
- Each member of A is assumed to be a good solution. So, they can be used to find a proper initialization of sol_c. But it is not advised that much because if A is used, the diversity of the solutions reduce too much.

The overall algorithm is given by:

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

4 Results

P-MEDA is examined on three different real-world problems namely Gas Sensor [4], Handwritten Digits [5], Object Recognition [6, 7] consisting of 23 domain adaptation cases. At first, P-MEDA is compared with three standalone classifiers namely KNN, Random Forest and SVM. In this experimental setting, SVM is found to be the worst one for domain adaptation. Although KNN with $k = 1$ is simpler than SVM and RF, it performs better than them because it is more generalizable which indicates that generalization is the most important key for domain adaptation. P-MEDA outperforms all of them for 21 out of the 23 cases.

In the next round, P-MEDA is compared with state-of-the-art domain adaptation techniques like TCA, JDA, GFK and MEDA (predecessor of P-MEDA) and it was able to achieve better results for 15 out of the 23 cases.

5 Conclusion

In conclusion, the paper proposes a modified version of an existing domain adaptation technique known as MEDA and introduces evolutionary computation concepts to it in an attempt to address two of the major concerns regarding modern domain adaptation techniques: getting stuck at local optima and single classifier overfitting. The proposed method P-MEDA uses a population of classifiers, instead of focusing on developing a single classifier, to avoid local minima and get rid of the overfitting problem (better generalization). From the results, it has been verified that the proposed technique is able to outperform the state-of-the-art domain adaptation techniques.

References

[1] Jindong Wang, Wenjie Feng, Yiqiang Chen, Han Yu, Meiyu Huang, and Philip S Yu. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 402–410, 2018.

[2] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE conference on computer vision and pattern recognition*, pages 2066–2073. IEEE, 2012.

[3] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 22(14):e49–e57, 2006.

[4] Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A Ryan, Margie L Homer, and Ramón Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.

[5] Jafar Tahmoresnezhad and Sattar Hashemi. An efficient yet effective random partitioning and feature weighting approach for transfer learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 30(02):1651003, 2016.

[6] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *2011 international conference on computer vision*, pages 999–1006. IEEE, 2011.

[7] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.