Project in Advanced Machine Learning
Transfer Learning and Optimal Transport

Jefin Paul and Annsana Baby[1]

Abstract

In this project, we explored two domain adaption technique such as Subspace Alignment and Optimal Transport focusing on the Office/Caltech dataset comprising of two domains: Webcam and DSLR. The Source is Webcam and target is DSLR, Subspace alignment method projects the Source and target samples into subspaces spanned by their principal components, minimizing the shift between eigenvectors and their accuracy is compared using 1NN classifier. Then we presents a domain adaptation process using entropic regularized optimal transport. Focusing on the Webcam to DSLR transition, the cost matrix is calculated and normalized for uniformity. The coupling matrix guides data alignment, effectively minimizing the domain shift. Source data is transformed into the target domain.

1. About the Dataset

In this project we used the dataset Office/Caltech dataset where classification task is to assign an image to a class based on its content.

There were total four different domains in this dataset coming from GoogleNet, CaffeNet and surf, we chose GoogleNet for this particular project since it had the most number of data available, the four domains are Caltech, DSLR, Webcam and Amazon

For our project we use two domain, we consider the Source as Webcam(W) and the target as DSLR(D).

After loading the data of webcam and DSLR, we obtained its Data matrix X_source and X_target which are the respective features for webcam and DSLR. Then we obtained the vector of labels y_source and y_target which are the vector of labels of webcam and DSLR respectively.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

X_source and X_target are obtained after performing the z-score normalization is a pre-processing step and is done to rescale features so that they have the properties of a standard normal distribution with a mean of 0 and a standard deviation of 1. This allows the domain adaptation algorithms to focus on capturing domain-specific differences

2. Domain Adaptation

It focuses on transferring knowledge from one domain (source domain) to another domain (target domain) in order to improve the performance of a machine learning model on the target domain. It can be of different types where there can occur same domain but different tasks, or in some cases the tasks are same but the domains are different, this project is inspired by this, in our project we have the same task but the domains are different.

It can be tackled using various techniques. Common techniques in domain adaptation are Feature space alignment which aims to align the feature space of source and target domains. Then we can alter the importance of instances from the source domain to make them more relevant to the target domain. Then we can propagate labels from the source domain to target domain based on data similarity. One another method can be by using Adversial Training to align feature distribution between domains by using a Gradient reversal layer. Also, we can assign different weights to samples from the source and target domains during training.

However, for our project and for our specific problem we use two techniques to solve which are subspace alignment and optimal transport, in subspace alignment it aims to find a common subspace where the source and target domains can be effectively compared. By mapping the data from both domains into this shared subspace.

Whereas optimal transport, also known as Wasserstein distance provides a powerful mathematical framework for measuring the dissimilarity between probability distributions.

3. Subspace Alignment

Subspace Alignment is a new domain adaption algorithm technique, which aims to align the source and target domains in a common subspace spanned by eigenvectors.This method seeks domain invariant representation. It learns a mapping function which aligns the source subspace with the target one using the eigenvectors. In the aligned subspace, the knowledge learned from the source domain is transferred to the target domain. which improves the performance on the target data.

3.1. Approach

The fundamental idea behind Subspace alignment is to perform Principal Component Analysis (PCA) on both Source and Target data independently. By obtaining the principal components. Then we align the principal components from the source and target domains.

In this approach, we focus on aligning the source and target eigenvectors rather than minimizing the shift between the raw data. This strategy offers a significant advantage in preventing overfitting because minor modifications in the data have a limited impact on the eigenvectors.

First, we project the source data into the first *d* principal components, where *d* represents the number of selected eigenvalues. Similarly, we apply the same process to the target data. The number of eigenvalues, denoted by *d* is the hyperparameter. For different value of *d* the accuracy of K Nearest Neighbors is compared.

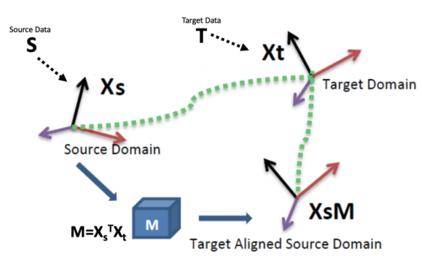

Figure 1. Principle of subspace alignment-based DA

Instead of aligning the source and target data directly, the objective here is to align their respective eigenvectors. The optimal solution for this alignment involves a linear transformation of the source eigenvector through a Alignment Matrix *M*

$$M = X_s^T X_t$$

The alignment matrix is obtained by taking the transpose of the source eigenvector and mutiplying it with the target eigenvector matrix. And then we compute Xa

$$X_a = X_s X_s^T X_t$$

M is only applied on the Source data, because it is the optimal projection to sending the source data in the target space.

Then after computing Xa we Compute the source and target projected data where we compute Sa:

$$S_a = S X_a$$

and Ta is computed as:

$$T_a = T X_t$$

Here S and T are the initial data matrices.

And Finally, based on the obtained value, the accuracy is evaluated using K Nearest Neighbors for K=1 classification. Here we have only one hyperparameter which is eigenvectors *d*.

3.2. Algorithm

We have implemented the subspace alignment as a function in Python taking in as input the initial data matrices X_source, X_target, d, y_source and y_target and it outputs the accuracy of the 1_NN classifier on the vector labels.

Algorithm 1 Subspace Alignment and 1-NN Classification

Input: Source data S, target data T, dimensionality d, source labels Y_{source}, target labels Y_{target}
$X_S \leftarrow \text{PCA}(S, d)$
$X_T \leftarrow \text{PCA}(T, d)$
Align source data in the subspace: $X_a \leftarrow X_S \times X_S' \times X_T$
$S_a \leftarrow S \times X_a$
$T_a \leftarrow T \times X_T$
$fit \leftarrow \text{1-NN.fit}(S_a, Y_{\text{source}})$
$predict \leftarrow \text{1-NN.predict}(T_a)$
$accuracy \leftarrow \text{accuracy}(Y_{\text{target}}, predict)$
Output: $accuracy$

3.3. Result for Subspace Alignment

We aim to determine the the optimal eigenvector *d*, for subspace alignment. The goal was to find the most suitable d that maximizes the accuracy of our machine learning model on the target domain.

To achieve this we employed k-fold cross-validation, dividing the source domain into k subsets for training and validation. For each d value in the predefined list of *d* values, the algorithm iterates through the k folds, then the accuracy of 1-NN classifier were obtained.

The algorithm systematically evaluates different *d* values, allowing us to identify the dimensionality that leads to the highest average accuracy across the folds.

After iterating through entire list of *d*, the algorithm outputs the best d along with their corresponding accuracy. We got the d = 30 as the best value with almost 100% accuracy.

4. Entropic regularized optimal transport

The concept of Entropic regularized optimal transport is a fundamental element in this domain adaptation project. It serves as the cornerstone for aligning two distinct domains, Webcam (W) and DSLR (D), within the Office Caltech dataset. At its core, this technique leverages the power of optimal transport to find the most efficient way to transport information from one domain to another.

4.1. Initialization of Marginal Distributions

The main step in this task is to prepare the essential components for implementing the Sinkhorn-Knopp algorithm for domain adaptation using Optimal transport. The we need to define two uniform vectors a and b which represent the marginal distributions of the source and target domains. In our case a corresponds to the source domain (Webcam) and b corresponds to the target domain(DSLR). Each vector contains n_s and n_t elements which are equal to the number of samples in the source and target domains, respectively. For a we initialize it as a uniform vector by setting each element to 1/n_s, ensuring that the sum of all elements in a equals 1. Similarly for b, we follow the same procedure setting each element equals to 1. This choice of uniform distribution signifies that at this stage, we do not favor any particular data point within the source or target domain over another

4.2. Cost matrix calculation and Normalization

The cost matrix M plays a central role in the entropic regularized optimal transport algorithm as it quantifies the distance or dissimilarity between each pair of data points from the source and target domains. We calculated M to compute distances between corresponding data points X_Source and X_target. Here we choose to calculate the Euclidean distance and result is a matrix where (i,j)-th element represents the distance between 'X_Source[i]' and 'X_target[j]'.

To ensure uniformity and consistency in our computations we normalize the obtained cost matrix 'M' by dividing each element by the maximum value present in the matrix. This normalization step scales the the distances between different data points to a range between 0 and 1 and ensures the cost matrix 'M' is effectively integrated into the Sinkhorn-Knopp algorithm.

4.3. Coupling Matrix Computation

The coupling matrix 'γ' essentially represents the optimal transport plan which dictates how data points from the source domain should be transported to the target domain. To achieve this we made use of the Python Optimal Transport (POT) library. The following equation encapsulates the entire process.

$$\gamma = \text{ot.sinkhorn(a,b,M,rege)}$$

The entropic regularization parameter 'rege' controls the degree of regularization applied during the optimization process. A smaller value of 'rege' leads to a more sparse and precise transport plan. We computed the 'γ' by calling 'ot.sinkhorn' function. This matrix contains the transportation plan that aligns the source and target domains, ensuring that the domain shift is minimized.

4.4. Source Data Transformation to Target Domain

Here we focused on the practical application of the optimal coupling matrix 'γ' that we obtained in the previous step. The objective here is to transport the data points from the source domain to the target domain creating a transformed source dataset('Sa') that shares the same features as target domain. The process of transforming the source data to the target domain is achieved through a simple matrix multiplication. We took the obtained coupling matrix 'γ' and multiply it with the target data. The resulting matrix contains the source data points that have been adapted to the target domain's feature space thus enabling the source and target data to share the same feature representation. By aligning the features we ensure that the model is trained on the source data can effectively apply its learned knowledge to the target data.

4.5. Evaluation of 1-NN classifier before and after adaptation

We evaluated the effectiveness of domain adaptation in enhancing classification performance through the utilization of the algorithm, 1-Nearest Neighbour(1-NN) classifier. Our primary goal is to access how well the adaptation process, using optimal transport-based transformation('Sa') improves the accuracy of predicting labels in the target domain. We first run the 1-NN classifier on the source data before any adaptation. The classifier is trained using the source data features and corresponding labels. Then we applied this trained classifier to make predictions on the target domain. Then we introduced the transformation of the source data into the target domain which was achieved by optimal transport process. we then applied the 1-NN classifier to the transformed data and made predictions on the target domain. The results clearly indicates the improvement in the classification accuracy after adaptation. Before adaptation, the 1-NN classifier achieved an accuracy of 76.61%. However after applying the optimal transport based adaptation, the accuracy reaches 100%

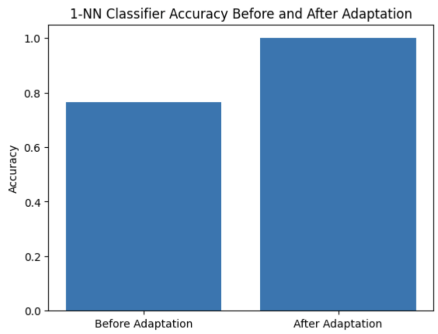

Figure 2. Accuracy before and after adaptation

This indicates a significant enhancement in the classifier's ability to accurately predict labels in the DSLR domain. These results underscore the power of domain adaptation , particularly through optimal transport in mitigating domain shift and improving model performance.

5. Conclusion

In this study, we explored the challenges of domain adaptation, we focused on two domain adaption technique such as Subspace Alignment and Optimal Transport, where the task remains consistent but the domain differ between the source and target.

For the first part of the project through the application of subspace alignment, where the goal was to find a common subspace to compare the source and the target value to transfer the knowledge from the source to target, we successfully aligned the source and target domains in a shared subspace spanned by eigenvectors. Later computed the transformed version of Source and Target to align in the subspace and found the best d(d=30)or the eigen vectors and compared them with the classifer 1-NN which gave the maximum accuracy.

For the second part of the project we outlined the key steps of our domain adaptation project utilizing the entropic regularized optimal methodology. We commenced by establishing uniform marginal distributions for the source and target domains ensuring an unbiased starting point for the adaptation process. The subsequent calculation and normalization of the cost matrix enabled us to quantify the dissimilarity between data points. The optimal transport plan, represented by the coupling matrix 'γ,' played a central role in achieving domain adaptation. Leveraging the Python Optimal Transport library, we effectively transported data points from the source to the target domain. The regularization parameter offered control over the degree of regularization. The results assessed through a 1-Nearest Neighbour (1-NN) classifier, demonstrated the substantial impact of domain adaptation. Accuracy increased from 76.61% before adaptation to 100% after adaptation, underlining the effectiveness of our approach enhancing model performance.

References

[1] https://pythonot.github.io/all.html

[2] https://github.com/PythonOT/POT/blob/master/ot/bregman.py

[3] https://openaccess.thecvf.com/content_iccv_2013/papers/Fernando_Unsupervised_Visual_Domain_2013_ICCV_paper.pdf

[4] https://arxiv.org/abs/1812.11806