Advanced Machine Learning: Transfer Learning and Optimal Transport

Ismail BACHCHAR [1]

Abstract

In this work, we implement and compare two domain adaptation algorithms with the objective of studying their performance in how, using the 1-Nearest Neighbors classifier, we could classify images of a target domain while being trained on images from a source domain before and after the application of domain adaptation methods.

All the code presented in this work is written in Python, and the data is available in the Code and Data section.

Code source: Advanced Machine Learning: Transfer Learning and Optimal Transport

Keywords— Transfer Learning, Optimal Transport, Domain Adaptation

1. Subspace Alignment

In this first part, we implement a subspace alignment algorithm of the source and target data domains. This method aims to project the labeled source and unlabeled target samples $S(n_s \times D)$ and $T(n_t \times D)$ in two subspaces spanned by their principal components so that the divergence between the two domains is minimized. It proceeds as follows:

- 1- Compute d principal components having the highest variance and denote them by $X_s \in \mathbb{R}^{D \times d}$ for the source and $X_t \in \mathbb{R}^{D \times d}$ for the target.
- 2- Define the alignment matrix $M = X_s^T X_t$ and compute $X_a = X_s X_s^T X_t$.
- 3- Compute the source and target projected data $S_a = SX_a$ and $T_a = TX_t$, where S and T are the initial data matrices.
- 4- Run a 1-NN classifier on S_a and make predictions on T_a.

[1]Jean Monnet University, Saint-Etienne, France. Ismail BACHCHAR <isbachchar@gmail.com>.

1.1. Experiment Results

For this particular algorithm, we want to study how 1NN classifier performance would change when being trained on data from the source domain and evaluated on a test set from the target domain without any subspace alignment between the two domains and then after the subspace alignment being applied to the data from both domains. This subspace alignment algorithm has only one parameter, d: the number of principal components to keep when doing PCA. Therefore, we also study the influence of this parameter.

In what follows, we present several results showing different used values of the d parameter, the 1NN accuracy in the original space (trained on raw source data and evaluated on raw target data) and the alignment space, and the source and target domains; the first two rows are the DSLR and Webcam domains from the CaffeNet dataset, the second two rows from the GoogleNet, and the last two rows are from the Surf dataset.

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.993
webcam	dslr	0.994	1
dslr	webcam	0.997	0.990
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.800
webcam	dslr	0.561	0.892

Table 1. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=100).

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.993
webcam	dslr	0.994	1
dslr	webcam	0.997	0.986
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.844
webcam	dslr	0.561	0.917

Table 2. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=80).

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.997
webcam	dslr	0.994	1
dslr	webcam	0.997	0.993
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.847
webcam	dslr	0.561	0.892

Table 3. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=60).

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.993
webcam	dslr	0.994	1
dslr	webcam	0.997	0.983
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.810
webcam	dslr	0.561	0.796

Table 4. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=30).

Source	Target	1NN before	1NN after
dslr	webcam	0.963	1
webcam	dslr	0.994	1
dslr	webcam	0.997	0.986
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.756
webcam	dslr	0.561	0.783

Table 5. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=20).

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.990
webcam	dslr	0.994	0.975
dslr	webcam	0.997	0.963
webcam	dslr	0.981	0.981
dslr	webcam	0.441	0.675
webcam	dslr	0.561	0.701

Table 6. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=10).

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.678
webcam	dslr	0.994	0.771
dslr	webcam	0.997	0.841
webcam	dslr	0.981	0.752
dslr	webcam	0.441	0.614
webcam	dslr	0.561	0.573

Table 7. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=5).

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.573
webcam	dslr	0.994	0.548
dslr	webcam	0.997	0.359
webcam	dslr	0.981	0.248
dslr	webcam	0.441	0.203
webcam	dslr	0.561	0.306

Table 8. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With d=2).

The first notice we see from the above tables is that after applying the subspace alignment algorithm and learning the classifier on that space, the performance is always better, even if it's sometimes just slightly better. We also see that for the Surf dataset -the last two rows of every table- the classifier on the original space is not performing well in comparison with itself after the subspace alignment. For example, in Table 2 the last row, the 1NN classifier has an accuracy of $.561$ before doing the subspace alignment, and suddenly this accuracy got jumped to $.917$. This means that the domain adaptation for the Surf dataset is very important, if not required, in order to do transfer learning.
Lastly, we see also that the d parameter only changes the performance when it's being reduced a lot, such as decreasing it from 100 principal components to only 2. In this case, it's expected that the 1NN performance to be worse. Also, it is worth mentioning that in our experiments, the best value of d is 80 as it gives the best accuracies for almost all the data domains.
In conclusion, the d parameter act exactly as the PCA parameter when choosing the number of components to keep and has nothing in particular to the subspace alignment, which is understandable as the PCA part in this algorithm is well isolated.

2. Entropic Regularized Optimal Transport

In this section, we do the same study as before but with a different algorithm, the entropic regularized OT, which uses the Sinkhorn-Knopp algorithm to solve the problem between two empirical distributions of data matrices $S \in \mathbb{R}^{n_s \times d}$ and $T \in \mathbb{R}^{n_t \times d}$. Its implementation can be done as follows:

- 1- Define two uniform vectors a and b that have the size equal to n_s and n_t, respectively.
- 2- Calculate the cost matrix M where an element with index (i, j) is a distance between the example $x_i \in S$ and the example $y_j \in T$. Then M is normalized to its maximum value.
- 3- Use the POT functions to fit S to T by learning the coupling matrix γ.
  $\gamma = ot.sinkhorn(a, b, M, reg)$ where reg is the entropic regularization parameter.
- 4- Transport the points from S to T using the coupling matrix γ as follows: $S_a = \gamma T$
- Run a 1-NN classifier on S_a and make predictions on T.

2.1. Experiment Results

As in the previous section, the objective is to compare the performance of a 1NN classifier trained on the raw source data and evaluated on the raw target before and after applying the entropic regularized OT algorithm in order to make domain adaptation. We also study in this part the impact of the regularizer parameter reg in order to see its influence on the performance of the 1NN classifier.

In what follows, we present several tables showing different used values of the reg parameter, the 1NN accuracy in the original space and the alignment space, and the source and target domains; the first two rows are the DSLR and Webcam domains from the CaffeNet dataset, the second two rows from the GoogleNet, and the last two rows are from the Surf dataset.

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.885
webcam	dslr	0.994	0.987
dslr	webcam	0.997	0.980
webcam	dslr	0.981	0.987
dslr	webcam	0.441	0.505
webcam	dslr	0.561	0.554

Table 9. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=0.25)

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.966
webcam	dslr	0.994	0.987
dslr	webcam	0.997	0.983
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.420
webcam	dslr	0.561	0.529

Table 10. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=1)

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.953
webcam	dslr	0.994	0.968
dslr	webcam	0.997	0.980
webcam	dslr	0.981	0.987
dslr	webcam	0.441	0.427
webcam	dslr	0.561	0.484

Table 11. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=10)

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.969
webcam	dslr	0.994	0.936
dslr	webcam	0.997	0.976
webcam	dslr	0.981	0.981
dslr	webcam	0.441	0.464
webcam	dslr	0.561	0.618

Table 12. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=50)

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.929
webcam	dslr	0.994	0.924
dslr	webcam	0.997	0.986
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.410
webcam	dslr	0.561	0.541

Table 13. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=100)

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.925
webcam	dslr	0.994	0.930
dslr	webcam	0.997	0.990
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.478
webcam	dslr	0.561	0.510

Table 14. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=200)

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.973
webcam	dslr	0.994	0.930
dslr	webcam	0.997	0.980
webcam	dslr	0.981	0.987
dslr	webcam	0.441	0.529
webcam	dslr	0.561	0.580

Table 15. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=500)

Source	Target	1NN before	1NN after
dslr	webcam	0.963	0.953
webcam	dslr	0.994	0.981
dslr	webcam	0.997	0.980
webcam	dslr	0.981	0.994
dslr	webcam	0.441	0.468
webcam	dslr	0.561	0.497

Table 16. 1NN Accuracy Comparison Between Data from Different Domains in The Original Space (1NN before) and After Subspace Alignment (1NN after) (With reg=1000)

In contrast to the first algorithm, this entropic regularized OT method doesn't improve the accuracy of the 1NN classifier in all the datasets, especially the Surf one (the last two rows in every table). Even after tuning the reg parameter, ranging in $[0.25, 1, 10, 50, 100, 200, 500, 1000]$, nothing is being changed, which seems the algorithm is not able to transport the source data points to any place near the target data points. We got a slightly improved score from 0.561 to 618 in Table 12 (last row) with the $reg = 50$.
To conclude, this algorithm's results were disappointing and not expected at all. In almost all cases, the 1NN performance in the original space is better than after the transportation of source data.

Code and Data

The code implementation of this project is available on GitHub (Advanced Machine Learning Bandits-Reinforcement Learning). Presented in a reproducible manner so that similar results could be re-obtained. The repository contains one Jupyter Notebook containing the code implementation of the two algorithms and experiments generator.
Considering the data, we provide the Office/Caltech dataset, a well-known dataset in the domain adaptation field. It contains images' extracted FTS -Fourier Transform Spectrometer- features.

3. Conclusion

In this work, we implemented from scratch and compared two domain adaptation algorithms. This comparison was based on the availability of improving the accuracy of the 1NN classifier from applying directly to the raw data in the original space to applying to data in the alignment space. As a result of this comparison, we found out that the subspace alignment method is way better than the entropic regularized OT, at least for the given experiments we did, and the data was used.