Università di Genova
DIPARTIMENTO
INFORMATICA, BIOINGEGNERIA,
ROBOTICA E INGEGNERIA DEI SISTEMI

Computer Science Workshop
PhD program in Computer Science and Systems Engineering

Trainingless datasets distance

An efficient method for transfer learning

Introduction

A common context in deep learning is the following: given a new problem with limited resources we would like to find a simple yet efficient way to deal with it. One typical answer to these issues can be found in transfer learning. Surely this process has been shown to be very effective in many situations. An alternative could be to fine-tune a model M' pre-trained on a dataset *closer* to the task we want to solve.

Main Objectives

- Develop a pipeline in which, given as input a new dataset, we receive as output a vector measuring the distance between the dataset and a bag of previously seen datasets.
- A key aspect in this project is efficiency. Is not desirable to spend too much time in determine if a new problem could be similar to a previous one. So we would like to develop a pipeline that does not involve a training process.

Previously some other works like [1][2] proposed a way to compute datasets similarity between simple datasets. Still, this methods could not handle efficiently large datasets, particularly when dealing with images.

Materials and Methods

In machine learning we usually try to solve a problem $\mathcal{P}$ by having a set of data points.
We can consider a *bag* of d datasets $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_d\}$.
A dataset $\mathcal{D}_i$ is made by n_i images and the corresponding n_i labels:

$\mathcal{D}_i = (X_i, Y_i)$ where
$X_i \subset \mathbb{R}^{n_i \times p}$ and $Y_i \subset \mathbb{R}^{n_i \times 1}$ and where p is the #pixel
Usually we would like to find the distribution $\mathcal{P}_i$ associated to the i-th dataset. In practice we can find an empirical version $\hat{\mathcal{P}}_i$ of it:

$$\hat{\mathcal{P}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \delta(X_{ij}, Y_{ij}))$$

we want to obtain a distance measure between datasets that is based on these empirical distributions:

$$d(\mathcal{D}_i, \mathcal{D}_j) \approx d(\hat{\mathcal{P}}_i, \hat{\mathcal{P}}_j)$$

Our pipeline is made by four phases:

1. Features extraction: map every data point from the image space to a different, encoded space. We want to use a map function Φ:
   $\Phi : \mathbb{R}^p \to \mathbb{R}^f$
   where f is the size of the feature produced by the map Φ.
   This can be done by using a pre-trained convolutional model like ResNet, DenseNet. The output produced is a $(d \times n_i \times f)$ matrix.

2. Aggregation: instead of using the $(n_i \times f)$ matrix we consider separately its columns and produce a b bins histogram for every feature. The range r_{f_j} considered for the j-th histogram is:

   $$r_{f_j} = [0, \max_{f_j}\{P_{p\%}\}]$$

   i.e. between zero and the highest value between all the datasets that can cover the p percentile of the dataset. The output produced is a $(d \times f \times b)$ matrix.

3. Kernel PCA: reduce the dimensionality of our data by using a kernel PCA technique, mapping every dataset in a new space with c components. The output is a $(d \times c)$ matrix.

4. Distance in PCA space: compute a distance measure between them. The output of the last phase is a $(d \times d)$ matrix, expressing the distance between every dataset.

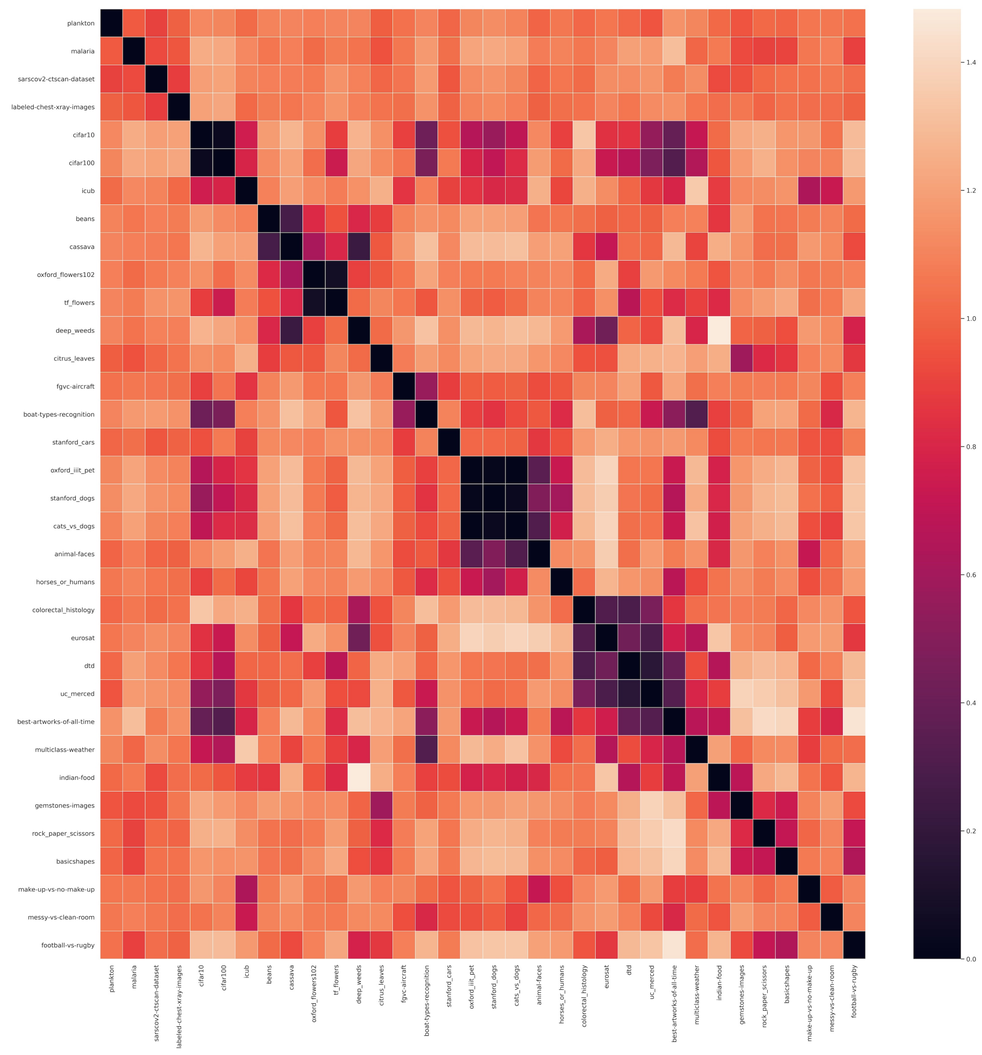

Figure 1: Heatmap representing the distance between different datasets

Conclusions

Our preliminary results seem to show a clustering between similar datasets. This is shown in Figure 1 where we can see a clustering between: medical datasets, general purpose datasets, flower datasets, animal datasets and texture datasets.

Forthcoming Research

As we can see in Figure 1, not all the cluster are well separated. This could be due to a non-optimal parameters configuration of our pipeline. Different parameters are involved in every phase:

1. Features extraction: width and height of the input image, pre-trained model
2. Aggregation: number of bins, percentile
3. Kernel PCA: kernel and its parameters if needed
4. Distance: which distance metric

All the parameters should be carefully tuned to give us a better representation of our data

References

[1] David Alvarez-Melis and Nicoló Fusi. Geometric dataset distances via optimal transport. *CoRR*, abs/2002.02923, 2020.

[2] Nikolaj Tatti. Distances between data sets based on summary statistics. *Journal of Machine Learning Research*, 8(5):131–154, 2007.

Acknowledgements

We would like to thank MaLGa Center, DIBRIS, Università di Genova and the financial support of the European Research Council (grant SLING 819789), the AFOSR projects FA9550-18-1-7009, FA9550-17-1-0390 and BAA-AFRL-AFOSR-2016-0007 (European Office of Aerospace Research and Development), and the EU H2020-MSCA-RISE project NoMADS - DLV-777826.

CONTACTS

Paolo Didier Alfano
paolodidier.alfano
@edu.unige.it

Vito Paolo Pastore
Vito.Paolo.Pastore
@edu.unige.it

Francesca Odone
Francesca.Odone@unige.it

Lorenzo Rosasco
lorenzo.rosasco@unige.it