ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

Feature selection for transfer learning using particle swarm optimization and complexity measures

Guillermo Castillo-García[1], Laura Morán-Fernández[2], Verónica Bolón-Canedo[2] *

1- Universidad Internacional Menéndez Pelayo, Spain

2- CITIC, Universidade da Coruña, A Coruña, Spain

Abstract. Particle Swarm Optimization is an optimization algorithm that explores a search space guided by a fitness function in order to find a good solution. We apply it to perform feature selection for domain adaptation. Usually, classification error is used in the fitness function to evaluate the goodness of subsets of features. In this paper, we propose to employ complexity metrics instead, as we assume that reducing the complexity of the problem will lead to good results while being less computationally demanding and independent from the classifier used for testing. We found out that our method is indeed faster and selects fewer features, obtaining competitive classification accuracy results.

1 Introduction

Feature selection [3] is a machine learning technique used to reduce the dimensionality of a dataset. The goal is to select only the relevant features for our predictive model, leaving out the ones that are redundant or do not provide useful information. Transfer learning, on its side, is a technique that aims to use acquired knowledge from an existing source domain to improve learning performance in a different, yet similar target domain. In our case, we deal with problems where there is a common feature space in both the source and target datasets. The task is common, but the source and target domains have different distributions. This case of transfer learning is known as *domain adaptation*.

The focus of this study is to use feature selection for domain adaptation, with the objective of identifying a common subset of features that optimizes classification performance in both the source and target datasets. Different approaches have been proposed to deal with this problem, among which is using Particle Swarm Optimization (PSO) [4], an algorithm that mimics the behaviour of a flock of birds, where multiple particles move around the search space trying to find a good solution, guided by a fitness function. In particular, there have been in the literature some attempts to use PSO approaches for domain adaptation [6, 2], although these works used the classification accuracy to evaluate the goodness of the subsets of features, which is a very time-consuming approach.

Differently from other works, in this paper we propose to employ complexity metrics in the fitness function instead of classifiers. The reason behind using data

*This work has been supported by the Spanish Ministerio de Economía y Competitividad (Grant PID2019-109238GB-C22) and by the Xunta de Galicia (Grants ED431C 2018/34 and ED431G 2019/01) with part of European Union ERDF funds.

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

complexity metrics [5] is based on the assumption that a good choice of features should lower the complexity of the data, and will therefore lead to competitive performance results. Apart from that, employing complexity metrics provides results independent of the classifier used in a subsequent phase, and should decrease the computational time.

2 Materials and methods

2.1 Feature-based domain adaptation

Given a source domain D^S associated to a source task T^S and a target domain D^T associated to a target task T^T, transfer learning can be defined as the process of using the related information from D^S and T^S in trying to improve a target predictive function $f^T(\cdot)$, where $D^S \neq D^T$ or $T^S \neq T^T$.

Feature-based transfer learning is focused on finding a feature representation that simultaneously achieves significant predictive accuracy in both the source and target domains, as well as reducing the difference between data distributions. In particular, in this work we will try to obtain common meaningful features for both domains by reducing the marginal distribution between them, which is known as feature-based domain adaptation.

2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [4] is an algorithm for solving optimization problems inspired by the behaviour of a flock of birds; it sets a swarm of particles which explore the search space in parallel. Each particle is a solution candidate, and consists of a position and velocity (or momentum). The value of each particle is computed using a fitness function, which guides them to a good solution.

It was originally proposed to solve continuous problems, therefore we opted to use Sticky Binary Particle Swarm Optimization (SBPSO) [6], which redefines the concept of momentum making it appropriate for binary problems.

2.3 Proposed fitness function

Based on the work by Nguyen et al. [6], we will compare the use of classifiers in the fitness function with our proposal based on complexity measures. The fitness function is defined as follows:

$$Fitness = sw * srcPen + tw * tarPen + stw * diffST \tag{1}$$

where sw, tw and stw are weights, $srcPen$ and $tarPen$ are the penalty on source and target data, and $diffST$ measures how different the marginal distributions of each data partition are, using Maximum Mean Discrepancy.

For computing the penalty on source and target data, we will use the two approaches mentioned, using the accuracy of different classifiers, and using data complexity metrics [5], the following being the ones we will use:

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

- **k-Nearest Neighbors (kNN)** is classifier which uses proximity to make predictions about the class to which an individual data point belongs.
- **Support Vector Machine (SVM)** aims to find hyperplanes that separate the classes by trying to find, for each pair of classes, the plane that maximizes the distance between the data points of both classes.
- **Naive-Bayes (NB)** is a probabilistic classifier based on Bayes' theorem.
- **F1** is a complexity measure that returns the maximum Fisher's discriminant ratio over all the features, where the higher the F1, the easier the problem is.
- **F2** measures the amount of overlap of the bounding boxes of two classes, and it is zero if there exists at least one feature for which the values of the two classes do not overlap. Therefore, high values indicate overlap and thus complex classification tasks. In our case, instead of multiplying, we use the sum, which makes it the length instead of the volume. The side effect of employing the product is that the value of this measure decreases drastically as dimensionality increases, which is not appropriate for our fitness function, as it would make the last component useless.
- **F3** returns the maximum (individual) feature efficiency, i.e. the largest fraction of points distinguishable with only one feature. This complexity measure considers only separating hyperplanes perpendicular to the features axes so, even for a linear problem, it can be less than 1 if the optimal separating hyperplane is oblique. The higher the F3, the easier the problem is.

2.4 Overall algorithm

Figure 1 shows the general structure of the proposed methodology. First, we have Src_{train} and Tar_{train} which are the source and target data for training. Both of them are used in the feature selection process, using SBPSO, which is marked in blue. This gives a common feature space for both source and target data, which is applied to the test data (Src'_{test} and Tar'_{test}). A classifier is trained using Src'_{test}, and then Tar'_{test} is used for testing on the trained classifier, producing the corresponding classification performance.

2.5 Datasets

For testing the proposed approaches, we will use two different datasets. The first one is Gas Sensor [7], which consists of 13,910 instances with 127 features and six classes to predict. The dataset is split into 10 batches. We used the first as source dataset, and the remaining ones as target dataset, which results in nine domain adaptation cases. The second dataset is the TOX-171 microarray dataset [1], which has 171 instances, 5748 features, and four different classes. In each run of the algorithm, the dataset is split into two parts, making the source

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

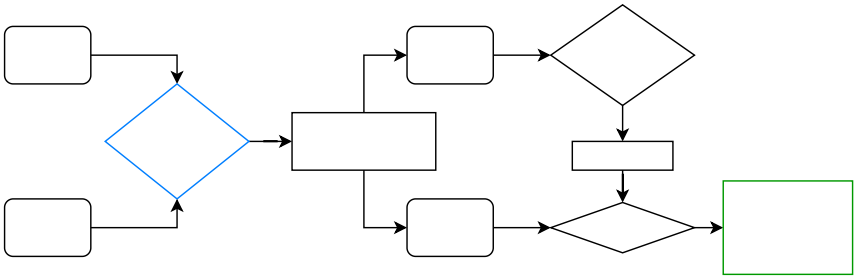

Fig. 1: Diagram of feature selection for domain adaptation using SBPSO

and target datasets. To preprocess the data, we normalized the values of the features. For Gas Sensor, as it consists of different splits of the same dataset, we normalized all the batches together. In both datasets, we used 70% of the data for training and 30% for testing.

3 Experimental results

This section presents the results obtained by testing the two types of penalty in the fitness function in two different datasets for transfer learning. First, we carried out a comparison of the time per iteration required when using each fitness function, using only the first two components, as the third one is common for all of them (Table 1). For this test, we executed 50 iterations for each dataset and fitness function, and the values shown are the mean obtained. As expected, the fitness functions using complexity metrics are notably faster than the ones using classifiers. It stands out how slow SVM and NB were compared to the rest. This fact prevented their use for the classification experiments, as they required too much time to be good choices for the PSO algorithm.

Table 1: Mean time (in seconds) per iteration for each dataset and fitness function. Lowest results are marked in bold.

	F1	F2	F3	kNN	SVM	NB
Gas Sensor	**0.0027**	0.0041	0.0032	0.2998	0.8714	8.6793
TOX-171	0.0277	0.0402	**0.0273**	2.7219	10.8670	53.4256

Then, to check the impact of the different fitness functions, we studied the classification performance obtained by three different classifiers. Every experiment was run 25 times. After testing multiple combinations of parameters, we decided to use a swarm size of 50, maximum life of a particle (the maximum number of iterations it can keep the same value) of 40, and a maximum of 3000 iterations, stopping if it has not improved for 300 iterations.

For each variation of the algorithm, we set four parameters: the weights for each component of the fitness function (sw, tw, stw) and a minimum percentage of features to select ($minFeatures$), in order to counter the tendency to select

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

as few features as possible with F2 and F3 metrics. The values that performed better and were therefore used in the test for the classification comparison are presented in Table 2.

Table 2: Selected parameters for each dataset and fitness function

Dataset	Fitness function	sw	tw	stw	minFeatures
Gas Sensor	F1	0.2	0.2	0.6	0.25
Gas Sensor	F2	0.1	0.1	0.8	0.2
Gas Sensor	F3	0.05	0.15	0.6	0.25
Gas Sensor	kNN	0.1	0.9	0	0.25
TOX-171	F1	0.6	0.2	0.2	0.1
TOX-171	F2	0.025	0.025	0.95	0.2
TOX-171	F3	0.3	0.3	0.4	0.1
TOX-171	kNN	0.5	0.5	0	0.1

Table 3 shows the mean accuracy obtained for each fitness function and each dataset in all the runs (the results for Gas sensor are the means of the results of all batches), and for each classifier. The last column shows the mean percentage of features selected.

Table 3: Mean classification accuracy and percentage of features selected results. The best result for each dataset and classifier is marked in bold.

Dataset	Fitness	kNN Acc.	SVM Acc.	NB Acc.	% Features
Gas Sensor	F1	0.406	0.395	0.381	**17.31**
Gas Sensor	F2	0.295	0.342	0.346	20.47
Gas Sensor	F3	0.383	0.375	0.379	26.60
Gas Sensor	kNN	**0.548**	**0.400**	**0.412**	41.11
TOX-171	F1	0.542	0.532	0.543	50.10
TOX-171	F2	0.540	0.527	0.537	**37.60**
TOX-171	F3	**0.548**	**0.538**	**0.550**	49.18
TOX-171	kNN	0.545	0.523	0.543	49.91

As we can see, for the Gas Sensor dataset, the option which produced the best results was using kNN classifier in the fitness function, and kNN as the subsequent classifier, which was expected as the features selected were the best possible ones for this learning algorithm. However, for the other classifiers, the results obtained with complexity measures are competitive (e.g. for SVM, there is a difference of 0.005 between using F1 and kNN in the fitness function). With regard to TOX-171 dataset, the best results were achieved when using complexity metrics in the fitness function, with the added benefit of this type of methods to be classifier-independent and less computationally demanding.

Regarding the number of selected features, it seems that using complexity metrics provides an advantage, as they generally select less features. This leads to simpler datasets, with a more compact subset of relevant features which fa-

ESANN 2022 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 5-7 October 2022, i6doc.com publ., ISBN 978287587084-1. Available from http://www.i6doc.com/en/.

cilitates interpretability of the data.

4 Conclusions

Transfer learning is a recent trend in machine learning and a prolific field of research. In particular, in this paper we focused on domain adaptation, with the goal of obtaining a common representation of meaningful features both for the source and target data. In particular, we proposed the use of PSO to find the relevant features. Instead of using classification performance to evaluate the subsets of attributes, we introduced a new fitness function which uses data complexity metrics for this task. Our hypothesis was that a subset of data with the correct features results in a less complex dataset.

After carrying out experiments over two datasets suitable for transfer learning evaluation, we demonstrated the competitiveness of our proposed approach. The use of complexity measures in the fitness function led to a reduction in the computational time and the number of features selected, without showing degradation in the classification accuracy. Moreover, an added advantage of our proposal is that it is classifier-independent, not conditioning the choice of a given classifier in a posterior classification stage.

References

[1] Arizona State University. Accessed April 2022. Feature selection datasets. URL: `https://jundongl.github.io/scikit-feature/datasets.html`.

[2] H. Dhrif, V. Bolón-Canedo, and S. Wuchty. "Gene Subset Selection for Transfer Learning using Bilevel Particle Swarm Optimization". In: *2020 19th IEEE Int. Conf. on Machine Learning and Applications (ICMLA)*. IEEE. 2020, pp. 1317–1323.

[3] I. Guyon et al. *Feature extraction: foundations and applications*. Vol. 207. Springer, 2008.

[4] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Proc. of ICNN'95 Int. Conf. on Neural Networks*. Vol. 4. IEEE. 1995, pp. 1942–1948.

[5] A.C. Lorena et al. "How Complex is your classification problem? A survey on measuring classification complexity". In: *ACM Computing Surveys (CSUR)* 52.5 (2019), pp. 1–34.

[6] B.H. Nguyen, B. Xue, and P. Andreae. "A particle swarm optimization based feature selection approach to transfer learning in classification". In: *Proc. of Genetic and Evolutionary Computation Conf.* 2018, pp. 37–44.

[7] A. Vergara et al. "Chemical gas sensor drift compensation using classifier ensembles". In: *Sensors and Actuators B: Chemical* 166 (2012), pp. 320–329.