Unsupervised Transfer Learning for Time Series via Self-Predictive Modelling

Witali Aswolinskiy[1] and Barbara Hammer[2]

[1] Research Institute for Cognition and Robotics - CoR-Lab,
[2] CITEC Center of Excellence, Bielefeld, Germany

Abstract. Real-world machine learning applications must be able to adapt to systematic changes in the data, e.g. a new subject or sensor displacement. This can be seen as a form of transfer learning, where the goal is to reuse the old (source) model by adapting the new (target) data. This is a challenging task, if no labels for the target data are available. Here, we propose to use the structure of the source and target data to find a transformation from the source to target space in an unsupervised manner. Our preliminary experiments on multivariate time series data show the feasibility of the approach, but also its limits.

Keywords: domain adaptation, transductive transfer lerning, time series classification, predictive modelling, echo state networks

1 Introduction

In data-driven machine learning, a model is trained on the available training data and applied to new data. A good model must be able to extract the required information from the new data, even when systematic changes in the data distribution occur. For example, if the data contains information from sensors, a sensor might be replaced with a different calibrated one or the position of the sensors might change. Another common case is the application of the model to a new subject, e.g. in gesture recognition.

Problems of this type can be addressed by transfer learning, which considers transferring knowledge from a source domain and a source learning task to a learning task in the target domain [10]. Transfer learning has been successfully applied in diverse scenarios including robotics [2], computer vision [12] and language translation [5]. Here, we consider transductive transfer learning or domain adaptation, where the source and target domains are different, but the source and target tasks are the same [1, 9]. We assume a difference in the data distribution in the domains and that a linear transformation from source to target space is possible.

An expensive solution to this problem would be to collect a new dataset in the target space with supervised information and to train a new model. Since data labeling is often done manually by experts, this might be time consuming and impractical. A more efficient solution proposed in [8] is to gather only few labels and to find a linear transformation from the target space to the source space,

so that the original source model can be applied. Still, this solution requires sufficient supervised information to find the transformation. Here, we attempt to use only the temporal structure of the source and target data to find such a transformation.

More precisely, we propose to build self-predictive reservoir models, which capture the spatio-temporal relationships in the time series. We then use these models as surrogates for the transfer learning tasks to find a transformation from source to target space. In the following, we will formalize this methodology and present examples, where it enables a transfer without any given labeling. We will also showcase an example, where a transfer is not possible due to a limited correlation of the temporal dynamics and the supervised transfer learning task.

2 Unsupervised Transfer Learning via Self-Predictive Modeling

2.1 A hypothetical example

To illustrate the idea, let us consider a simple, hypothetical classification example, as visualized on the left side of Fig. 1. The filled circles represent the source data with known class labels. The empty circles represent the target data with unknown class labels. If we assume, that the target data is a linear transformation of the source data, we can transform the target data back to the source space preserving aligning their triangle-structure by a simple translation visualized by the arrow.

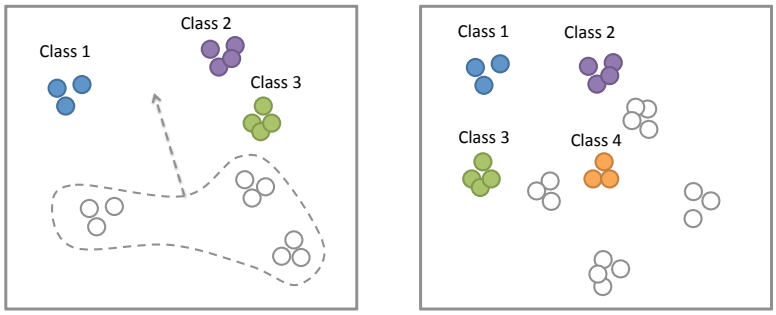

Fig. 1: Hypothetical classification examples. The filled circles represent the labeled source data and the empty circles the unlabeled target data. In the case on the left, there is enough structural information to find the correct linear transformation to the source space, as visualized my the arrow, but not in the case on the right.

The example on the right visualizes a case, where the structure of the data does not provide sufficient information to transform the target data back so that the classes can be correctly estimated. Because of the quadratic structure, four different rotations are possible, but only one of them will map the classes correctly.

As these examples show, in order to find the correct transformation from the target space back to the source space, the data must have a very distinctive structure and the classes must also have structurally distinctive properties.

Next, we present a general framework for domain adaption using the structure of the data.

2.2 Unsupervised transfer of structured data via self-predictive models

The general approach is sketched in Fig. 2. Additionally to the supervised learner (regressor, classifier, etc.), we train a self-predictive model on the source data using its structure to define a training goal, e.g. to predict the next step in time series or to predict a nearby pixel in images. No external information is used to train the predictive model. Then, we try to find a linear transformation from the target to the source data, minimizing the error of the predictive model applied to the transformed target data. A small error of a precise predictive model should indicate a good mapping. After finding such a mapping, the learner, which was trained on the source data, can be applied to the transformed target data.

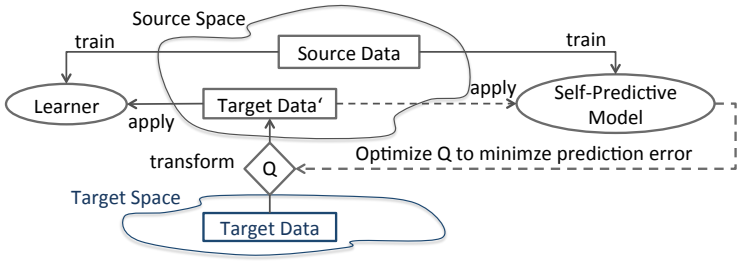

Fig. 2: Approach for unsupervised learning of a linear transformation Q from the target to source space using a self-predictive model of the structured data.

The approach is based on the assumption that the structure is distinctive enough to train a precise model and to find the correct transformation from the target to the source domain. The approach should work the better the more structural information is contained in the data. We will focus therefore on time series data, where a very prominent structuring element is available, namely the temporal progression of the observations. In the next section, we instantiate our framework for linear domain adaptation of time series.

2.3 Unsupervised transfer of time series via self-predictive reservoir networks

Given time series from source space $\mathcal{S}$ and target space $\mathcal{T}$, we want to find a linear transformation $Q : \mathcal{T} \to \mathcal{S}$ such that the temporal dynamics match those of the source domain. Our approach is visualized in Fig. 3. First, we learn a self-predictive model on the source time series.

Self-Predictive Echo State Network For the self-predictive modeling of the time series we use Echo State Networks (ESN, [4]). An ESN consists of a reservoir of recurrently connected neurons and a linear readout (cf. Fig. 3). The reservoir provides a non-linear fading memory of the inputs $\boldsymbol{u} \in \mathbb{R}^I$. The reservoir states $\boldsymbol{x} \in \mathbb{R}^N$ and the readouts $\boldsymbol{y} \in \mathbb{R}^O$ are updated according to

$$\boldsymbol{x}(k) = (1-\lambda)\boldsymbol{x}(k-1) + \lambda f(\boldsymbol{W}^{rec}\boldsymbol{x}(k-1) + \boldsymbol{W}^{in}\boldsymbol{u}(k)) \quad (1)$$

$$\boldsymbol{y}(k) = \boldsymbol{W}^{out}\boldsymbol{x}(k), \quad (2)$$

where N is the number of neurons, λ the leak rate, f the activation function, e.g. $tanh$, $\mathbf{W}^{rec} \in \mathbb{R}^{NxN}$ the recurrent weight matrix, $\mathbf{W}^{in} \in \mathbb{R}^{NxI}$ the weight matrix from the inputs to the reservoir neurons and $\boldsymbol{W}^{out} \in \mathbb{R}^{OxN}$ the weight matrix from the reservoir neurons to the readouts. $\mathbf{W}^{in}$ and $\mathbf{W}^{rec}$ are initialized randomly, scaled and remain fixed. $\mathbf{W}^{rec}$ is scaled to fulfill the Echo State Property (ESP, [4]). The necessary condition for the ESP is typically achieved by scaling the spectral radius of $\mathbf{W}^{rec}$ to be smaller than one. The readout weights $\mathbf{W}^{out}$ can be learned with ridge regression: $\boldsymbol{W}^{out} = (\boldsymbol{X}^T\boldsymbol{X} + \alpha\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{T}$, where $\boldsymbol{X}$ are the row-wise collected neuron activations, $\boldsymbol{T}$ the corresponding target values and α is the regularization strength.

We train the ESN for one-step-ahead-prediction: the readout is trained to predict the next input value $\boldsymbol{u}(t+1)$ from the current reservoir activation $\boldsymbol{x}(t)$. Thus, for a time series of length L, $\boldsymbol{X} = (\boldsymbol{x}(1); \ldots; \boldsymbol{x}(L-1))$ and $\boldsymbol{T} = (\boldsymbol{u}(2); \ldots; \boldsymbol{u}(L))$. The resulting model approximates the function $P(\boldsymbol{u}(t)) = \boldsymbol{u}(t+1)$.

Learning the linear transfer function Having determined the one-step-ahead-prediction dynamics P for the source domain, we now learn the linear transfer function $Q(\boldsymbol{u'}) = \boldsymbol{R}\boldsymbol{u'}$ on the target data, such that the source dynamics apply: $P(Q(u'(k-1)) \approx Q(u'(k))$. For this purpose, we evolve the linear transformation matrix $\boldsymbol{R}$ using the CMA-ES [3] optimization technique by minimizing the squared mean error of the previous expression.

After learning the linear transformation Q, we can map the target time series into the source space and apply the supervised task-specific learner trained in the source space.

3 Experiments

For our experiments we use ESNs not only for the self-predictive model, but also for the actual learning task. Since any other learning method suitable for

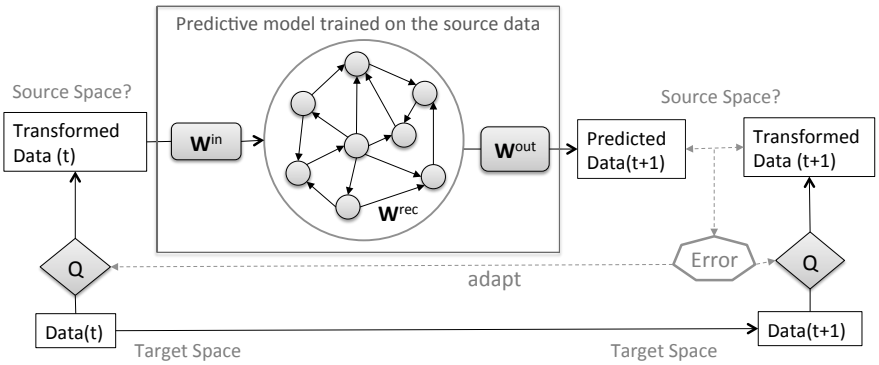

Fig. 3: Approach for unsupervised transfer of the target data through the linear transformation Q by using a self-predictive ESN trained on the source data.

time series learning would work as well, we omit the description of the learner training.

3.1 Sine wave regression

As first example we consider a synthetic, two-dimensional dataset consisting of two sine waves $u_1 = sin(0.2x), u_2 = sin(0.25x)$ with the learning target $y = u_1(t-2) + u_2(t+2)$. Fig. 4 shows the original data on the left and the same data after a random linear transformation on the right (regression goal y remains the same).

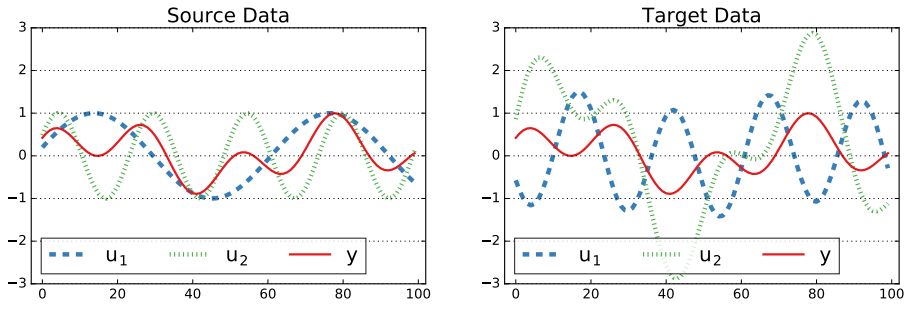

Fig. 4: Synthetic sine regression dataset with source data on the left and target data on the right.

For both the learner and the self-predictive model, a reservoir with 50 neurons was used. Fig. 5 shows the prediction error and the transfer error (the regression error of the learner on the transformed target data) during evolution of the

transformation matrix. After approximately 80 iterations of CMA-ES, the target data is successfully mapped back into the source space.

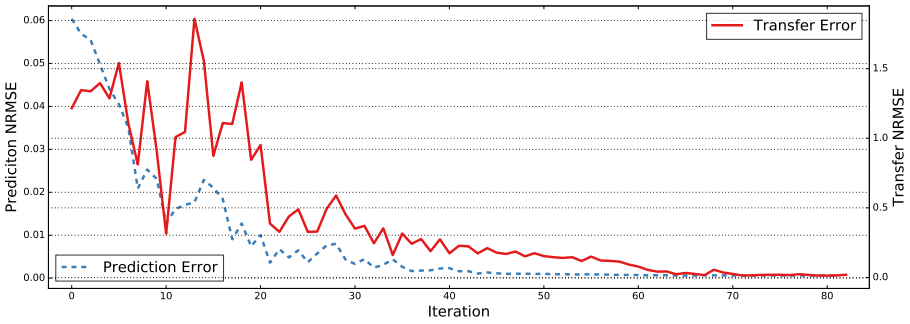

Fig. 5: Evolution of the optimization of the transformation matrix. Shown is the prediction and the regression error on the target data after transformation.

3.2 Time series classification - success

Here, we apply the approach to time series classification of multivariate time series. The character trajectories dataset [11] obtained from [6] contains 2858 pen tip trajectories (x,y,force) with lengths from 109 to 205 recorded during writing of twenty characters by a single subject. 300 sequences are used for training and the rest for testing. Again, we simulate a systematic change in the data, by transforming the test sequences through multiplication with a random matrix. We then try to discover the transformation back into the source space using a self-predictive ESN with 300 neurons trained on the original training sequences. The goal is to apply the classifier trained on the original training data to the back-transformed test sequences.

Fig. 6 (top) shows the evolution of the prediction and target classification error ('Transfer Error'). Initially, the classifier has a high error rate of about 90%. After 140 iterations, both the prediction and classification error are low.

3.3 Time series classification - failure

The uWave[7] dataset consists of 4478 three-dimensional (x,y,z) sequences of length 315 containing eight different gestures recorded from eight subjects. 200 sequences were used for training and the rest for testing. Fig. 6 (bottom) shows the evolution of the prediction and target classification error. Here, the transformation reduces the prediction error, but increases the classification error - the approach did not work for this dataset.

The hypothetical example in 2.1 showed that a transformation from the target to source space, which aligns the data distributions in the respective spaces,

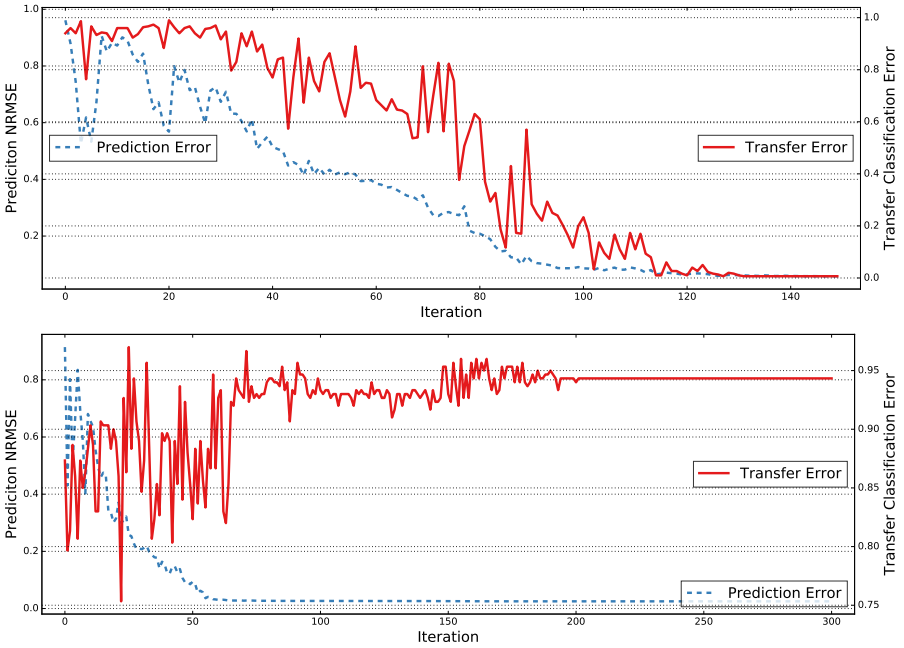

Fig. 6: Evolution of the optimization of the transformation matrix for classification of the character trajectories (top) and uWave dataset (bottom). Shown is the prediction and the classification error on the target test data after transformation.

may still be wrong semantically. The failed transfer of the uWave target data may be an example of this problem. Despite a very small prediction error, the classification results were wrong. We hypothesize that the dynamical invariant in the time series is not relevant for the classification in this case, hence a transformation based on the preservation of the temporal dynamics cannot be used as a surrogate for the learner.

Two other possible reasons for a failed unsupervised transfer are:

- Inaccurate predictive model. If the predictive model has a high prediction error on the source data (due to noisy data, badly chosen basis functions, etc.), there will be many transformations resulting in similar prediction errors on the transformed target data. Such an inexact mapping might lead to misclassifications by the learner.
- Local error minima in the transformation space. Let us again consider the triangle-data in the hypothetical example visualized in Fig. 1. The linear transformation from the the target to the source space is a simple translation. However, a translation together with half a rotation would result in only a slightly higher prediction, but a complete classification failure. Intermediate

rotation angles might lead to better classification, but would have higher prediction errors. Thus, finding a global optimum might be more important here than in other applications.

4 Conclusion

In this paper we presented an unsupervised approach for transfer learning for structured data and particularly time series. As proof of concept we evaluated the approach on one synthetic and two real-world datasets. The positive result on the synthetic and one of the real-world datasets confirm the applicability of the approach to some data sets. Further work is required to determine the conditions for a successful application of the approach and to evaluate it on real-world use cases.

References

1. Arnold, A., Nallapati, R., Cohen, W.W.: A comparative study of methods for transductive transfer learning. In: Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on. pp. 77–82. IEEE (2007)
2. Barrett, S., Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning on a physical robot. In: Ninth International Conference on Autonomous Agents and Multiagent Systems-Adaptive Learning Agents Workshop (AAMAS-ALA) (2010)
3. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary computation 9(2), 159–195 (2001)
4. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. GMD Technical Report 148, 34 (2001)
5. Johnson, M., Schuster, M., Le, Q.V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al.: Google's multilingual neural machine translation system: enabling zero-shot translation. arXiv preprint arXiv:1611.04558 (2016)
6. Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
7. Liu, J., Zhong, L., Wickramasuriya, J., Vasudevan, V.: uwave: Accelerometer-based personalized gesture recognition and its applications. Pervasive and Mobile Computing 5(6), 657–675 (2009)
8. Paaßen, B., Schulz, A., Hammer, B.: Linear supervised transfer learning for generalized matrix lvq. In: Proceedings of the Workshop New Challenges in Neural Computation 2016. No. 4 (2016)
9. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22(10), 1345–1359 (2010)
10. Torrey, L., Shavlik, J.: Transfer learning. Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques 1, 242 (2009)
11. Williams, B.H., Toussaint, M., Storkey, A.J.: Extracting motion primitives from natural handwriting data. In: International Conference on Artificial Neural Networks. pp. 634–643. Springer (2006)
12. Wu, P., Dietterich, T.G.: Improving svm accuracy by training on auxiliary data sources. In: Proceedings of the twenty-first international conference on Machine learning. p. 110. ACM (2004)