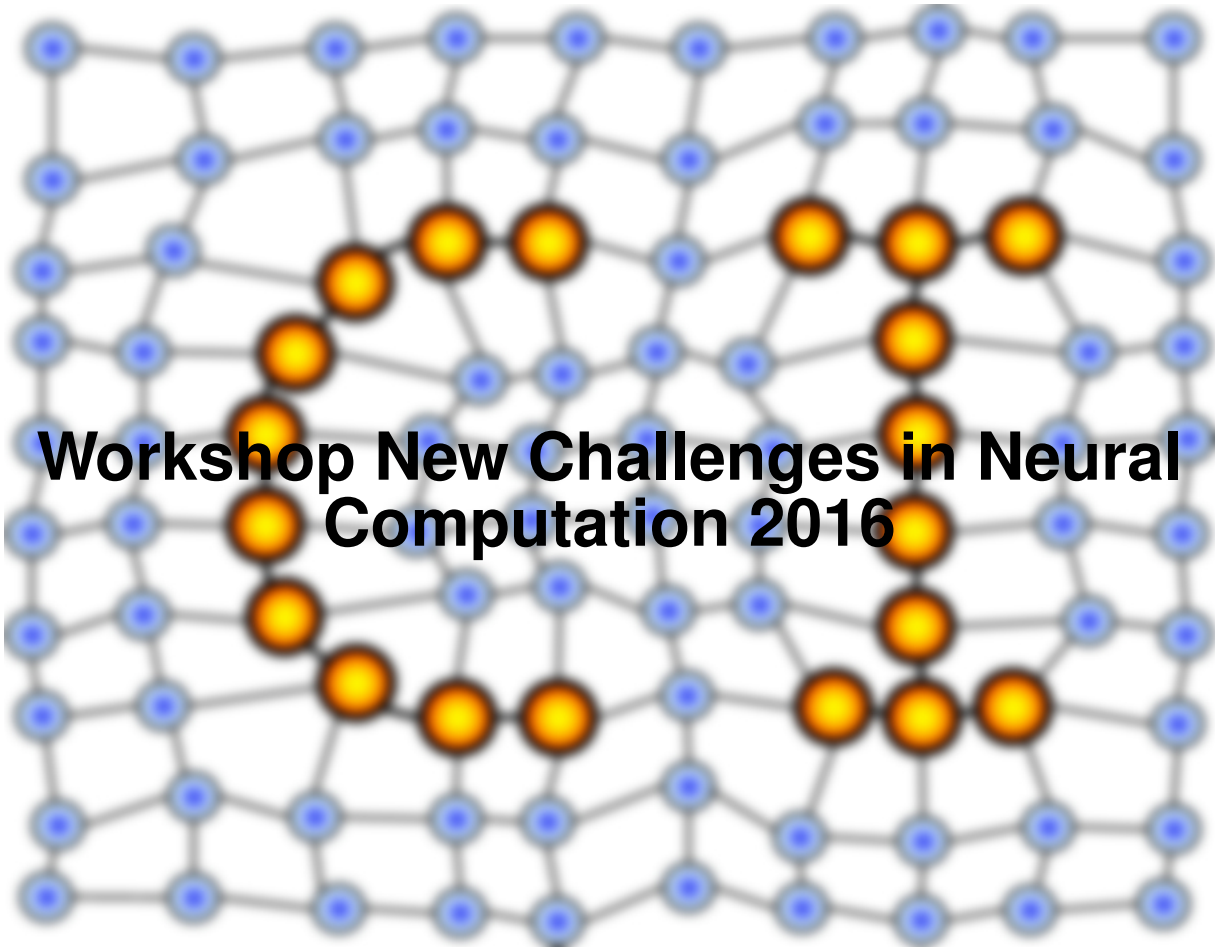
MACHINE LEARNING REPORTS



Workshop New Challenges in Neural Computation 2016

Report 04/2016
Submitted: 02.09.2016
Published: 12.09.2016

Barbara Hammer[1], Thomas Martinetz[2], Thomas Villmann[3] (Eds.)
(1) CITEC - Centre of Excellence, University of Bielefeld, Germany
(2) Institute for Neuro- and Bioinformatics, University of Lübeck, Germany
(3) Faculty of Mathematics / Natural and Computer Sciences, University of Applied Sciences Mittweida, Germany

New Challenges in Neural Computation NC^2 – 2016

Barbara Hammer[1], Thomas Martinetz[2], and Thomas Villmann[3]

1 – Cognitive Interaction Technology – Center of Excellence,
Bielefeld University, Germany

2 – Institute for Neuro- and Bioinformatics, University of Lübeck, Germany

3 – Faculty of Mathematics / Natural and Computer Sciences,
University of Applied Sciences Mittweida, Germany

The workshop New Challenges in Neural Computation, NC^2, takes place for the seventh time in a row. As became a custom, it accompanies the prestigious GCPR conference, which takes place in Hanover this year, and it is collocated with two tutorials on embeddings/metric learning and NVIDIA, respectively. Hanover as the thirteenth largest city in Germany is well known for its major trade fairs such as the Hanover fair and CeBIT, providing an inspiring background for the conference.

The workshop itself centres around challenges and novel developments of neural systems and machine learning, covering recent research in theoretical advances as well as practical applications. This year, thirteen contributions from international participants have been accepted as regular contributions, spanning the range from deep learning, robotics, vision and language processing up to advanced learning models, which go beyond standard vector-based data representations, and intriguing applications. In addition, we welcome two renowned researchers as guest speakers, Prof. Dr. Marc Toussaint from University of Stuttgart talks about representation learning, Prof. Dr. Jörg Lücke from University of Oldenburg, presents a new deep learning paradigm based on so-called neural simpletrons. The workshop is supported by the German Neural Network Society (GNNS), and by the CITEC centre of excellence from Bielefeld University, Germany. Within the workshop, a meeting of the GI Fachgruppe on Neural Networks and the GNNS takes place.

We would like to thank our international program committee for their work in reviewing the contributions in a short period of time, the organisers of GCPR for their excellent support, as well as all participants for their stimulating contributions to the workshop.

Contents

Keynote talk: Representation Learning - I've heard that one before

Marc Toussaint, University of Stuttgart, Germany

Abstract:
The revival of NNs surprised some, including me. Back then I considered NNs problematic especially because of their 'representational limitations' in comparison to the explicit structure that can be represented (and learned), e.g., with graphical models, or probabilistic relational models, or representing functions indirectly via optimization or planning problems, as often done in robotics. In fact, the limitation seemed not only w.r.t. representational capacity, but also w.r.t. the computational operations on such representations. It is however interesting to see that 'Representation Learning' became, again, a central research topic in the NN community. I introduce the talk discussing this controversy between the (perhaps feasible?) dream of learning everything in a generic, essentially 'no-prior' substrate ('end-to-end learning') versus the tough science of trying to identify what we believe is essential problem structure and learning relative to such priors. I mention some older work of mine as well as some newer that might seem to move away from the 'representation issue', but never really has.

Keynote talk: Neural Simpletrons - Minimalistic Deep Neural Networks for Probabilistic Learning with Few Labels

Jörg Lücke, University of Oldenburg, Germany

Abstract:
Deep learning is intensively studied using supervised and unsupervised learning, and by applying probabilistic, deterministic, and bio-inspired approaches. Comparisons of different approaches such as generative and discriminative neural networks is made difficult, however, because of differences in the semantics of their graphical descriptions, different learning methods, different benchmarking objectives and different scalability. In this talk I will discuss novel neural networks that are derived from generative modeling approaches but can be formulated as neural networks, i.e., they take a form similar to standard discriminative networks such as perceptrons. These novel networks, which we term Neural Simpletrons, are especially well suited for applications to data with no or few labels because of their roots in generative models. The weakly labelled setting is also well suited for a quantitative comparison with standard and recent state-of-the-art neural networks. Empirical evaluations on common benchmarks show that for weakly labeled data, Neural Simpletrons improve on all standard deep learning approaches and are competitive with their recent variants. As models for neural information processing, our research results suggest neural bottom-up / top-down integration for optimal processing and it assigns important functional roles to synaptic plasticity, synaptic scaling, and intrinsic plasticity.

Classification Margin Dependent Exploration Horizons of Prototypes for Outlier Robust Classification in Learning Vector Quantization

T. Villmann[1], M. Kaden[1], and A. Bohnsack[2]

[1] Computational Intelligence Group, Univ. Applied Sciences Mittweida, DE
[2] Berufliches Schulzentrum Döbeln-Mittweida, DE

Abstract. In this paper we consider an outlier sensitive model for learning vector quantization based on outlier costs compared to misclassification costs. For this purpose, we introduce the exploration domain of an learning vector quantization (LVQ) model obtained by local exploration horizons of the prototypes. These exploration horizons are related to the classification margin for those prototypes localized at the class borders.

1 Introduction

Classification learning by prototype based models gained a large attractiveness during the last years because of its generally good classification performance. Beside the performance power, easy model interpretability and robust adaptation behavior are additional reasons for increasing number of application of those models.

One of the most intuitive classification learning models based on prototypes is learning vector quantization (LVQ,[1]). The model distributes class-dependent prototypes in the data space by a simple attraction and repulsion procedure to recognize the class distributions [2]. This adaption scheme is heuristically motivated but refers to Hebbian learning. A cost based LVQ variant was developed by Sato&Yamada (Generalized LVQ, GLVQ - [3]) approximating the classification error to be optimized by a cost function based on geometric decision model regarding the used data dissimilarity measure, e.g. the Euclidean distance. One of the most interesting advantage of this modification is that GLVQ belongs to the model class of classification margin optimizers [4]. In particular, it maximizes the so-called hypothesis margin.

Yet, class distributions are not always compact. Thus outliers or drift in data may occur, the models has to deal with, e.g. by transformation invariant metrics [5]. Recently, respective reject options were developed for GLVQ to handle those samples during the application phase of the model [6].

In this paper we propose an approach, how to integrate the knowledge about possible outliers in class distributions during learning to obtain an outlier sensitive learning model. Particularly, we relate the acceptance of outliers by the GLVQ classifier depending on outliers costs, which are compared to misclassification cost. For this purpose, the so-called exploration horizon of a prototype is considered, which determines the range of secure classification regarding outliers. In this sense, outlier detection (and acceptance) can be implicitly related to the classification margin.

2 The GLVQ Model for Classification

The GLVQ model assumes data $\mathbf{v} \in \mathbb{R}^N$ with class labels $c(\mathbf{v}) \in \mathscr{C} = \{1, \ldots, C\}$. Further the set $W = \{\mathbf{w}_k\}_{k=1,\ldots,M}$ is considered with class labels $c_k = c(\mathbf{w}_k)$ such that each class is represented by at least one prototype. Classification takes places as a winner-take-all (WTA) rule

$$s(\mathbf{v}) = \text{argmin}_k \left(d(\mathbf{v}, \mathbf{w}_k) \right) \tag{1}$$

where d is a pre-defined dissimilarity measure, e.g. the squared Euclidean distance and $\mathbf{w}_s$ denotes the respective prototype. Thus, the data vector $\mathbf{v}$ is classified as belonging to class c_s.

Let further, $\mathbf{w}^+(\mathbf{v})$ be the best matching prototype for a given data vector $\mathbf{v}$ with respect to the WTA rule (1) which belongs to the same (correct) class as $\mathbf{v}$, i.e. $c(\mathbf{v}) = c(\mathbf{w}^+)$. We define the respective quantity $d^+(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}^+)$. Analogously, $\mathbf{w}^-(\mathbf{v})$ denotes the best matching prototype with $c(\mathbf{v}) \neq c(\mathbf{w}^-)$ (incorrect class) and $d^-(\mathbf{v})$. The cost function of GLVQ optimized by stochastic gradient descent learning (SGDL) with respect to the prototypes is given as

$$E_{GLVQ} = \sum_{\mathbf{v}} C_e \cdot f(\mu(\mathbf{v})) \tag{2}$$

where f is a sigmoid function with $f(x) \in [0, 1]$ and

$$\mu(\mathbf{v}) = \frac{d^+(\mathbf{v}) - d^-(\mathbf{v})}{d^+(\mathbf{v}) + d^-(\mathbf{v})}$$

is the classifier function, which become negative for correct classification, e.g. in case of $d^+(\mathbf{v}) < d^-(\mathbf{v})$. The quantity C_e denotes the cost for a classification error. The *local* hypothesis margin is given as

$$m_h(c_{\tilde{s}}, c_s | \mathbf{v}) = \frac{d(\mathbf{w}_{\tilde{s}}, \mathbf{w}_s)}{2} \tag{3}$$

where $\mathbf{w}_{\tilde{s}}(\mathbf{v})$ is the second best matching prototype with label $c(\mathbf{w}_{\tilde{s}}) \neq c(\mathbf{w}_s)$, i.e

$$\mathbf{w}_{\tilde{s}}(\mathbf{v}) = \begin{cases} \mathbf{w}^+(\mathbf{v}) & , \text{ if } \mathbf{w}_s(\mathbf{v}) = \mathbf{w}^-(\mathbf{v}) \\ \mathbf{w}^-(\mathbf{v}) & , \text{ if } \mathbf{w}_s(\mathbf{v}) = \mathbf{w}^+(\mathbf{v}) \end{cases}, \tag{4}$$

following the definition in [4]. In this way, the local hypothesis margin determines a local range of decision.

Unfortunately, standard GLVQ does not always generates class typical prototypes [7]. To ensure this property, the cost function E_{GLVQ} has to be extended to

$$E_{G-GLVQ} = E_{GLVQ} + \gamma \sum_{\mathbf{v}} d_s(\mathbf{v})$$

with $d_s(\mathbf{v}) = d(\mathbf{v}, \mathbf{w}_s)$ [8], which leads to an additional update term for $\mathbf{w}_s$ in the SGDL scheme [9]. We denote this variant as the generative GLVQ (G-GLVQ).

3 Exploration Horizon for Prototypes

In the following we will consider the outlier problem for G-GLVQ. Particularly, we will introduce so-called exploration horizon for each prototype such that all classification decision for data points inside of this are seen as to be secure with respect to the outlier possibility (O-secure).

For this purpose we assume that are at least several prototypes per class to describe the class distributions. Further, we introduce the exploration horizon

$$H(k) = \frac{d\left(\mathbf{w}_{n(k)}, \mathbf{w}_k\right)}{2} \tag{5}$$

for a prototype $\mathbf{w}_k$, where $\mathbf{w}_{n(k)}$ is the prototype with the smallest dissimilarity value $d\left(\mathbf{w}_{n(k)}, \mathbf{w}_k\right)$. The class label of $\mathbf{w}_{n(k)}$ is denoted by $c_{n(k)} = c\left(\mathbf{w}_{n(k)}\right)$. This situation is visualized in Fig.(1).

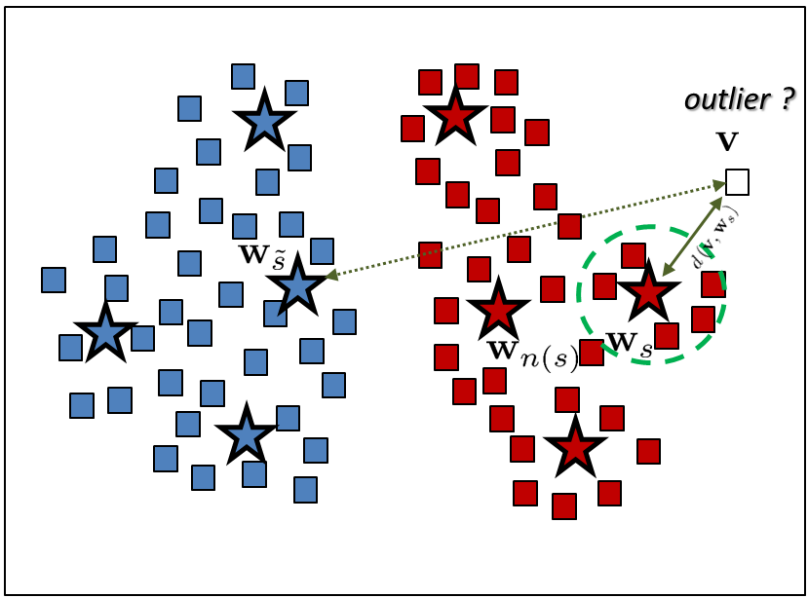


Fig. 1. Visualization of the relations between $\mathbf{w}_s$, $\mathbf{w}_{\bar{s}}$, and $\mathbf{w}_{n(S)}$ to determine the exploration horizon $H(s)$ (green circle) of $\mathbf{w}_s$ by means of the hypothesis margin $m_h(c_{\bar{s}}, c_s)$.

All data points being inside of the exploration horizon of a prototype form the local exploration domain of the prototype. The conjunction of all those domains is denoted as the model exploration domain.

With this notations, a classification according to the WTA-rule (1) is called to be O-secure if

$$\Delta\left(d_s, H\left(s\right)\right) = \frac{d_s - H\left(s\right)}{d_s + H\left(s\right)} > 0 \tag{6}$$

is valid with $d_s\left(\mathbf{v}\right) = d\left(\mathbf{v}, \mathbf{w}_{s(\mathbf{v})}\right)$, i.e. the data vector $\mathbf{v}$ belongs to the local exploration domain of the winning prototype $\mathbf{w}_{s(\mathbf{v})}$. Otherwise, the data sample is considered to be an outlier. We remark that if the nearest neighbor $\mathbf{w}_{n(s)}$ of the overall winning prototype $\mathbf{w}_{s(\mathbf{v})}$ is identical with the second winner $\mathbf{w}_{\tilde{s}}$ from (4), then

$$H\left(s\right) = m_h\left(c_{\tilde{s}}, c_s|\mathbf{v}\right)$$

is valid, i.e. the exploration horizon coincides with the hypothesis margin. Hence, the exploration horizon for prototypes at the class borders is related to the local hypothesis margin, see Fig.(2).

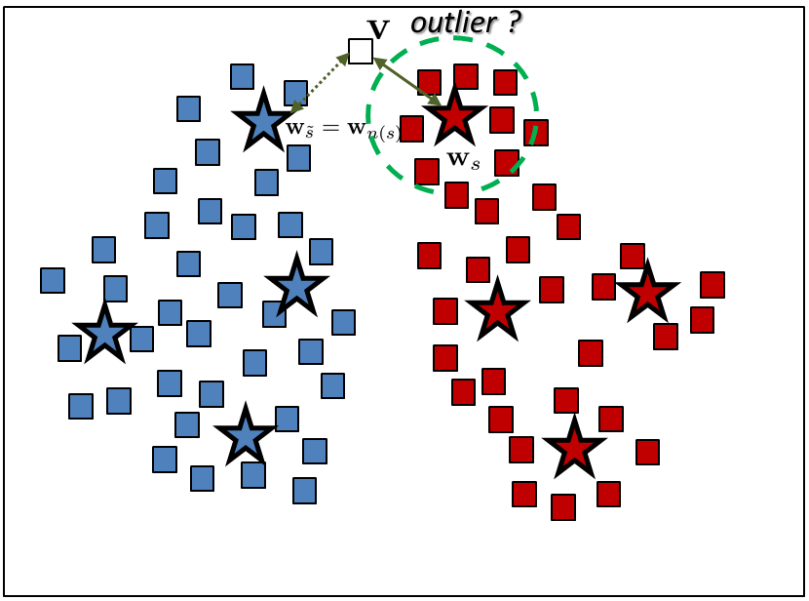


Fig. 2. Visualization of the relations between $\mathbf{w}_s$, $\mathbf{w}_{\tilde{s}}$, and $\mathbf{w}_{n(S)}$ to determine the exploration horizon $H\left(s\right)$ (green circle) of $\mathbf{w}_s$. Here, $\mathbf{w}_{\tilde{s}} = \mathbf{w}_{n(S)}$ is valid such that the local hypothesis margin becomes $m_h\left(c_{\tilde{s}}, c_s|\mathbf{v}\right) = H\left(s\right)$, i.e. the exploration horizon coincides with the margin.

In the following we modify the G-GLVQ such that it is able to adapt regarding outliers. For this purpose and keeping a cost based approach in G-GLVQ as suggested in [10] and [11], we relate outliers to costs C_0 collected in the additional outlier penalty function

$$E_O = C_o \sum_{\mathbf{v} \in \mathrm{V}} f\left(\Delta\left(d_s, H\left(s\right)\right)\right) \tag{7}$$

such that we get

$$E_{GO-GLVQ} = \sum_{\mathbf{v} \in \mathrm{V}} C_e \cdot f\left(\mu\left(\mathbf{v}\right)\right) + C_o \cdot f\left(\Delta\left(d_s, H\left(s\right)\right)\right) + \gamma \cdot d_s\left(\mathbf{v}\right)$$

as the overall cost function for an outlier sensitive G-GLVQ (GO-GLVQ). with the cost function asfor SGDL optimization. The outlier penalty function (7) leads to the additional SGDL updates

$$\begin{aligned}
\frac{\partial E_o}{\partial \mathbf{w}_s} &= C_o f' \cdot \frac{\partial \Delta\left(d_s, H\left(s\right)\right)}{d_s} \cdot \frac{\partial d_s}{\partial \mathbf{w}_s} + f' \cdot \frac{\partial \Delta\left(d_s, H\left(s\right)\right)}{d_{n(s)}} \cdot \frac{\partial H\left(s\right)}{\partial \mathbf{w}_s} \\
&= C_o f' \cdot \frac{d_{n(s)}}{\left(d_s + \frac{1}{2} d_{n(s)}\right)^2} \cdot \frac{\partial d_s}{\partial \mathbf{w}_s} + C_o f' \cdot \frac{-\frac{3}{2} d_s}{\left(d_s + \frac{1}{2} d_{n(s)}\right)^2} \cdot \frac{\partial d_{n(s)}}{\partial \mathbf{w}_s}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial E_o}{\partial \mathbf{w}_{n(s)}} &= C_o\ f' \cdot \frac{\partial \Delta(d_s, H\left(s\right))}{d_{n(s)}} \cdot \frac{\partial d_{n(s)}}{\partial \mathbf{w}_{n(s)}} \\
&= \quad C_o f' \cdot \frac{-\frac{3}{2} d_s}{\left(d_s + \frac{1}{2} d_{n(s)}\right)^2} \cdot \frac{\partial d_{n(s)}}{\partial \mathbf{w}_{n(s)}}
\end{aligned}$$

for $\mathbf{w}_s$ and $\mathbf{w}_{n(s)}$, respectively, in G-GLVQ learning. Here, $d_{n(s)}$ is the abbreviation for $d_{n(s)}\left(\mathbf{v}\right) = d\left(\mathbf{v}, \mathbf{w}_{n(s)}\right)$.

4 Illustrating Example

As an illustrating example we consider the (artificial) data set depicted in Fig.(3), whereby one class shows outlier subsets.

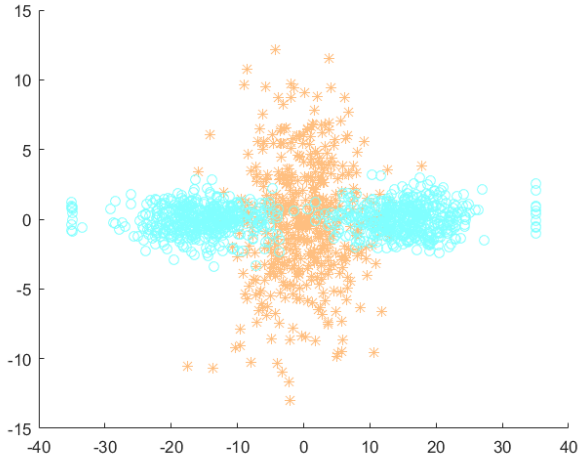


Fig. 3. Visualization of the artificial data set for the illustrating example. The blue class (symmetric horizontal distribution) with 1000 data samples and the red class (vertical) with 500 samples. Remark, the two outlier sets for the blue class.

Applying G-GLVQ with 4 prototypes for the horizontal blue class and two prototypes for the vertical red class we obtain an error rate of 8.0% with 105 samples detected as outliers according to the criterion (6). The distribution of the prototypes in the data space together with the model exploration domain is visualized in Fig.(5). Applying GO-GLVQ with $C_e = 1$ and $C_0 = \frac{1}{25}$ leads to a zero number of outliers but with increased error rate of 9.2%, see Fig.(5).

5 Conclusion

In this paper we discussed an approach of outlier sensitive learning in GLVQ based on the evaluation of the local exploration domains of the prototypes, which can be related to the classification hypothesis margin at the class borders. If outliers should be avoided, a cost based GLVQ approach can be derived balancing misclassification and outlier costs. A first experiment for artificial but illustrating data shows the expected behavior. Yet, real world applications as well as stability analysis of the approach should in the focus of future work.

References

1. Teuvo Kohonen. Learning Vector Quantization. *Neural Networks*, 1(Supplement 1):303, 1988.
2. Teuvo Kohonen. Improved versions of Learning Vector Quantization. In *Proc. IJCNN-90, International Joint Conference on Neural Networks, San Diego*, volume I, pages 545–550, Piscataway, NJ, 1990. IEEE Service Center.

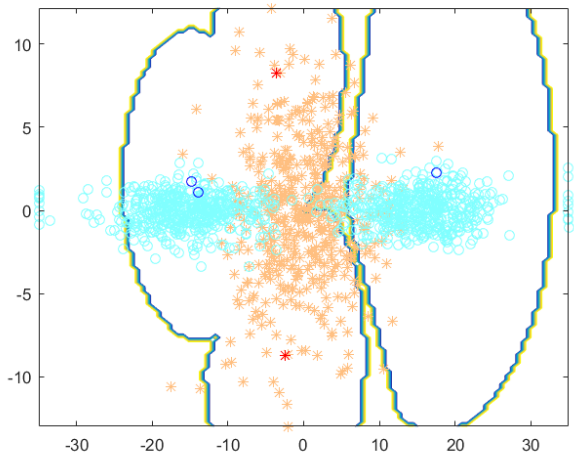


Fig. 4. G-GLVQ training result for the artificial data. The blue circles $'\circ'$ and red stars $'*'$ are the learned prototypes. The lines visualize the model exploration domain. We observe many outliers.

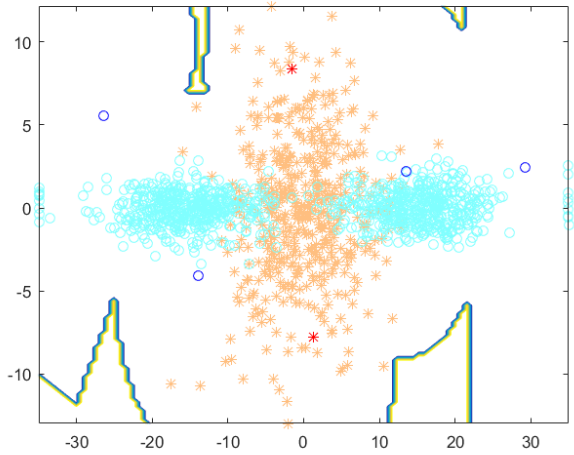


Fig. 5. GO-GLVQ training result for the artificial data. The blue circles $'\circ'$ and red stars $'*'$ are the learned prototypes. The lines visualize the model exploration domain. Comparing with G-GLVQ the prototypes $'\circ'$ are moved to the border regions to capture the outliers.

3. A. Sato and K. Yamada. Generalized learning vector quantization. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9. MIT Press, Cambridge, MA, USA, 1996.
4. K. Crammer, R. Gilad-Bachrach, A. Navot, and A.Tishby. Margin analysis of the LVQ algorithm. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing (Proc. NIPS 2002)*, volume 15, pages 462–469, Cambridge, MA, 2003. MIT Press.
5. S. Saralajew and T. Villmann. Adaptive tangent metrics in generalized learning vector quantization for transformation and distortion invariant classification learning. In *Proceedings of the International Joint Conference on Neural networks (IJCNN) , Vancover*, pages 2672–2679. IEEE Computer Society Press, 2016.
6. L. Fischer, B. Hammer, and H. Wersing. Efficient rejection strategies for prototype-based classification. *Neurocomputing*, 169:334–342, 2015.
7. M. Biehl, B. Hammer, F.-M. Schleif, P. Schneider, and T. Villmann. Stationarity of matrix relevance LVQ. In *Proc. of the International Joint Conference on Neural Networks 2015 (IJCNN)*, pages 1–8, Los Alamitos, 2015. IEEE Computer Society Press.
8. K.L. Oehler and R.M. Gray. Combining image compressing and classification using vector quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):461–473, 1995.
9. B. Hammer, D. Nebel, M. Riedel, and T. Villmann. Generative versus discriminative prototype based classification. In T. Villmann, F.-M. Schleif, M. Kaden, and M. Lange, editors, *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of 10th International Workshop WSOM 2014, Mittweida*, volume 295 of *Advances in Intelligent Systems and Computing*, pages 123–132, Berlin, 2014. Springer.
10. C.K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions in Information Theory*, 16(1):41–46, 1970.
11. T. Villmann, M. Kaden, A. Bohnsack, S. Saralajew, J.-M. Villmann, T. Drogies, and B. Hammer. Self-adjusting reject options in prototype based classification. In E. Merényi, M.J. Mendenhall, and P. O'Driscoll, editors, *Advances in Self-Organizing Maps and Learning Vector Quantization: Proceedings of 11th International Workshop WSOM 2016*, volume 428 of *Advances in Intelligent Systems and Computing*, pages 269–279, Berlin-Heidelberg, 2016. Springer.

Linear Supervised Transfer Learning for Generalized Matrix LVQ

Benjamin Paassen, Alexander Schulz, and Barbara Hammer *

CITEC Center of Excellence, Bielefeld, Germany
bpaassen@techfak.uni-bielefeld.de

Abstract. The utility of machine learning models in everyday applications critically depends on their robustness with respect to systematic changes in the input data. However, many machine learning models trained under lab conditions do break down if they are confronted with such systematic changes. Transfer learning addresses this issue by modelling changes in the input as transfer functions, which can be used to map the data to a space where the learned machine learning model is applicable again.
In this contribution we introduce linear supervised transfer learning as a novel transfer learning scheme and propose a realization based on generalized matrix learning vector quantization. We evaluate our approach in a practical application from the medical domain, namely classifying the intended arm motion from a muscle signal, which can be used by amputees to control a bionic prosthesis and regain hand function after limb loss.

1 Introduction

The robustness of machine learning models under real-world conditions remains a hot topic of machine learning research with significant practical implications. Consider the example of bionic prostheses. For decades, researchers have attempted to develop machine learning models which reliably infer a user's intended motion from muscle signals (Electromyogram, EMG), such that an amputee is able to control her prosthesis just like her former limb [2]. However, current models are still vulnerable to systematic changes in the input data due to electrode shifts, posture changes, sweat, fatigue, etc. [5,13]. In general terms, machine learning models are trained on a certain input data representation. If this repesentation changes, the model is likely to be inaccurate, i.e. models are not *robust* with respect to systematic changes in the data representation [8].

The issue of robustness has been approached from different perspectives in the past. First, it has been suggested to construct features which are *invariant under transformations*, such that certain expected changes to the input data do not influence the input to the machine learning model [6,11].

* Funding by the DFG under grant numbers HA2719/6-2 and HA2719/7-1 and the CITEC center of excellence (EXC 277) is gratefully acknowledged.

Second, in the theory of on-line systems, the notion of *concept drift* has been developed, referring to a change in the conditional distribution of the output given the input [3]. The focus of this approach is not so much on changes in the input data as on changes in the *relation* between input and output, while the input data distribution remains unchanged.

We take a third perspective on the issue of robustness, namely the perspective of *transfer learning*. We assume that the data stems from a stationary, underlying distribution, but is mapped by some function to a different space, in which our machine learning model is not applicable anymore. Our task is to map the data back to a space in which our trained model is valid by means of a so-called *transfer function* [8].

In this contribution, we develop a new transfer learning approach, namely learning a linear transfer function using labelled data to improve the performance of a Generalized Matrix Learning Vector Quantization (GMLVQ) classifier. We evaluate our approach on artificial data as well as myoelectric recordings for prosthesis control.

2 Related Work

Transfer learning is a well-established field concerned with utilizing knowledge from one domain/task in a related domain/task [8]. In this case, we are concerned with systematic changes in the input data representation, while the learning task stays essentially the same. This scenario has has been dubbed *transductive transfer learning* by [8]. [1] further distinguishes the unsupervised case (which they call *transductive*) and the supervised case (which they call *inductive*). We focus here on the supervised case, where some labels for changed input data are available. In contrast to previous approaches in this field, we do *not* adapt the learned model, but rather learn a linear transfer function explicitly, which maps from the new representation to the old representation, such that our original model can be applied again.

3 Supervised Transfer Learning

We phrase a supervised machine learning task as finding a function $f : \mathcal{X} \to \mathcal{Y}$ which maps input data from a space $\mathcal{X}$ to output data from a space $\mathcal{Y}$, such that for some example dataset $Z \subset \mathcal{X} \times \mathcal{Y}$ an error $E(Z, f)$ is minimized. As such, a machine learning problem has the form

$$\min_f E(Z, f) \tag{1}$$

After this problem has been solved, a systematic change in the input data representation occurs via a function $g : \mathcal{X} \to \hat{\mathcal{X}}$, mapping input data from the *source space* $\mathcal{X}$ to a different *target space* $\hat{\mathcal{X}}$. In this space, our learned model f is not necessarily applicable anymore. Note that we assume that data in the target

space are drawn from the *same* underlying distribution as data in the source space, but are transformed via g after generation.

Our aim is to make our model f applicable to the data in the target space. In an unsupervised transfer learning setting, we would attempt to find an approximation of the inverse mapping g^{-1} by means of minimizing the difference between the data distribution in the source space and the distribution of the target-data after mapping to the source space via g^{-1} [1]. However, such an approach has two drawbacks: First, it reproduces features of the source space which are irrelevant to our machine learning task. Second, it does not exploit additional features in the target space which might help to improve the performance in the machine learning task. To address these issues, we propose a *supervised* transfer learning approach, which we characterize as follows: We intend to learn a *transfer function* $h : \hat{\mathcal{X}} \to \mathcal{X}$ which minimizes our error on data from the target space after mapping it back to the source space via h. More precisely, assume that we have access to a small example data set $\hat{Z} \subset \hat{\mathcal{X}} \times \mathcal{Y}$ from the target space. Then we are interested in solving the optimization problem

$$\min_h E(\hat{Z}, f \circ h) \tag{2}$$

where $\circ$ denotes function composition.

4 Linear Supervised Transfer Learning for GMLVQ

In this contribution, we propose a novel realization of supervised transfer learning, namely linear supervised transfer learning for Generalized Matrix Learning Vector Quantization (GMLVQ) [12]. GMLVQ is a prototype-based classification algorithm representing each of the available classes $y \in \{1, \ldots, L\}$ by prototypes $w_{y,1}, \ldots, w_{y,m} \in \mathcal{X}$. Classification is done by assigning the label of the closest prototype:

$$f(x) = \operatorname*{argmin}_y \min_j d(x, w_{y,j}) \tag{3}$$

where the distance d is a general quadratic form:

$$d(x, w) = (x - w)^T \cdot \Omega^T \cdot \Omega \cdot (x - w) \tag{4}$$

The matrix Ω can be viewed as a linear projection of the input data to a space that enhances classification accuracy. A GMLVQ model is learned by adjusting the prototypes as well as the matrix Ω to minimize the cost function

$$E_{\mathrm{GMLVQ}} = \sum_x \Phi\left(\frac{d^+(x) - d^-(x)}{d^+(x) + d^-(x)}\right) \tag{5}$$

where Φ is some nonlinear function (typically sigmoid) and $d^{+/-}(x)$ refers to the distance to the closest prototype with the same/different label as the data point x.

Assume now that a trained GMLVQ model is given and we want to apply it to a setting where the input data is changed by some function g. Let $\mathcal{X} = \mathbb{R}^n$ and $\hat{\mathcal{X}} = \mathbb{R}^{\hat{n}}$. Under the assumption of linearity we can express a transfer function as $h(\hat{x}) = H \cdot \hat{x}$ for some matrix $H \in \mathbb{R}^{n \times \hat{n}}$. Our transfer learning problem in turn is expressed by the minimization problem:

$$\min_{H \in \mathbb{R}^{n \times \hat{n}}} \sum_{\hat{x}} \Phi \left(\frac{d^+(H \cdot \hat{x}) - d^-(H \cdot \hat{x})}{d^+(H \cdot \hat{x}) + d^-(H \cdot \hat{x})} \right) + \lambda \cdot \|H\|_F^2 \tag{6}$$

where $\lambda \cdot \|H\|_F$ is a regularization term with $\lambda \in \mathbb{R}$.

Note that this minimization problem is *not* convex. Still, a local optimum can be found efficiently by initializing H as the identity matrix and adjusting it iteratively by stochastic gradient descent using the gradient

$$\begin{aligned}
&\frac{\partial}{\partial H} \sum_{\hat{x}} \Phi \left(\frac{d^+(H \cdot \hat{x}) - d^-(H \cdot \hat{x})}{d^+(H \cdot \hat{x}) + d^-(H \cdot \hat{x})} \right) + \lambda \cdot \|H\|_F \\
&= \sum_{\hat{x}} \Phi' \cdot \frac{2 \cdot (\frac{\partial}{\partial H} d^+(H \cdot \hat{x})) \cdot d^-(H \cdot \hat{x}) - 2 \cdot (\frac{\partial}{\partial H} d^-(H \cdot \hat{x})) \cdot d^+(H \cdot \hat{x})}{(d^+(H \cdot \hat{x}) + d^-(H \cdot \hat{x}))^2} + 2\lambda \cdot H
\end{aligned}$$

where the gradient of the distance is given as

$$\begin{aligned}
\frac{\partial}{\partial H} d(H \cdot \hat{x}, w) &= \frac{\partial}{\partial H} (H \cdot \hat{x} - w)^T \cdot \Omega^T \cdot \Omega \cdot (H \cdot \hat{x} - w) \\
&= 2 \cdot \Omega^T \cdot \Omega \cdot (H \cdot \hat{x} - w) \cdot \hat{x}^T
\end{aligned}$$

This scheme is by no means specific to GMLVQ. Linear supervised transfer learning can be extended to any machine learning model with a differentiable cost function.

5 Experiments

We evaluate our supervised transfer learning approach on two data sets, one artificial and one consisting of real myoelectric data. We compare our transfer learning algorithm with two baselines: 1) the naive application of the source model to the target space without any adjustment, and 2) a new GMLVQ model trained only on the available training data in the target space (retrain). For training the GMLVQ models we use the GMLVQ implementation provided as part of the CIS SOM Toolbox Version 2.1 (`http://research.ics.aalto.fi/software/somtoolbox/`). Gradient descent for transfer learning is realized using the R-prop algorithm [10]. In both experiments we evaluate the classification error on test data from the target space in a 10-fold crossvalidation. Further, in each fold we vary the number of available training data points in the target space.

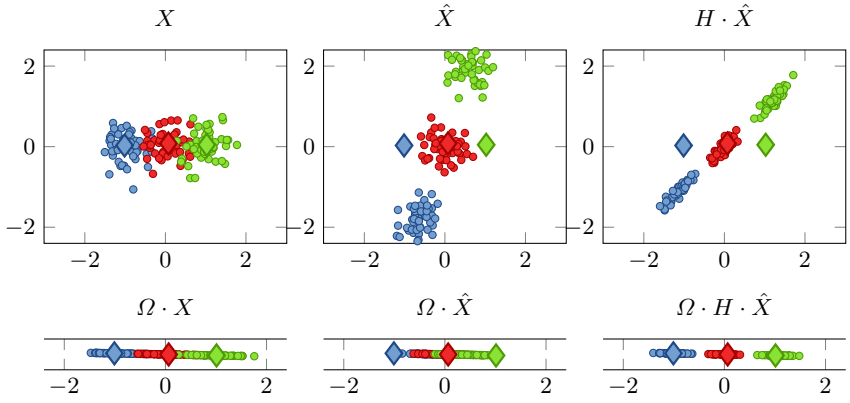


Fig. 1. A visualization of the two-dimensional toy dataset. Data points are displayed as circles, GMLVQ prototypes as diamonds. Colors indicate the class label. The left column shows the dataset in the source space, the middle column in the target space and the right column after transfer mapping via H back to the source space (right). The bottom row displays the dataset after projection via the relevance matrix Ω learned by GMLVQ.

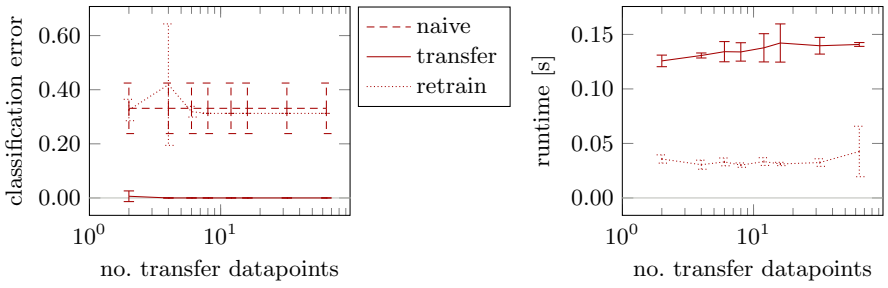


Fig. 2. The experimental results for the toy data set. The x axis shows the number of target space data points used for training in logarithmic scaling. The left plot displays the average classification error on test data from the target space, while the right plot displays the average runtime for training. The standard deviation across cross validation trials is marked by error bars. Different line styles indicate different classification schemes.

5.1 Toy Dataset

Our first dataset consists of 3 classes, each corresponding to a two-dimensional radial Gaussian cluster with 50 data points and standard deviation $\sigma = 0.3$. The means are given as $\boldsymbol{\mu}_1 = (-1, 0)$, $\boldsymbol{\mu}_2 = (0, 0)$ and $\boldsymbol{\mu}_3 = (1, 0)$ respectively (see Figure 1, top left). In this setting, GMLVQ correctly identifies the second dimension as irrelevant and discards it via the projection matrix $\Omega \approx \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ (see Figure 1, bottom left). For the target space, we generated another 50 data points per cluster with the same variance, but moved μ_1 to $R \cdot (-2, 0)$ and μ_3 to $R \cdot (2, 0)$, where R is the rotation matrix for 72° (see Figure 1, top middle). After applying the projection Ω, the data overlaps strongly, rendering classification difficult (see Figure 1, bottom middle).

For learning in the target space we used data from the first two classes only. Yet, even without any information regarding the last class, our proposed transfer learning scheme (with regularization $\lambda = 0.1$) yields a transfer matrix H which sufficiently rotates the data, such that the original GMLVQ model is applicable again (see Figure 1, right). As discussed above, the transfer matrix does *not* map the target space data distribution to the source space data distribution. Instead, it achieves even better class separation than was possible in the source space.

Figure 2 displays the quantitative results. Given 4 our more training data points from the target space, our transfer learning scheme is able to identify a transfer matrix leading to zero classification error in all crossvalidation trials. In comparison, a naive application of the source space model leads to an average classification error of about 33%. This is also the case if we retrain a new GMLVQ model on the available target space data, because all data points from the third class get misclassified.

Regarding runtime, we note that in this simple setting, the GMLVQ training is considerably faster compared to our transfer learning implementation.

5.2 Myoelectric Dataset

Our second data set consists of myoelectric (EMG) data recorded at the Medical University of Vienna [9] [1]. Four healthy subjects were instructed to execute negative and positive activity in three degrees of freedom (wrist rotation, wrist extension, as well as hand open/close) as well as combined movements in two degrees of freedom simultaneously. Subjects executed each movement for five seconds, followed by two seconds of rest. Muscle activity was recorded at 1000Hz sampling rate with an eight channel Ottobock Healthcare electrode array (13E200) attached around the forearm. We preprocessed the raw data by accumulating time windows of 100ms with 50ms overlap. As features, we used the 17 standard features offered by *BioPatRec* [7], in addition to the log-variance as suggested by [4]. We modelled the movement classification via three different GMLVQ classifiers, one for each degree of freedom with three classes each

[1] Special thanks go to Cosima Prahm for permission to use the data set.

(movement in negative direction, no movement, movement in positive direction), such that combined movements in multiple degrees of freedom could be classified as well. Disturbance was applied by shifting the electrode array by 8mm transversally and recording all movements one more time.

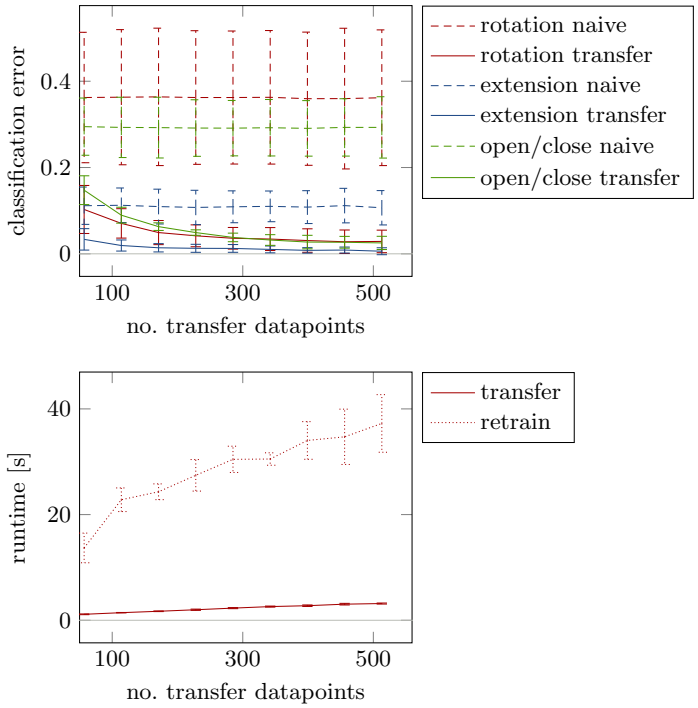


Fig. 3. The experimental results for the toy data set. The x axis shows the number of target space data points used for training in logarithmic scaling. The left plot displays the average classification error on test data from the target space, while the right plot displays the average runtime for training. In the left plot, error bars mark the standard deviation across subjects and colors indicate the degree of freedom. In the right plot, error bars indicate the standard deviation across crossvalidation trials.

The experimental results are shown in Figure 3. In the source space, GMLVQ achieves classification error below 1% for all degrees of freedom. A naive application to target space data, however, yields errors of about $\approx 36\%$ for wrist rotation. Our proposed transfer learning scheme (without regularization) reduces the error to below 4% for all degrees of freedom. Compared to a retraining of GMLVQ in the target space, transfer learning is considerably faster, with constant factors of $10 - 15$. Due to the high runtime required for retraining the GMLVQ model, we did not repeat the full experiment for retraining. However,

results for a single subject in crossvalidation strongly indicated that GMLVQ retraining achieves similar or even better classification accuracy compared to transfer learning, if data from all classes is available.

6 Conclusion

In this contribution we extended transfer learning by proposing a realization via a linear transfer function on generalized matrix learning vector quantization (GMLVQ) classifiers. We demonstrated that using labels in the target space has benefits beyond unsupervised transfer learning approaches, namely ignoring irrelevant features of the source space and exploiting relevant features of the target space. Further, linear supervised transfer learning can outperform a simple retraining of the classification model, if either the model is too complex, leading to prohibitive runtime, or if labelled data is not available for all classes. The theoretical foundations for linear supervised transfer learning provide opportunity for further research. In particular, it would be beneficial to identify conditions under which data from few classes in the target space only is sufficient for successful transfer learning.

References

1. Arnold, A., Nallapati, R., Cohen, W.W.: A comparative study of methods for transductive transfer learning. In: Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007). pp. 77–82 (2007)
2. Farina, D., Jiang, N., Rehbaum, H., Holobar, A., Graimann, B., Dietl, H., Aszmann, O.C.: The extraction of neural information from the surface emg for the control of upper-limb prostheses: Emerging avenues and challenges. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22(4), 797–809 (July 2014)
3. Gama, J.a., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. 46(4), 44:1–44:37 (Mar 2014)
4. Hahne, J.M., Biebmann, F., Jiang, N., Rehbaum, H., Farina, D., Meinecke, F.C., Mller, K.R., Parra, L.C.: Linear and nonlinear regression techniques for simultaneous and proportional myoelectric control. IEEE Transactions on Neural Systems and Rehabilitation Engineering 22(2), 269–279 (2014)
5. Khushaba, R.N., Takruri, M., Miro, J.V., Kodagoda, S.: Towards limb position invariant myoelectric pattern recognition using time-dependent spectral features. Neural Networks 55, 42–58 (2014)
6. LeCun, Y.: Learning Invariant Feature Hierarchies, pp. 496–505. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
7. Ortiz-Catalan, M., Brånemark, R., Håkansson, B.: Biopatrec: A modular research platform for the control of artificial limbs based on pattern recognition algorithms. Source Code for Biology and Medicine 8(1), 1–18 (2013)
8. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering 22(10), 1345–1359 (Oct 2010)
9. Prahm, C., Paaßen, B., Schulz, A., Hammer, B., Aszmann, O.: Transfer learning for rapid re-calibration of a myoelectric prosthesis after electrode shift. In: Proceedings of the 3rd International Conference on Neural Rehabilitation (2016), accepted

10. Riedmiller, M., Braun, H.: A direct adaptive method for faster backpropagation learning: The rprop algorithm. In: IEEE International Conference on Neural Networks. pp. 586–591 (1993)
11. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contracting auto-encoders: Explicit invariance during feature extraction. In: In Proceedings of the Twenty-eight International Conference on Machine Learning (ICML11) (2011)
12. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization. Neural Computation 21(12), 3532–3561 (2009)
13. Vidovic, M., Hwang, H.J., Amsuss, S., Hahne, J., Farina, D., Mller, K.R.: Improving the robustness of myoelectric pattern recognition for upper limb prostheses by covariate shift adaptation. IEEE Transactions on Neural Systems and Rehabilitation Engineering (99) (2015)

Relevance Learning Vector Quantization in Variable Dimensional Spaces

Kerstin Bunte[1,3], Elizabeth S. Baranowski[2], Wiebke Arlt[2], and Peter Tino[1]

[1] School of Computer Science, The University of Birmingham, Birmingham, UK
[2] Institute of Metabolism and Systems Research, University of Birmingham, UK
[3] Now at: Faculty of Mathematics and Natural Sciences, University of Groningen, Groningen, The Netherlands

1 Introduction

Due to improved biochemical sensor technology and biobanking efforts in North America and Europe the amounts of complex biomedical data is growing constantly. With the data also the demand for interpretable interdisciplinary analysis techniques increases. Further difficulties arise since biomedical data is often very heterogeneous, either due to the availability of measurements or individual differences in the biological processes. Urine steroid metabolomics is a novel biomarker tool for adrenal cortex function [1] measured by gas chromatography-mass spectrometry (GC-MS), which is considered the reference standard for the biochemical diagnosis of inborn steroidogenic disorders. Steroidogenesis encompasses the complex process by which cholesterol is converted to biologically active steroid hormones. Inherited or inborn disorders of steroidogenesis result from genetic mutations which lead to defective production of any of the enzymes or a cofactor responsible for catalysing salt and glucose homeostasis, sex differentiation and sex specific development. Treatment involves replacing the deficient hormones which, if replaced adequately, will in turn suppress any compensatory up-regulation. Currently, up to 34 distinct steroid metabolite concentrations are extracted from a single GC-MS profile by automatic quantitation following selected-ion-monitoring (SIM) analysis, resulting in a 34 dimensional fingerprint vector. However, the interpretation of this fingerprint is difficult and requires enormous experience and expertise, which makes it a relatively inaccessible tool for most clinical endocrinologists.

In this paper we present a novel interpretable machine learning method for the computer-aided diagnosis of three conditions including the most prevalent, 21-hydroxylase deficiency (CYP21A2), and two other representative, but rare conditions, 5α-reductase type 2 deficiency (SRD5A2) and P450 oxidorectase deficiency (PORD). Our data set contains a large collection of steroid metabolomes from over 800 healthy controls of varying age (including neonates, infants, children, adolescents and adults) and over 100 patients with newly diagnosed, genetically confirmed inborn steroidogenic disorders. The clinical data will be presented at the Society for Endocrinology BES Conference [2].

The data set and problem formulation comprises several computational difficulties. On average 8% to 13% of measurements from healthy controls and patients respectively are missing or not detectable (indicated by 0). The problem

now arises because those measurements are not missing at random but systematically, since the data collection combines different studies and quantitation philosophy has changed over the years. Furthermore, the measurements are very heterogeneous. Neonates and infants naturally deliver less urine and only from Spot and Nappy instead of volume. Moreover the individual excretion amounts vary a lot due to natural adrenal development and peripheral factors even in healthy controls and of course severity of enzymatic deficiency in patients. To account for these difficulties we propose an interpretable prototype based machine learning method using a dissimilarity between two metabolomic profiles based on the angle Θ between them calculated on the observed dimensions. Using the angles instead of distances has two principal advantages: (1) distances calculated in spaces of varying dimensionality (depending on the number of shared observed dimensions in two metabolomic fingerprints) do not share the same scale and (2) the angles naturally express the idea that only the *proportional* characteristics of the individual profiles matter.

2 Method

We propose Angle Learning Vector Quantization (angle LVQ) as an extension to Generalized Relevance LVQ (GRLVQ) [5, 4]. As in the original formulation we assume training data given as z-transformed vectorial measurements (zero mean, unit standard deviation) accompanied by labels $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ and a user determined number of labelled prototypes $\{(\boldsymbol{w}_m, c(\boldsymbol{w}_m))\}_{m=1}^{M}$ representing the classes. Classification is performed following a Nearest Prototype Classification (NPC) scheme, where a new vector is assigned the class label of its closest prototype. Our approach differs from GRLVQ by using an angle based similarity instead of the Euclidean distance. The vector of (adaptive) relevances (one for each dimension), $\boldsymbol{r}$, weights now the influence of individual dimensions when calculating the angles, such that minimal within-class variation (fingerprints of the same conditions point in similar directions) and maximum inter-class variation (different conditions are well-separated in the angle space) is achieved.

Both prototypes and relevances $R = \text{diag}(\boldsymbol{r})$ are determined by a supervised training procedure minimizing the following cost function [5] calculated on the observed dimensions:

$$E = \sum_{i=1}^{N} \frac{d_i^J - d_i^K}{d_i^J + d_i^K} \ .$$

Here the dissimilarity of each data sample $\boldsymbol{x}_i$ with its nearest correct prototype with $y_i = c(\boldsymbol{w}_J)$ is defined by d_i^J and by d_i^K for the closest wrong prototype ($y_i \neq c(\boldsymbol{w}_K)$). Now the distances $d_i^{\{J,K\}}$ are replaced by angle-based dissimilarities:

$$d_i^L = g_\beta \left(\frac{\boldsymbol{x}_i R \boldsymbol{w}_L^\top}{\sqrt{\boldsymbol{x}_i R \boldsymbol{x}_i^\top} \sqrt{\boldsymbol{w}_L R \boldsymbol{w}_L^\top}} \right) \tag{1}$$

$$\text{with} \quad g_\beta(b) = \frac{\exp\{-\beta(b+1)\} - 1}{\exp(2\beta) - 1} \text{ and } L \in \{J, K\} \ . \tag{2}$$

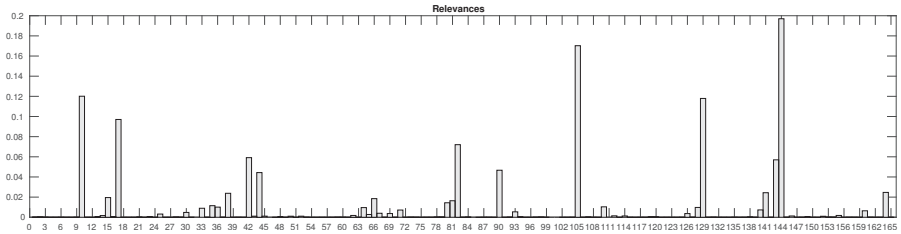


Fig. 1. Relevance vector of the best angle LVQ model found by cross validation.

Here, the exponential function g_β with slope β transforms the weighted dot product $b = \cos\Theta_R \in [-1, 1]$ to a dissimilarity $\in [0, 1]$. Finally, training is performed by minimizing the cost function E, which exhibits a large margin principle [3]. To ensure positivity of the relevances we set $r_j = a_j^2$ and we optimize a_j's collected in a vector $\boldsymbol{a}$. We furthermore restrict $\boldsymbol{r}$ by a penalty term $(1 - \sum_j r_i)^2$ added to E. Lastly, we added a regularization term $-\gamma \sum_j \log r_j$ to E to prevent oversimplification effects. Optimization can be performed for example by steepest gradient descent. The derivatives can be found in the appendix section 5.

3 Experiments

We test the proposed technique on the metabolomic data described above and classify the 3 conditions CYP21A2, PORD and SRD5A2 from healthy controls. Since the conditions affect enzyme activity we represent the metabolomic profiles by vectors of pair-wise steroid ratios. From the 34^2 possible ratios we select 165 by analysis of variance (ANOVA) of the conditions versus healthy. Furthermore, we randomly set aside over 700 healthy samples and ca. 4 samples of each condition as test set, so the majority class is down sampled. Now we train our angle LVQ method using 5 fold cross-validation on the remaining data using one prototype per class and regularization with $\gamma = 0.001$. We achieve a very good mean (std) sensitivity of 0.81 (0.049) for detecting patients with one of the three conditions trained, 0.73 (0.069) precision and an excellent specificity of 0.97 (0.008) for healthy controls. The resulting relevance vector of the best model is shown in figure 1, where distinct steroid ratios were identified as most important for classification. Note, that even samples with 30 to 79% of its ratios missing were on average 98.7% classified correctly with this model. In direct comparison GRLVQ with mean imputation for the missing values trained on the same data splits achieves in average 0.98 (0.018) specificity and 0.81 (0.2) precision for normal profiles, but only a sensitivity of 0.42 (0.106) for patients.

4 Conclusion and Future Work

We propose an angle and prototype based relevance learning technique called angle LVQ to learn data of variable dimensions. First results show very good sensitivity for the prediction of pathological fingerprints from urine metabolomic

profiles as well as excellent specificity to distinguish patients from healthy controls. Future work will include an in-depth analysis of the bio-medical impact of this findings. Furthermore, we plan to extend this approach for angle based transformation, compare its performance to alternative techniques for data with missing values and derive further theory for learning.

5 Appendix

The derivatives of E (Eq. 1) with $R_{jj} = a_j^2$ and $\|\boldsymbol{v}\|_A = \sqrt{\sum_{m=1}^{M} v_m^2 a_m^2}$ are:

$$\frac{\partial E}{\partial \boldsymbol{w}_J} = \sum_{i=1}^{N} \frac{2d_i^K}{(d_i^J + d_i^K)^2} \frac{\partial d_i^J}{\partial \boldsymbol{w}^J} \quad \text{and} \quad \frac{\partial E}{\partial \boldsymbol{w}_K} = \sum_{i=1}^{N} \frac{-2d_i^J}{(d_i^J + d_i^K)^2} \frac{\partial d_i^K}{\partial \boldsymbol{w}^K} \tag{3}$$

$$\frac{\partial g_\beta(b)}{\partial b} = \frac{-\beta \exp\{-\beta b + \beta\}}{\exp\{2\beta\} - 1} \tag{4}$$

$$\frac{\partial d^L}{\partial \boldsymbol{w}_{\{L,j\}}} = \frac{\partial g_\beta}{\partial \boldsymbol{w}_L} \frac{a_j^2 (x_j \sum_m w_{\{L,m\}}^2 a_m^2 - \sum_m x_m w_{\{L,m\}} a_m^2)}{\|\boldsymbol{x}\|_A \|\boldsymbol{w}_L\|_A^3} \tag{5}$$

$$\frac{\partial E}{\partial a_j} = \sum_{i=1}^{N} \frac{2d_i^K \frac{\partial d_i^J}{\partial a_j} - 2d_i^J \frac{\partial d_i^K}{\partial a_j}}{(d_i^J + d_i^K)^2} \tag{6}$$

$$\frac{\partial d^L}{\partial a_j} = \frac{a_j 2 x_j w_{\{L,j\}}}{\|\boldsymbol{x}\|_A \|\boldsymbol{w}_L\|_A} - \frac{x_j^2 \sum_m x_m w_{\{L,m\}} a_j^2}{\|\boldsymbol{x}\|_A^3 \|\boldsymbol{w}_L\|_A} - \frac{w_j^2 \sum_m x_m w_{\{L,m\}} a_m^2}{\|\boldsymbol{x}\|_A \|\boldsymbol{w}_L\|_A^3} \tag{7}$$

where $\boldsymbol{v}_{\{\cdot,j\}}$ denotes dimension j of vector $\boldsymbol{v}$.

References

1. Arlt, W., Biehl, M., Taylor, A.E., Hahner, S., Hughes, R.L.B.A., Schneider, P., Smith, D.J., Stiekema, H., Nils Krone, E.P., Opocher, G., Bertherat, J., Franco Mantero, B.A., Terzolo, M., Nightingale, P., Cedric H. L. Shackleton, X.B., Fassnacht, M., Stewart, P.M.: Urine steroid metabolomics as a biomarker tool for detecting malignancy in adrenal tumors. The Journal of Clinical Endocrinology and Metabolism 12(96), 3775–3784 (2011)
2. Baranowski, E.S., Bunte, K., Shackleton, C.H., Taylor, A.E., Hughes, B.A., Biehl, M., Tino, P., Guran, T., Arlt, W.: Steroid metabolomics for diagnosis of inborn steroidogenic disorders - bridging the gap between biochemist and clinician through computational approaches. Paper abstract for Society for Endocrinology BES (2016)
3. Hammer, B., Strickert, M., Villmann, T.: On the generalization ability of grlvq networks. Neural Processing Letters 21(2), 109–120 (2005)
4. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks 15(8–9), 1059 – 1068 (2002)
5. Sato, A.S., Yamada, K.: Generalized learning vector quantization. In: Advances in Neural Information Processing Systems. vol. 8, pp. 423–429 (1996)

Functional approximation for the classification of smooth time series

Friedrich Melchert[1,2], Udo Seiffert[2], and Michael Biehl[1]

[1] University of Groningen, Johann Bernoulli Institute for Mathematics and Computer Science, P.O. Box 407, 9700 AK Groningen, The Netherlands
[2] Fraunhofer Institute for Factory Operation and Automation IFF, Sandtorstrasse 22, 39106 Magdeburg, Germany

Abstract. Time series data are frequently analysed or classified by considering sequences of observations directly as high-dimensional feature vectors. The presence of several hundreds or thousands of input dimensions can lead to practical problems. Moreover, standard algorithms are not readily applicable when the time series data is non-equidistant or the sampling rate is non-uniform. We present an approach that allows for a massive reduction of input dimensions and explicitly takes advantage of the functional nature of the data. Furthermore, the application of standard classification algorithms becomes possible for inhomogeneously sampled time series. The presented approach is evaluated by applying it to four publicly available time series datasets.

Keywords: Classification; supervised learning; functional data; time series; Learning Vector Quantization; relevance learning; dimensionality reduction; missing values

1 Introduction

The classification of time series data is of interest in various domains including medicine, finance, entertainment and industry [19]. In many applications the time series data is sampled with high temporal resolution, resulting in high-dimensional feature vectors. Traditional classification schemes often display inferior performance when applied to nominally very high-dimensional data. However, due to temporal correlations, the large number of features does not necessarily correspond to high intrinsic dimension in time series data [18]. Although a variety of machine learning techniques are able to handle high-dimensional datasets, most of them were not designed to take advantage of the functional nature and temporal ordering of the features [8].

Here, we consider an explicit functional representation of time series data which exploits the correlation of subsequent measurements and reduces the number of input dimensions drastically. To implement the actual classification task, different machine learning algorithms can be applied, each having characteristic advantages and disadvantages. Here, we resort to prototype and distance based classifiers, such as *Learning Vector Quantization* (LVQ) [10], which are

straightforward to implement and allow for intuitive interpretation [1,3,4]. The prototypes in LVQ represent typical exemplars of their corresponding classes. Together with a suitable distance mesaure, they constitute an efficient classification system [3,4].

The choice of an appropriate distance is a key step in the design of any prototype based classification system. Although it is computational costly, *Dynamic Time Warping* (DTW) [14] is considered a standard choice for comparing time series [13]. Here, we employ a fast and adaptive quadratic distance measure in the framework of *Generalized Matrix Relevance LVQ* (GMLVQ), which is optimized in the training process [15,3]. This is not only more flexible than the use of fixed, predefined measures, it also facilitates the interpretation of the emerging distance measure which provides important insights into the structure of the input data with respect to the classification task [15,16].

Previously, similar variants of relevance LVQ were considered in the context of short term and long term predictions of time series in [17]. The use of a functional representation together with GMLVQ in coefficient space was discussed in [11] for spectral and other functional data. Here, we will transfer and extend this approach to smooth time series and their specific properties. In particular, we will show how the functional nature of the data can be exploited to cope with missing and non-equidistant sampled data.

In the next section we will outline the general framework of time series classification by combining GMLVQ with functional representations. In section 3 the performed experiments are described and their results are shown. We conclude with a discussion of the results and a brief outlook on open research questions.

2 Polynomial approximation of time series

We consider the general classification setup, where a training set of N labeled feature vectors $(\boldsymbol{x}_i, y_i) \in \mathbb{R}^d \times \{1 \dots A\}, i = 1 \dots N$ is used to train a classifier. Here d denotes the dimension of the data and A the number of different classes in the dataset. The trained classifier assigns a class label $y(\boldsymbol{x}) = 1 \dots A$ to any feature vector $\boldsymbol{x}$.

Furthermore, we assume that the feature vectors $\boldsymbol{x}_i$ represent discrete time series data, which result from sampling an unknown function $f_i(t)$ at some known time points t_j. In the following we will assume the time scale to be the interval $t \in [-1 \dots 1]$ and denote the discretized observations as

$$x_{i,j} = f_i\left(t_j\right). \tag{1}$$

Given a suitable set of basis function $g_k(t)$ it is possible to represent $f_i(t)$ as a weighted sum of the basis functions:

$$f_i\left(t\right) = \sum_{k=0}^{\infty} c_{i,k} g_k(t). \tag{2}$$

Restricting the number of coefficients to a finite number n, Eq. (2) becomes, in general, an approximation $\hat{f}_i(t)$ of the original function $f_i(t)$.

Although using a Fourier basis is first choice in many signal processing applications it is most suitable for periodic functions. Here we use Chebyshev polynomials of the first kind as basis functions. They provide an efficient way to represent non-periodic smooth functions and have favourable properties with respect to numerics [6]. The recursive definition reads

$$T_0(x) = 1; \qquad T_1(x) = x; \qquad T_n(x) = 2xT_{n-1} - T_{n-2}(x). \tag{3}$$

The approximation coefficients $c_{i,k}$ can be determined by minimizing a suitable optimization criterion, e.g. the quadratic error $e = \sum_{j=1}^{d}(f_i(t_j) - \hat{f}_i(t_j))^2$ or the maximum deviation $e = \max_{j=1\ldots d}(f_i(t_j) - \hat{f}_i(t_j))$. Here, we exploit the properties of truncated Chebyshev series to compute the coefficient values in an efficient way [9]:

$$c_{i,k} = \frac{2}{n+1}\sum_{l=0}^{n} f_i(t_l)T_k(t_l) \text{ , with } t_l = \cos\left(\left(l+\frac{1}{2}\right)\frac{\pi}{n+1}\right). \tag{4}$$

Given the maximum degree n, the sampling points t_l represent the roots of the Chebyshev polynomial of degree $(n+1)$. Since, in general, the original sampling points will not match these roots, we perform a simple, linear interpolation of the original data in order to obtain the values of $f_i(t_l)$. The linear interpolation is justified under the assumption that the distance of the t_l from the known sampling points is small compared to the overall length of the time series. It is, of course, possible to use more powerful interpolation schemes, e.g. Floater Hormann interpolants [7]. However, using a linear scheme has advantages in terms of computational effort and, moreover, its invertibility facilitates a suitable interpretation of the results as demonstrated and discussed below. Note that approximation quality is not the main goal in the following. The polynomial representation serves as a method fpr feature extraction in terms of the resulting coefficients.

We can summarize the transformation from the original data to the space of approximation coefficients by the equation

$$\boldsymbol{c}_i = \boldsymbol{S}\boldsymbol{P}\boldsymbol{x}_i = \boldsymbol{\Psi}\boldsymbol{x}_i, \tag{5}$$

where the matrix $\boldsymbol{S} \in \mathbb{R}^{n\times d}$ represents the linear interpolation of the original data at the sampling points t_l and the matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ represents the first n Chebyshev polynomials evaluated at the sampling points t_l.

The setup can be easily extended to non-equidistant and non-uniform sampled time series, since no assumption on the number and distribution of the original sampling points t_j is made. An extension to a particular sampling $t_{j,i}$, which could be even data point specific, is straightforward according to Eqs. (1-5) and only affects the interpolation matrix, introducing individual $\boldsymbol{S}_i$.

Under the assumption that the available data results from sampling a smooth time-dependent function, the presented approach allows for a transformation to the more abstract space of coefficients. This transformation is also feasible if the input data is not equidistant (different time intervals between sampled points) or not uniform (different number of time-points sampled).

Table 1: Selected datasets from the UCR Time Series Repository [5], together with the number of samples, sampling points and classes.

Dataset name	classes	sampling points	samples (training)	samples (validation)
ItalyPowerDemand	2	24	67	1029
Plane	7	144	105	105
StarLightCurves	3	1024	1000	8236
Strawberry	2	256	370	613

3 Application to example datasets

In order to evaluate the suggested approach, it is applied to four publicly available, relatively smooth time series datasets taken from the UCR repository [5]. The selected datasets and their key properties are listed in Table 1. Note that the repository does not provide detailed information with respect to, e.g., the interpretation of the values, the meaning of classes or the real world time scales.

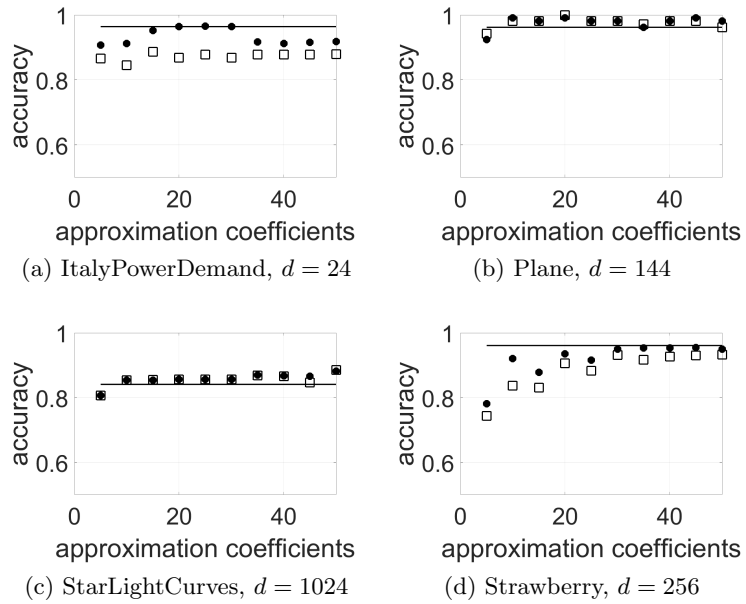
For each of the datasets three setups were considered for computer experiments. To obtain a natural baseline for the achievable classification performance in a first setup (A) the classifiers were trained from the original time series data.

For a second set of experiments (B) the data were transformed to vectors of approximation coefficients and GMLVQ training was performed in this space. The experiments were repeated for different numbers of coefficients: $n = 5, 10, \ldots 50$.

In the third expermimental setup (C) the original data was manipulated in order to simulate non-equidistant, non-uniform sampled data. To this end, a random number (between 20% and 60%) of values was discarded from each available feature vector. Which values were actually deleted was also chosen randomly and independently for each data point. This resulted in modified feature vectors with varying number of sampling points and randomized positions of the available points. The modified dataset $\{\tilde{\boldsymbol{x}}_i, \boldsymbol{t}_i\}$ was then used to transform the data to the space of approximation coefficients according to Eqs.(4,5). As in setup (B), the number of coefficients was varied as $n = 5, 10, \ldots 50$.

In all experiments a corresponding GMLVQ system was trained from the respective set of labeled feature vectors using the same set of parameters. All systems comprised one prototype per class. Before each training process the data was preprocessed in terms of a z-score transformation, yielding zero mean and unit variance in all dimensions, and therefore equalizing the magnitudes of the different features. The z-score transformation facilitates the intuitive interpretation of the emerging relevance matrices [15]. The relevance matrix was initialized as proportional to the identity, while the prototypes were initialized in the corresponding class-conditional means. As optimization scheme a batch gradient descent with adaptive step sizes along the lines of [12] was performed with default parameters as suggested in [2].

The performance of the emerging GMLVQ systems was evaluated as the overall classification accuracy with respect to the corresponding validation dataset



(a) ItalyPowerDemand, $d = 24$ (b) Plane, $d = 144$

(c) StarLightCurves, $d = 1024$ (d) Strawberry, $d = 256$

Fig. 1: Classification accuracies achieved in the respective validation sets as a function of the number of approximation coefficients. The solid lines represent the accuracy achieved in the full set of all available input features (experimental setup A). Filled circles correspond to accuracies resulting from the classification in the space of approximation coefficients (B). Empty squares mark the results achieved after the randomized deletion of time-points in setting (C). For comparison the original number of sampling points for each dataset is denoted.

in the UCR archive [5] (cf. Table 1). Validation data underwent the same preprocessing as the training set in each individual experiment. This includes the transformation to the space of approximation coefficients and the randomized deletion of time-points in setting (C). The z-score transformation of the data was performed with respect to the mean and variance determined from the training dataset. The results of the experiments are depicted in Figure 1.

4 Results and Discussion

In the example datasets considered here, we observe only insignificant or no increase of the classification accuracy. However, the transformation of the data to the space of approximation coefficients yields a massive reduction of input dimension. The largest reduction (99%) was achieved in the *StarLightCurves* dataset when using $n = 10$ coefficients.

The evaluation of results from setup (C), where up to 60% of the data points were disregarded, shows that the approach can compensate for missing data and irregular sampling to a very large extent. In fact, the results show that

the random removal of time-points had no impact on the overall classification performance achieved in the considered example problems.

One of the main advantages of prototype based classification is that the prototypes are determined in the domain of the original data. A GMLVQ system directly trained from time series data, yields interpretable prototypes and relevances with respect to the sampling points of the time series. In the setups (B) and (C), however, the GMLVQ system is adapted in the more abstract space of approximation coefficients. Hence, it is not obvious how to interpret prototypes and relevance matrices adequately. In previous work [11], the interpretation of prototypes and relevances in the space of coefficients was provided with respect to the characteristics of the basis functions. Since this is less intuitive than an interpretation in the original feature space, it is desirable to back-transform prototypes as well as relevance matrices to the original time series representation. In order to obtain such a transformation we can use the matrix $\boldsymbol{\Psi}$ introduced in Eq. (5): Including the transformation into the distance measure applied in GMLVQ [15] we obtain

$$d(\boldsymbol{x}, \boldsymbol{z}) = (\boldsymbol{x} - \boldsymbol{z})^\top \boldsymbol{\Psi}^\top \boldsymbol{\Lambda}_c \boldsymbol{\Psi} (\boldsymbol{x} - \boldsymbol{z}) \tag{6}$$

where $\boldsymbol{\Lambda}_c$ denotes the relevance matrix obtained in coefficient space. This yields the relation

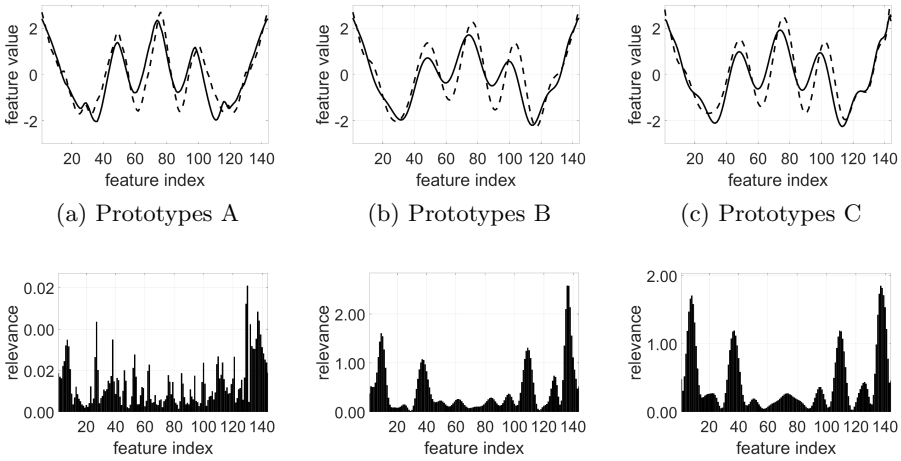
$$\boldsymbol{\Lambda} = \boldsymbol{\Psi}^\top \boldsymbol{\Lambda}_c \boldsymbol{\Psi} \tag{7}$$

which translates the obtained relevance matrix back to original feature space.

An illustrative example for the prototypes and relevance matrices obtained in settings (A), (B) and (C) for the *Plane* dataset is depicted in Fig. 2. Apart from the implicit smoothening it is evident that, both, prototypes and relevance profiles are very similar to those obtained in the original feature space. As a result of the applied normalization steps, the absolute values can be different, but the general shapes of the relevance profiles are essentially identical. The comparison of Figs. 2d, 2e, and 2f, does not reveal major differences. Note, in particular, that although there is a loss of information in experiments (C) due to the random dilution of time-points, prototypes as well as relevances can be transformed to a uniformly sampled input space. Therefore we maintain their interpretability over the complete input space.

5 Summary and Outlook

We have presented an approach for time series classification using a representation that takes the functional nature of smooth time series into account. Our computer experiments show that the approximation of the time series with a suitable set of basis functions yields a massive reduction of input dimensionality without significant loss of classification accuracy. Furthermore we studied the influence of irregular, missing data by randomly deleting up to 60% of the values in each sample. The achieved results show that the functional approximation of the data can compensate for the missing information to a very large extent. No significant decrease in classification accuracy was observed.



(a) Prototypes A (b) Prototypes B (c) Prototypes C

(d) Relevances A (e) Relevances B (f) Relevances C

Fig. 2: Prototypes and relevance profiles emerging from the different setups (A, B, C). For setting (A) prototypes and relevance profiles directly emerge from the model, in (B) and (C) they are shown after back-transformation to the original feature space. The shown results were achieved using $n = 20$ approximation coefficients. For the sake of clarity, only the prototypes for the first (solid line) and second (dashed line) class are shown.

The use of Chebyshev polynomials as basis functions in combination with a linear resampling of the data constitutes a suitable representation of time series. Furthermore the transformation of the data can be done in a single matrix multiplication and therefore has clear advantages over DTW in terms of computational effort. Finally, the linearity and invertibility of the transformation makes it possible to interpret the GMLVQ system also in the original input space. The interpretation of prototypes and relevances is maintained over the full time domain, even for time series with non-equidistant and non-uniform sampling.

Future work will concern the selection of alternative basis functions for the analysis of time series and other functional data. An interesting question concerns the choice of an optimal number of approximation coefficients corresponding to a minimum number of adaptive parameters while maintaining close to optimal accuracy. The presented approach allows for a compact representation of smooth time series, which should be very useful for the analysis of heterogeneous datasets comprising several data modalities.

Acknowledgments. F. Melchert thanks for support through an Ubbo-Emmius Sandwich Scholarship from the Faculty of Mathematics and Natural Sciences, University of Groningen.

References

1. Backhaus, A., Seiffert, U.: Classification in high-dimensional spectral data: Accuracy vs. interpretability vs. model size. Neurocomputing 131, 15–22 (2014)
2. Biehl, M.: A no-nonsense beginner's tool for GMLVQ. Available online, University of Groningen, `http://www.cs.rug.nl/~biehl/gmlvq`, Ver. 2.2
3. Biehl, M., Hammer, B., Villmann, T.: Distance measures for prototype based classification. In: Grandinetti, L., Petkov, N., Lippert, T. (eds.) BrainComp 2013, Proc. International Workshop on Brain-Inspired Computing, Cetraro/Italy, 2013. Lecture Notes in Computer Science, vol. 8603, pp. 100–116. Springer (2014)
4. Biehl, M., Hammer, B., Villmann, T.: Prototype-based models in machine learning. Wiley Interdisciplinary Reviews: Cognitive Science 7, 92–111 (2016), `http://dx.doi.org/10.1002/wcs.1378`
5. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR time series classification archive (July 2015), `www.cs.ucr.edu/~eamonn/time_series_data/`
6. Driscoll, T.A., Hale, N., Trefethen, L.N.: Chebfun guide. Pafnuty Publ. (2014)
7. Floater, M.S., Hormann, K.: Barycentric rational interpolation with no poles and high rates of approximation. Numerische Mathematik 107(2), 315–331 (2007)
8. Geurts, P.: Pattern extraction for time series classification. In: European Conf. on Principles of Data Mining and Knowledge Discovery. pp. 115–127. Springer (2001)
9. Gil, A., Segura, J., Temme, N.M.: Numerical methods for special functions. Siam (2007)
10. Kohonen, T.: Self-organizing maps. Springer, Berlin (1995)
11. Melchert, F., Seiffert, U., Biehl, M.: Functional representation of prototypes in LVQ and relevance learning. In: Advances in Self-Organizing Maps and Learning Vector Quantization, pp. 317–327. Springer (2016)
12. Papari, G., Bunte, K., Biehl, M.: Waypoint averaging and step size control in learning by gradient descent. Machine Learning Reports MLR-06/2011, 16 (2011)
13. Petitjean, F., Forestier, G., Webb, G.I., Nicholson, A.E., Chen, Y., Keogh, E.: Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. Knowledge and Information Systems 47(1), 1–26 (2016)
14. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing 26(1), 43–49 (1978)
15. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in Learning Vector Quantization. Neural Computation 21, 3532–3561 (2009)
16. Strickert, M., Hammer, B., Villmann, T., Biehl, M.: Regularization and improved interpretation of linear data mappings and adaptive distance measures. In: Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on. pp. 10–17 (April 2013)
17. Strickert, M., Bojer, T., Hammer, B.: Generalized relevance LVQ for time series. In: International Conf. on Artificial Neural Networks. pp. 677–683. Springer (2001)
18. Tomašev, N., Radovanović, M.: Clustering evaluation in high-dimensional data. In: Unsupervised Learning Algorithms, pp. 71–107. Springer (2016)
19. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on Machine learning. pp. 1033–1040. ACM (2006)

Parameterized Pattern Generation via Regression in the Model Space of Echo State Networks

Witali Aswolinskiy and Jochen Steil

Research Institute for Cognition and Robotics - CoR-Lab
Universitätsstraße 25, 33615 Bielefeld, Germany
waswolinskiy@cor-lab.uni-bielefeld.de
https://www.cor-lab.de/

Abstract. Recurrent neural networks capable of sequential pattern generation could facilitate new types of applications like music generation. Here, we explore the capability of echo state networks for parameterized pattern generation and present a new approach utilizing regression in the model space. The goal of the learning is a system that can generate patterns for previously unseen parameterizations. Contrary to other approaches, where a single network is trained to generate all pattern parameterizations, we learn to generate a different network for each pattern parameterization. We evaluate the classical and our modular approach on several synthetic, periodic datasets. We show that regression in the model space of echo state networks can generate parameterized patterns more precisely than a single echo state network.

Keywords: time series generation, pattern generation, echo state network, reservoir computing, model space

1 Introduction

Sequential pattern generation has potentially many applications in signal processing, e.g. filling gaps in time series, computational creativity, e.g music generation and time series modelling. Compared to the main areas of machine learning such as classification, regression and clustering, few advances have been made in pattern generation. The reasons for this include the lack of datasets and benchmarks and the difficulty of training recurrent neural networks, especially to generate stable output. Recently, several variants of Echo State Networks (ESNs, [6]) were applied to a range of pattern generation tasks including frequency modulation [7, 9, 10] and learning human motion [13, 8].

Here, we focus on parameterized pattern generation: Given a set of pattern sequences shaped by control parameters, the goal is to learn to generate patterns for new control parameter values. In a sine wave generator, for example, the control parameter would be the frequency and the goal the generation of a sine wave with a frequency not used during training. This is a more difficult task than

to learn to reproduce patterns, since it involves the learning of the underlying dynamical system producing the patterns and requires the learner to generalize in the space of the control parameters. The solution for this type of task with ESNs, as applied for similar tasks in [7, 9, 10, 13], is to train a single network, which receives the control parameter as input. We propose a different solution, where for each pattern a new network is generated based on the value of the control parameter. This approach is inspired by learning in the model space [3], which was successfully applied to time series classification [4, 2] and to the similar problem of modelling parameterized processes [1].

Fig. 1 visualizes the core architectures of both approaches. In contrast to the classical approach, in our modular approach for each control parameter value the *generalist* creates a *specialist* generator, which is only responsible for generating the corresponding pattern. The *generalist* is responsible for generalizing in the control parameter space, so that the *specialists* can concentrate on generating their specific patterns. Thus, the *generalist* maps the control parameter space to the space of *specialist* models - we refer therefore to our approach in accord with [1] as model space regression (MSR).

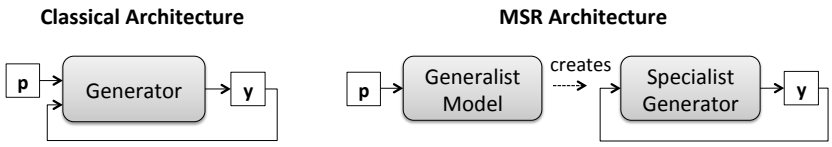


Fig. 1: Pattern generation with a classic (left) and a modular MSR architecture (right). The control parameter $\boldsymbol{p}$ shapes the produced output $\boldsymbol{y}$.

The remainder of this paper is structured as follows. In the next section, we describe the basic, classical ESN pattern generator. In Section 3 we present our modular approach. In Section 4, we compare both approaches on several synthetic datasets. The paper closes with a discussion and some concluding remarks.

2 Echo State Network pattern generator with the control parameters as inputs

An ESN consists of two parts: A reservoir of recurrently connected neurons and a linear readout. The reservoir provides a non-linear fading memory of the inputs. For pattern generation, the network operates with output feedback (cf. Fig. 2). The reservoir states $\boldsymbol{x} \in \mathbb{R}^N$ and readouts $\boldsymbol{y} \in \mathbb{R}^O$ are updated according to:

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

$$\boldsymbol{y}(k) = \hat{\boldsymbol{d}}(k+1) = \boldsymbol{W}^{out}\boldsymbol{x}(k), \quad (2)$$

where $\boldsymbol{u}(k) \in \mathbb{R}^U$ and $\boldsymbol{d}(k) \in \mathbb{R}^O$ with $k = 1, \ldots, K$ are the input and output sequences, respectively; λ is the leak rate, f the activation function, e.g. $tanh$, $\mathbf{W}^{rec}$ the recurrent weight matrix, $\mathbf{W}^{in}$ the input weight matrix, $\mathbf{W}^{out}$ the matrix from the reservoir to the output and $\mathbf{W}^{back}$ the matrix from the output to the reservoir. $\mathbf{W}^{in}$, $\mathbf{W}^{rec}$ and $\mathbf{W}^{back}$ are initialized randomly, scaled and remain fixed. $\boldsymbol{W}^{rec}$ is typically scaled to achieve a spectral radius smaller than one.

The readout is trained to predict the next pattern step, using the training sequence, which is known as teacher forcing [7]:

$$E(\boldsymbol{W}^{out}) = \frac{1}{K-1} \sum_{k=2}^{K} (\boldsymbol{d}(k) - \boldsymbol{W}_i^{out} \boldsymbol{x}(k-1))^2 + \alpha \left\| \boldsymbol{W}^{out} \right\|^2, \tag{3}$$

$$\boldsymbol{W}^{out} = (\boldsymbol{X}^T \boldsymbol{X} + \alpha \boldsymbol{I})^{-1} \boldsymbol{X}^T \boldsymbol{D}, \tag{4}$$

where $\boldsymbol{D}$ are the row-wise collected pattern signal values and $\boldsymbol{X}$ the corresponding reservoir activations. α is the regularization strength.

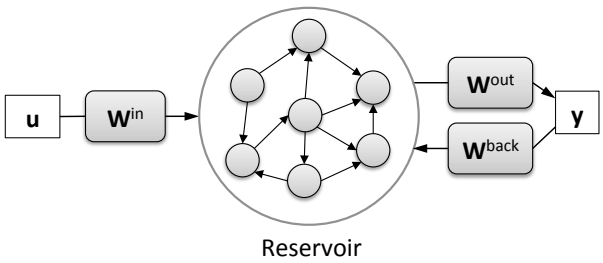


Fig. 2: Echo State Network with input $\boldsymbol{u}$ and output $\boldsymbol{y}$, which is fed back to the reservoir.

During testing (pattern generation) the output $y(k) = \hat{d}(k+1)$ serves as an estimation of the next pattern step and is fed back into the reservoir. In parameterized pattern generation, the input $\boldsymbol{u}$ corresponds to the control parameter, e.g. the sine frequency for a sine wave generator.

3 Parameterized pattern generation via regression in the model space

The training of the MSR architecture is depicted in Fig. 3. It consists of two steps: First, for each pattern, an ESN is trained using teacher forcing. The ESNs are trained independently, but share the same reservoir parameters $\boldsymbol{W}^{rec}$ and $\boldsymbol{W}^{back}$ in order to create a coherent model space. Second, an Extreme Learning Machine (ELM, [5]) is trained as *generalist* to map the control parameters to

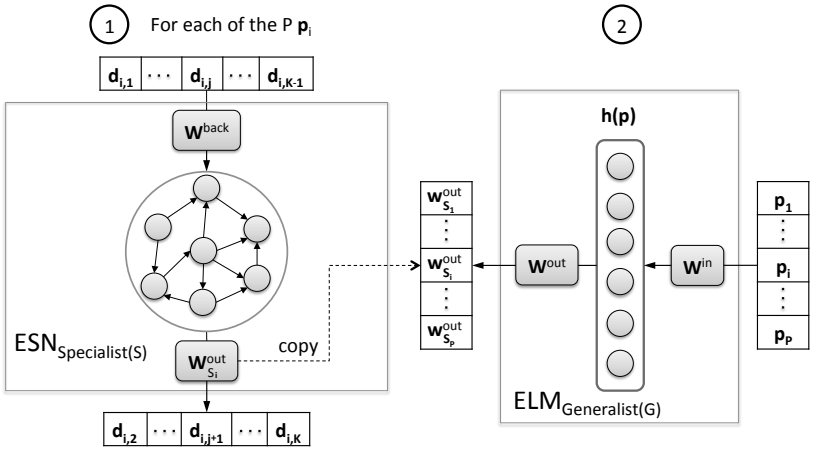


Fig. 3: Training with Model Space Regression (MSR). In the first step, for each of the P control parameters $\boldsymbol{p}_i$ a *specialist* generator is trained using the corresponding pattern outputs $\boldsymbol{d}_{i,j}$, where $j = 1, \ldots, K_i$ is the pattern sample index. In the second step, the *generalist* ELM is trained to map the the control parameters $\boldsymbol{p}_i$ to the ESN readout weights $\boldsymbol{W}_{S_i}^{out}$.

the trained readout weights of the ESNs. The ELM is a two-layer feed-forward network with a random hidden layer and a linear readout layer trained by ridge regression. We chose here the ELM for its simplicity and fast training time - other non-linear regressors like multilayer perceptrons could be used too.

During testing, the *generator* creates from a control parameter value the ESN readout weights. A new ESN is created with the reservoir shared by all ESNs during training and the created readout weights. Then, the created ESN is run autonomously in a feedback loop.

4 Results

We tested the classical ESN pattern generator and our MSR approach on several synthetic datasets. As testing scheme we used leave-one-out-cross-validation (LOOCV), where in P folds, $P-1$ patterns were used for training and the remaining patterns for testing. That is, the trained system, given the control parameter value, had to produce the corresponding pattern from a zero-state.

ESNs have several important hyper-parameters, e.g. input scaling and ridge factor, which have severe effect on the performance. Additionally, in MSR also the *generalist* needs tuning. We performed randomized grid parameter search to find good parameters for both approaches. As metric we used the distance between the target and the generated outputs computed via fast dynamic-time-warping [11].

4.1 Sine wave generation

We consider first a sine wave generator modelling $y = a \cdot sin(b \cdot x)$. The goal of the trained generator is to produce a sine wave with the given amplitude a and frequency b. We vary the frequency in the range $[0.2, 0.6]$ with step size 0.25 and the amplitude in the range $[0.5, 2]$ with step size 0.75. For each pattern, 500 steps were used for training and testing. The last 100 steps of the best LOOCV generation results are shown in Fig. 4. While MSR is able to generate a sine wave with a given amplitude and frequency, the classic ESN generator fails to produce the sine wave with the lowest frequency and highest amplitude (cf. Fig. 4 bottom left).

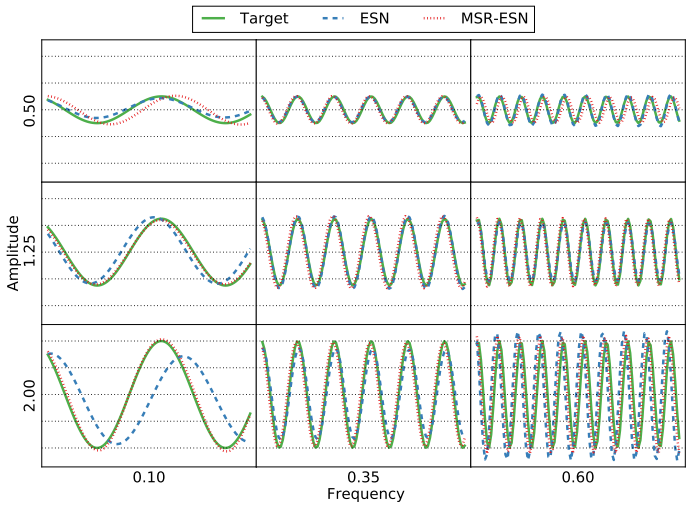


Fig. 4: Sine wave generator with ESN and MSR-ESN. Each cell depicts the LOOCV test generation results over the last 100 generated steps for the denoted frequency and amplitude.

4.2 Skewed figure eight generation

As second task we consider the two-dimensional figure eight pattern:

$$y_1 = sin(x)$$
$$y_2 = a \cdot sin(2x - b) + (1 - a) \cdot cos(x + b),$$

where a controls the shape and b the skewness. When y_2 is plotted over y_1, $(a = 0, b = 0)$ corresponds to a circle and $(a = 1, b = 0)$ to the figure eight. We varied a and b in the range $[0, 1]$ with step size 0.5 and recorded the resulting nine patterns for 300 steps.

MSR produces the target signal with high precision, while the classic ESN shows a relative strong deviation (cf. Fig. 5). A version with a constant b, where only a was varied, posed no problem for either approach.

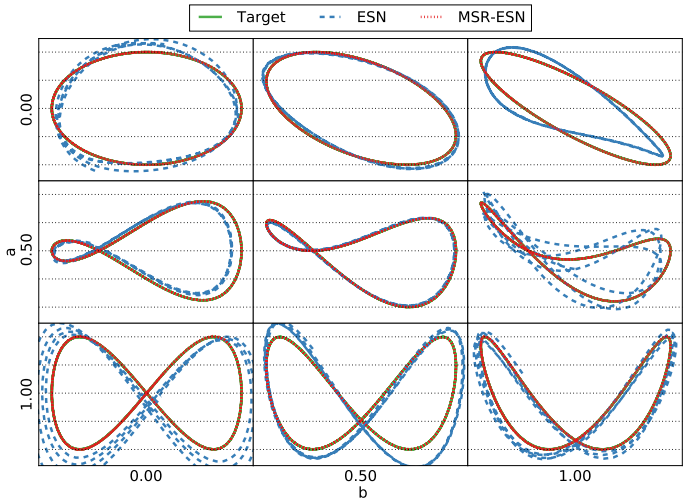


Fig. 5: Figure eight generator with ESN and MSR-ESN. Each cell depicts the LOOCV generation results for the corresponding values of the parameters (a, b).

4.3 Teardrop generation

The teardrop curve is defined as:

$$y_1 = cos(x) \tag{5}$$

$$y_2 = sin(x) \cdot sin^m(0.5x). \tag{6}$$

We varied m uniformly in the range $[2, 10]$ with step size 2 and recorded each pattern for 300 steps. While both ESN and MSR-ESN capture the overall shape, neither is able to create a new curve with precision (cf. Fig. 6).

4.4 More complex tasks

We also experimented with a parameterized multiple superimposed oscillator (P-MSO) in it's simplest form: $y = sin(a \cdot x) + sin(b \cdot x)$, and varied (a, b) in the range $[0.1, 0.6]$. We were, however, unable to train either architecture successfully. This is not surprising, considering that a (non-parameterized) MSO is not an easy task for ESNs and requires additional measures to solve (cf. [12]).

The ability of an ESN to generate each pattern places a natural limit on what can be learned - if an ESN can not learn a single pattern, than it will not

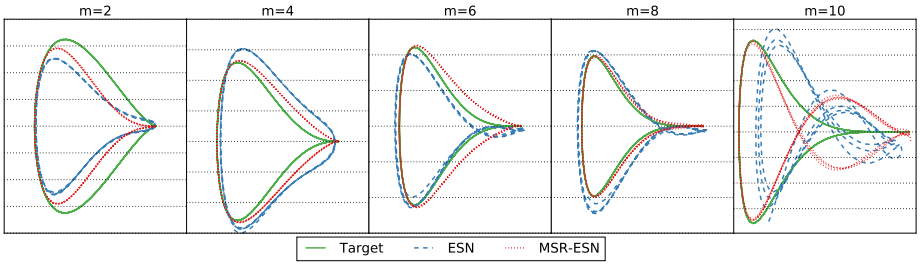


Fig. 6: Teardrop generator with ESN and MSR-ESN. Each cell depicts the LOOCV generation results for the corresponding value of the parameter m.

be possible to generate an ESN which can, or train an ESN to produce multiple patterns.

5 Discussion

The training of a single network to generate different patterns presents two challenges. First, the number of patterns that the network can learn is inherently limited. Similar patterns, as in the case of parameterized patterns, might require less memory, but might also interfere with each other in the state space because of their similarity. Second, the network must be able to change it's output according to the control parameter - it has to be able to reach the attractor corresponding to the control parameter pattern from any state. Both challenges were tackled recently by Jaeger's Conceptors [8]. However, the conceptors were used for morphing between different patterns, and not to learn parameterized patterns - it is unclear, how the conceptor concept can be extended to learn to generate patterns for new control parameter values.

MSR bypasses both challenges by creating networks tailored to each pattern. The basic assumption is, that similar control parameter values result in similar sequences and that the *generalist* can learn this relationship.

6 Conclusion

In this paper we introduced regression in the model space of ESNs for parameterized pattern generation. In contrast to other approaches, where a single ESN is trained to generate different patterns, in our modular approach for each pattern a specialist ESN (more precise: a readout) is created. The specialist ESN then autonomously generates the pattern. An evaluation on several synthetic datasets showed that MSR-ESN can generate parameterized patterns with a higher precision than a single ESN.

The successful application to the synthetic datasets shows that for some tasks the readout weights can be expressed as a function of the control parameters and

learned from few examples. Further research is required to assess whether the recurrent network weights can also be learned from the control parameters and to extend the approach to more complex tasks.

Acknowledgments. This project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition "it's OWL" (intelligent technical systems OstWestfalenLippe) and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication.

References

1. Aswolinskiy, W., Reinhart, F., Steil, J.: Modelling parameterized processes via regression in the model space. In: European Symposium on Artificial Neural Networks (ESANN) (2016)
2. Aswolinskiy, W., Reinhart, F., Steil, J.: Time series classification in reservoir- and model-space: a comparison. In: Workshop on Artificial Neural Networks in Pattern Recognition (ANNPR) (2016), accepted
3. Chen, H., Tino, P., Rodan, A., Yao, X.: Learning in the model space for cognitive fault diagnosis. IEEE Trans. on Neural Networks and Learning Systems 25(1), 124–136 (2014)
4. Chen, H., Tang, F., Tino, P., Yao, X.: Model-based kernel for efficient time series analysis. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 392–400 (2013)
5. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE International Joint Conference on Neural Networks. vol. 2, pp. 985–990 (2004)
6. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. GMD Technical Report 148, 34 (2001)
7. Jaeger, H.: Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the" echo state network" approach. GMD-Forschungszentrum Informationstechnik (2002)
8. Jaeger, H.: Controlling recurrent neural networks by conceptors. arXiv preprint arXiv:1403.3369 (2014)
9. Li, J., Jaeger, H.: Minimal energy control of an esn pattern generator. Jacobs University technical report (26) (2011)
10. Li, J., Waegeman, T., Schrauwen, B., Jaeger, H., et al.: Frequency modulation of large oscillatory neural networks. Biological cybernetics 108(2), 145–157 (2014)
11. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis 11(5), 561–580 (2007)
12. Steil, J.J.: Several ways to solve the mso problem. In: European Symposium on Artificial Neural Networks (ESANN). pp. 489–494 (2007)
13. Wyffels, F., Schrauwen, B.: Design of a central pattern generator using reservoir computing for learning human motion. In: Advanced Technologies for Enhanced Quality of Life, 2009. AT-EQUAL'09. pp. 118–122. IEEE (2009)

Symbolic Association Learning inspired by the Symbol Grounding Problem

Federico Raue[1,2], Marcus Liwicki[1], and Andreas Dengel[1,2]

[1]University of Kaiserslautern, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Germany
{federico.raue,andreas.dengel}@dfki.de, liwicki@cs.uni-kl.de

Abstract. In this work, we present a novel model for a cognitive association task where two visual sequences represent different instances of the same semantic sequence. Also, the model learns the binding between abstract concepts and vectorial representations (e.g., 1-of-K scheme). In this case, the output vector of a network are used as symbolic features, and the network learns to ground the abstract concepts to them. This task is inspired by the *Symbol Grounding Problem*. Our model uses one Long Short Term Memory (LSTM) with an EM-training rule. One important feature of the training is to use one of the two sequences as a target of the other sequence for updating the LSTM network, and vice versa. Our architecture is based on a recent model that uses two LSTM networks for this association task. We compare our model using a generated dataset from MNIST. The presented model reaches similar results against the model with two LSTM networks. Also, our model is compared to a trained LSTM using only one sequence with a predefined binding of the abstract concepts, and the performance is also similar.

1 Introduction

The language development in humans relies on learning the binding between abstract concepts and the physical world. In more detail, the brain encodes the information produced by the sensory input signals, e.g., visual, audio, and haptic. Cognitive Science, Neuroscience, and Artificial Intelligence are exploring this challenging task, which is still an open problem. Harnad [7] denoted *Symbol Grounding* as the mapping between abstract concepts and the physical world.

Moreover, infant development and the *Symbol Grounding Problem* have a relation, which starts learning the *semantic* association between all sensory input signals (mono- and multi-modal) and the recognition for each input signal. As a result, infants are able to classify different representations (visual, audio) of the same semantic entity. In addition, a relation between object recognition (visual) and vocabulary acquisition (audio) is found by Gershkoff-Stowe and Smith [5]. The authors claimed the first hundred words have a visual representation, e.g., dad, mom. Another example of the semantic relation between sensory input signals is found by Asano *et al.* [1]. They mentioned that infant brains show two

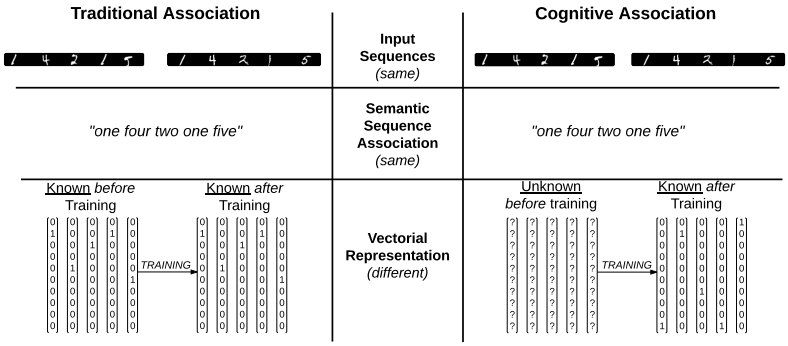


Fig. 1. Differences between components of traditional and *cognitive* association tasks. Consider two independent instances that represent the same semantic sequence. The traditional association has a predefined binding between each semantic concept in the sequence and the vectorial representation before training. In contrast, the cognitive association task includes learning the binding during training. This is similar to the *Symbol Grounding Problem* where abstract concepts are grounded to the physical world.

different activity patterns, which depend on the correct or incorrect semantic relation between visual and audio sensory input signals.

In this work, we are interested in exploiting the semantic relation between two input sequences that express the same semantic sequence. Note that here the semantic sequence is a sequence of abstract concepts. With this in mind, it is possible to use one sequence as a target of the other sequence, and vice versa. In addition, our *cognitive association* task adds a new constraint, in which the semantic concepts are *known*, and their vectorial representation (e.g., *1-of-K scheme*) are *unknown*. In other words, the semantic concepts without a predefined vectorial representation are fed to a classifier, and the classifier learns to bind semantic concepts and vectorial representations. This binding process can be seen as the *Symbol Grounding Problem*. In contrast, a traditional association setup requires the binding between them before training. Figure 1 shows the difference between the traditional and the cognitive association tasks. This cognitive constraint was introduced by Raue *et al.* [11]. Furthermore, the presented model relies on Long Short Term Memory (LSTM), mainly in sequence classification for unsegmented input. Note that unsegmented input means that LSTM does not require to assign a label for each feature vector of the sequence (more information in Section 2). Our contributions in this paper are the following:

- We reduce the complexity of the model proposed by [11]. Our model uses only one instead of two LSTMs. In addition, one LSTM network is robust to handle one sequence input as a target of another sequence, and vice versa.
- We evaluate our model in a dataset that contains pair samples of different instances that express the same semantic sequence. Our model reaches similar results to the model proposed by Raue *et al.* [11]. Also, we compare our

model with a LSTM trained using only one sequence, and the performance of our model reaches similar results.

2 Long Short-Term Memory (LSTM)

Long Short Term Memory (LSTM) is a Recurrent Neural Network that avoid the problem of vanishing gradients for learning long sequences [9, 8]. Moreover, LSTM uses a set of gates in order to read, forget, and update the information through a memory cell. LSTM networks have been applied for learning sequences in several tasks, such as, Neural Machine Translation [2], Image Captioning [13]. In this work, we are interested in exploiting the results in Speech Recognition [6] and OCR [4] for sequence classification in unsegmented input. This has been accomplished by introducing a new layer called *Connectionist Temporal Classification (CTC)*. The target sequence includes a *blank class (b)*, which contributes to align the sequence without pre-segmenting the input sample. For example, the sequence '353' is converted to 'b3b5b3b'. As a result, LSTM aligned its output against a forward-backward algorithm (similar to Hidden Markov Models). In other words, the desired target of CTC layer is a combination of the forward and the backward propagation of probabilities. The label classification is obtained by decoding LSTM output vectors.

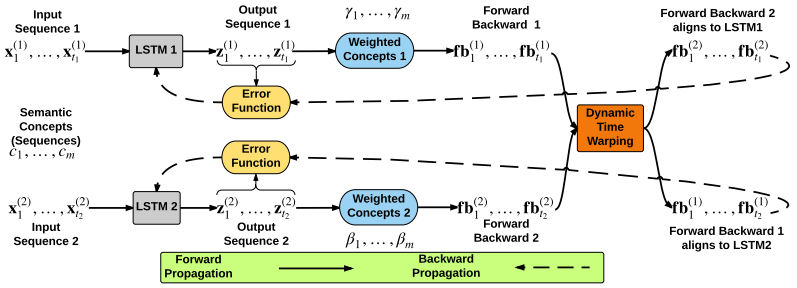
3 Cognitive Association Learning

Our paper is based on the association model proposed by Raue *et al.* [11]. The model associates two text lines[1] that represent the same sequence of semantic concepts. The authors use the network output as symbolic features, and the output of one network is used as a target of the other network. Their model learns the relation between the abstract concepts and the symbolic features. Additionally, both LSTM networks learn to agree on the same association because of the shared semantic relationship. With this in mind, they proposed a model that has two parallel LSTM networks with an EM-training rule and alignment between both LSTM networks. Figure 2a shows a general view of the architecture.

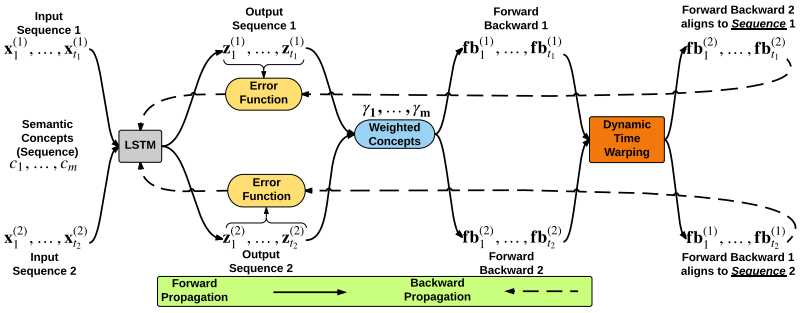
3.1 Association Training

Initially, two input sequences are represented by the vector sequences $\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_{t_1}^{(1)}$ and $\boldsymbol{x}_1^{(2)}, \ldots, \boldsymbol{x}_{t_2}^{(2)} \in \mathbb{R}^n$, where t_1 and t_2 are the sequence lengths. Each input are fed to each LSTM for obtaining the symbolic features represented by the output vectors $\boldsymbol{z}_1^{(1)}, \ldots, \boldsymbol{z}_{t_1}^{(1)}$ and $\boldsymbol{z}_1^{(2)}, \ldots, \boldsymbol{z}_{t_2}^{(2)} \in \mathbb{R}^c$. Note that the semantic concept does not have a vectorial representation. Thus, a set of weighted vectors γ_i and β_i, where $i = 1, \ldots, c$, is introduced for learning the binding between

[1] text line is a visual representation of a text. This term is common use in Document Analysis Field.



(a) Model proposed by Raue *et al.* [11]



(b) Model presented in this work

Fig. 2. Differences with the model proposed by Raue *et al.* [11] and this work. Figure 2a shows two LSTM networks that used the information of one network as a target of the other network. Figure 2b shows the proposed reduction that uses one LSTM network, and one sequence uses the other sequence as a target.

abstract concepts and their vectorial representation in the network (more information in Section 3.2). At this step, the role of the weighted concepts is to generate vector representations for each semantic concept in the sequence for applying the forward-backward algorithm for CTC (*cf.* Section 2). Consequently, each LSTM generates $\boldsymbol{fb}_1^{(1)}, \ldots, \boldsymbol{fb}_{t_1}^{(1)}$ and $\boldsymbol{fb}_1^{(2)}, \ldots, \boldsymbol{fb}_{t_2}^{(2)}$ as the result of the forward-backward algorithm. Then, $LSTM1$ uses the information of $LSTM2$ as a target. We can state that $LSTM1$ is trained $p(\boldsymbol{z}_1^{(1)}, \ldots, \boldsymbol{z}_{t_1}^{(1)} | \boldsymbol{x}_1^{(2)}, \ldots, \boldsymbol{x}_{t_2}^{(2)})$. The model aligns both sequences via *Dynamic Time Warping (DTW)* [3]. In this manner, the output sequence of one network can be approximated to the other network. In other words, the target sequence in $LSTM1$ $\boldsymbol{fb}_1^{(1)}, \ldots, \boldsymbol{fb}_{t_1}^{(1)}$ is converted to $\boldsymbol{fb}_1^{(1)}, \ldots, \boldsymbol{fb}_{t_2}^{(1)}$ as the $LSTM2$ target. Afterwards, the weights of both LSTM networks are updated with the previous approach, and the weighted concepts are updated based on the statistical distribution.

3.2 Mapping Semantic Concepts and Vectorial Representations

The relation between semantic concepts and their vectorial representation is learnt during training. Hence, a set of weighted vectors $\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_c \in \mathbb{R}^c$, where c is the number of semantic concepts. The weighted vectors encode and decode the mentioned relation. For example, the semantic concept "five" could be represented by the index 2 of a one-hot vector where all values are zero except the position 2, which is one. The learning algorithm is based on an EM-approach.

The *E-step* estimates the relation between the semantic concepts and their vectorial representation

$$\hat{\boldsymbol{z}}_i \leftarrow f(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t, \boldsymbol{\gamma}_i), \ \textit{where } i = 1, \ldots, c, \tag{1}$$

$$\hat{\boldsymbol{Z}} = \left[\hat{\boldsymbol{z}}_1 \ldots \hat{\boldsymbol{z}}_c\right], \tag{2}$$

where the $function\ f(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t, \boldsymbol{\gamma}_i)$ is an average weighted sum given the output vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_t$ and the weighted concept vector $\boldsymbol{\gamma}_i$. The intuition of $\hat{\boldsymbol{z}}_i$ is to have an approximation of the probabilistic distribution of all the symbolic features. The matrix $\hat{\boldsymbol{Z}}$ is assembly for each semantic concept. Afterwards, an elimination mechanism is applied for determining the one-to-one relation between semantic concepts (columns) and vectorial representations (rows).

The *M-step* updates the weighted concepts given a statistical distribution as a target. With this in mind, the loss function for each semantic concept is defined by

$$cost_{\gamma_i} = \left(\hat{\boldsymbol{z}}_i - \boldsymbol{\phi}_i\right)^2, \ \textit{where } i = 1, \ldots, c, \tag{3}$$

where $\boldsymbol{\phi}_i$ is the target statistical distribution vector (e.g., uniform distribution). The weighted vectors are updated by gradient descent. The weighted concepts not only ground the semantic concept to the symbolic feature represented by a vectorial representation, but also decodes from the symbolic features to the semantic concepts.

4 Complexity Reduction

In this paper, we propose a complexity reduction of the association model (Section 3). We reduce to one LSTM network for learning the sequence association task. In this case, one sequence is the target of the other sequence that represent the same semantic sequence. The training rule is updated for one LSTM. Initially, both input sequences ($\boldsymbol{x}_1^{(1)}, \ldots, \boldsymbol{x}_{t_1}^{(1)}$ and $\boldsymbol{x}_1^{(2)}, \ldots, \boldsymbol{x}_{t_2}^{(2)}$) are propagated forward to the common LSTM network. The activations from both sequences after the forward pass are stored. Each output sequence ($\boldsymbol{z}_1^{(1)}, \ldots, \boldsymbol{z}_{t_1}^{(1)}$ and $\boldsymbol{z}_1^{(2)}, \ldots, \boldsymbol{z}_{t_2}^{(2)}$) are fed independently to the weighted concept vectors ($\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_c$) based on Equation 1. In contrast to the original model, only one set of weighted concept vectors is required. Afterwards, the CTC-forward-backward algorithm is applied to each output sequence. Both output sequences from the forward-backward step ($\boldsymbol{fb}_1^{(1)}, \ldots, \boldsymbol{fb}_{t_1}^{(1)}$ and $\boldsymbol{fb}_1^{(2)}, \ldots, \boldsymbol{fb}_{t_2}^{(2)}$) are aligned in the time-axis to each other via DTW. In this reduction, the LSTM network has two error functions.

One error function is between the output of sequence 1 and the output from the forward-backward step of sequence 2. The other error function is between the output of sequence 2 and the output from the forward-backward step of sequence 1. Figure 2b shows the presented model, for which the required number of parameters is reduced to half.

5 Experiment Design

Our cognitive association task is between two parallel text lines that represent different instances of the same semantic sequence. We used the same generated dataset in [11], which is based on MNIST [10].
Semantic Sequence Generation: The length of each semantic sequence was randomly chosen between four and eight digits. The training set and testing set had 50,000 samples and 15,000 samples, respectively.
Visual Sequence Generation: After the *semantic concept generation*, two parallel text lines of digits (called *sequence1* and *sequence2*) were generated from MNIST [10]. First, each digit from the semantic sequence was represented with different instances for each sequence input. Second, all digit instances were horizontally stacked for generating the visual sequences. The space between the digits in the text lines has random size. Our training and testing datasets have digits only came from the training and testing set of MNIST, respectively.

We compared the presented model against two different setups that are our baselines. The first setup is the original model proposed by [11], which has two parallel LSTM. The second setup is one LSTM trained on one sequence independently with a pre-defined relation between the semantic labels and their vectorial representation. The parameters of the proposed simplification model are: hidden size was set to 20 memory cells, momentum was set to 0.9, the learning rate of LSTM network was set to 1e-4, and the learning rate for learning the coding scheme was set to 0.001. The pixels of the text lines were normalized between 0.0 and 1.0.

6 Results and Discussion

The cognitive association task is measured by the number of correct elements found on both sequences. We reported the average results of the *Association Accuracy (AAcc)*, which is defined by

$$AAcc = \frac{\sum_{i=1}^{N} LCS(output_i^{(1)}, output_i^{(2)}, gt_i)}{\sum_{i=1}^{N} len(gt_i)}, \tag{4}$$

where $output_i^{(1)}$ and $output_i^{(2)}$ are the label classification of the input sample i, N is the number of samples, gt_i is the ground-truth label, $function\ LCS$ is the length of the longest common sequence between $output_i^{(1)}$, $output_i^{(2)}$, and gt_i; and $len(gt_i)$ is the number of elements in the ground-truth sequence gt_i. In this

Table 1. Average results of the Association Accuracy (%) and Label Error Rate (%). The proposed reduction reaches similar results to the original model and the standard LSTM.

Models	Association Accuracy (%)	Label Error Rate (%)	
		Sequence 1	Sequence 2
LSTM trained for one sequence	93.07 ± 1.47	3.47 ± 0.99	3.52 ± 0.80
original model (Raue *et al.* [11])	95.69 ± 0.27	2.29 ± 0.27	2.21 ± 0.17
proposed reduction (this work)	95.87 ± 0.88	2.12 ± 0.46	2.15 ± 0.43

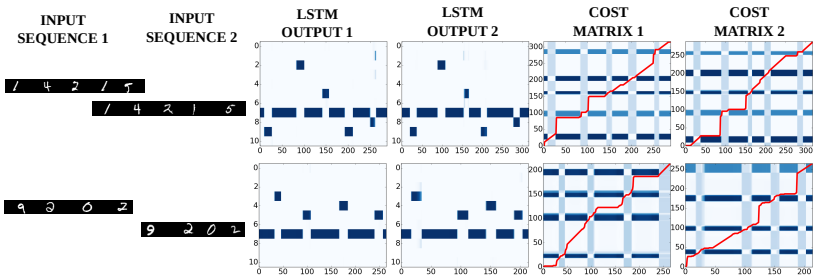


Fig. 3. The presented examples shows our model learns correctly to classify and agrees on the same coding scheme for both sequences. For example, the concept 'one' is represented by the one-hot vector 9 (dark blue). In addition, the cost matrix shows the alignment path (red line) between sequences.

work, we choose 10,000 samples from the training set and 3,000 samples from the testing set. This procedure were repeated ten times. In addition, we also reported *Label Error Rate (LER)* (similar to [6]) for evaluating each sequence independently.

Table 1 shows the performance between the presented reduction and the original model. The proposed reduction reached similar results to the original model with only one LSTM. Also, we compare our model and LSTM networks that were trained using only *Sequence 1* or *Sequence 2*. Note that the neuron activations from one sequence are used for updating the weight connections given another sequence as a target. The training between sequences did not reduce the overall performance of the presented model.

Figure 3 illustrates two output examples. It can be seen that both outputs for each example agree on the same vectorial representation. For instance, semantic concept 'one' of the first example (first row) is represented by the coding vector '9' (dark blue) in both LSTM outputs. In addition, the cost matrix shows the alignment path (red line) between both outputs. In more detail, the path crosses over several areas that represents the alignment under two cases: alignment between blank space and alignment between semantic concepts.

7 Conclusions

In this paper, we were interested in the association of two sequences and learning the symbolic representation at the same time. We proposed a reduction of an association model, where the coding scheme is not defined before training. A new learning rule is introduced where one LSTM was able to agree on the same coding scheme of two independent sequences that were represented by text lines. However, this scenario has some limitations. We will work on more realistic scenarios, where the semantic concepts are not available. In addition, we will work on different types of text lines, such as printed text vs handwritten text. Moreover, we want to evaluate the behavior of our model if one sequence is more dominant than the other sequence. In other words, one sequence is only aligned to the dominant sequence, and the dominant sequences is trained without alignment. Finally, the symbol grounding problem is not a simple task and still is an open problem for understanding the language development [12].

References

1. Asano, M., Imai, M., Kita, S., Kitajo, K., Okada, H., Thierry, G.: Sound symbolism scaffolds language development in preverbal infants. cortex 63, 196–205 (2015)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Berndt, D.J., Clifford, J.: Using Dynamic Time Warping to Find Patterns in Time Series pp. 359–370 (1994)
4. Breuel, T., Ul-Hasan, A., Al-Azawi, M., Shafait, F.: High-performance ocr for printed english and fraktur using lstm networks. In: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. pp. 683–687 (Aug 2013)
5. Gershkoff-Stowe, L., Smith, L.B.: Shape and the first hundred nouns. Child development 75(4), 1098–114 (2004)
6. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification. In: Proceedings of the 23rd international conference on Machine learning - ICML '06. pp. 369–376. ACM Press, New York, New York, USA (2006)
7. Harnad, S.: The symbol grounding problem. Physica D: Nonlinear Phenomena 42(1), 335–346 (1990)
8. Hochreiter, S.: The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 06(02), 107–116 (Apr 1998)
9. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. Neural Computation 9(8), 1735—-1780 (1997)
10. Lecun, Y., Cortes, C.: The MNIST database of handwritten digits
11. Raue, F., Byeon, W., Breuel, T., Liwicki, M.: Parallel Sequence Classification using Recurrent Neural Networks and Alignment. In: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on (2015)
12. Steels, L.: The symbol grounding problem has been solved, so whats next ? Symbols, Embodiment and Meaning. Oxford University Press, Oxford, UK (2005), 223–244 (2008)
13. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. arXiv preprint arXiv:1411.4555 (2014)

Unsupervised Word Discovery from Speech using Bayesian Hierarchical Models

Oliver Walter and Reinhold Häb-Umbach

Paderborn University

Abstract. In this paper we demonstrate an algorithm to learn words from speech using non-parametric Bayesian hierarchical models in an unsupervised setting. We exploit the assumption of a hierarchical structure of speech, namely the formation of spoken words as a sequence of phonemes. We employ the Nested Hierarchical Pitman-Yor Language Model, which allows an a priori unknown and possibly unlimited number of words. We assume the n-gram probabilities of words, the m-gram probabilities of phoneme sequences in words and the phoneme sequences of the words themselves as latent variables to be learned. We evaluate the algorithm on a cross language task using an existing speech recognizer trained on English speech to decode speech in the Xitsonga language supplied for the 2015 ZeroSpeech challenge. We apply the learning algorithm on the resulting phoneme graphs and achieve the highest token precision and F score compared to present systems.

1 Introduction

Automatic speech recognition (ASR) systems mostly rely on supervised learning, with an acoustic model and a language model, trained from transcribed speech and text data. Both, the inventory of words and phonemes are known, and a lexicon with word pronunciations in terms of phoneme sequences is given.

Here we consider a setting, where neither the pronunciation lexicon nor the vocabulary are known in advance, since the acoustic training data come without labels. In general, the phoneme inventory is not know either, however here we use the acoustic models of another language to decode the acoustic data, demonstrating the effectiveness of cross language transfer.

As depicted in Figure 1 an audio recording is typically represented as a time series of feature vectors. A symbolic representation can be learned by discovering repeated sequences of vectors and assigning the same labels to similar sequences, corresponding to phone-like units [1, 19, 17, 13]. On this label sequence again similar sequences are discovered and given labels from another label set, thus arriving at a segmentation into words [18, 8, 4, 5, 7]. An n-gram language model is learned simultaneously and used to calculate the probabilities of words, depending on their $n-1$ preceding words.

Figure 2 depicts the generative model: a language Model $\mathbf{G}$ and the lexicon are generated from a prior process, the Nested Hierarchical Pitman-Yor process. Within the nested process, a word language model is drawn from a Hierarchical

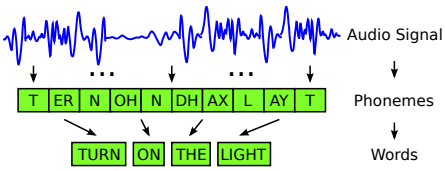


Fig. 1. Hierarchical model of Speech, phonemes and words.

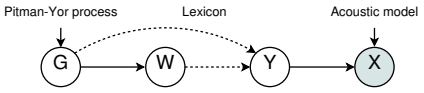


Fig. 2. Language Model **G**, words **W**, phonemes **Y** and feature vectors **X**.

Pitman-Yor process, whose base distribution is a distribution over all possible phoneme sequences, calculated by a phoneme language model. Phoneme Sequences not corresponding to a word in the lexicon, and therefore new words, are obtained as draws from the same phoneme language model whose prior is again a Hierarchical Pitman-Yor process with a uniform base distribution over phonemes. The words **W** are generated (drawn) using the language model and mapped to phoneme sequences **Y** using the lexicon. Acoustic feature vectors **X** are finally generated employing an acoustic model.

Here we will focus on the discovery of words from phoneme sequences, where the phoneme sequences have been generated by a phoneme recognizer, trained with another language, assuming a phoneme set and acoustic models for each of the phonemes to be known.

2 Unsupervised Word Segmentation

If neither the pronunciation lexicon nor the language model are known, and we are left with the task to segment a phoneme string into the most probable word sequence, we have to learn the language model together with the words. We use the Nested Hierarchical Pitman-Yor Language Model (NHPYLM), denoted by **G**, which is a Bayesian language model and allows new, previously unseen words, to evolve and assign probabilities to them. It is based on the Pitman-Yor process prior, which produces power-law distributions that resemble the statistics found in natural languages [14, 15, 7].

An n-gram language model $G_{\mathbf{u}}$ is a categorical distribution over the N words of the vocabulary, conditioned on the $n-1$ preceding words $\mathbf{u} = w_{l-1}, \dots w_{l-n+1}$: $G_{\mathbf{u}} = \{P(w^1|\mathbf{u}), \dots, P(w^N|\mathbf{u})\}$. In a Hierarchical Pitman-Yor process, $G_{\mathbf{u}}$ is modeled as a draw

$$G_{\mathbf{u}} \sim PY(d_{|\mathbf{u}|}, \theta_{|\mathbf{u}|}, G_{\pi(\mathbf{u})}) \tag{1}$$

from a Pitman-Yor process with base measure $G_{\pi(\mathbf{u})}$, strength parameter $d_{|\mathbf{u}|}$ and discount parameter $\Theta_{|\mathbf{u}|}$ [15]. The base measure corresponds to the expected probability distribution of the draws and is set to the language model $G_{\pi(\mathbf{u})}$ of the parent $(n-1)$-gram. This process is repeated until the parent language model is a zerogram, which in the supervised case means that all words have the same probability, given by one over the number of words. Since in the unsupervised setting the vocabulary size is not known in advance, the zerogram cannot be

specified. It is therefore replaced by the likelihood for the word being a phoneme sequence, calculated by a Hierarchical Pitman-Yor Language Model (HPYLM) of phonemes $\mathbf{H}'$ where a hierarchy of phoneme language models is built up to some order m, similar to (1). The phoneme zerogram is finally set to a uniform distribution over the phoneme set. The resulting structure is the NHPYLM, which consists of a HPYLM for words and a HPYLM for phonemes.

Since we now have to learn the NHPYLM along with the words and the phoneme sequence, the maximization problem becomes:

$$\begin{aligned}(\hat{\mathbf{W}}, \hat{\mathbf{G}}, \hat{\mathbf{Y}}) &= \arg\max_{\mathbf{W},\mathbf{G},\mathbf{Y}} P(\mathbf{W}, \mathbf{G}, \mathbf{Y}|\mathbf{X}) \\ &= \arg\max_{\mathbf{W},\mathbf{G},\mathbf{Y}} P(\mathbf{W}, \mathbf{Y}|\mathbf{X}, \mathbf{G}) P(\mathbf{G}) \end{aligned} \tag{2}$$

The Nested Hierarchical Pitman-Yor process prior $P(\mathbf{G})$ over the language model is introduced. Instead of having one particular language model, we have to find that pair of language model, word sequence and phoneme sequence which maximizes the joint probability.

The maximization is carried out by Gibbs sampling, first jointly sampling a word and phoneme sequence from $P(\mathbf{W}, \mathbf{Y}|\mathbf{X}, \mathbf{G})$ [8], by keeping $\mathbf{G}$ constant in (2) and then sampling the NHPYLM from $P(\mathbf{G}|\mathbf{W})$ [7] in an alternating and iterative fashion for each utterance. To avoid the recomputation of the acoustic model scores with every iteration, we use a speech recognizer to produce a phoneme lattice, containing the most likely phoneme sequences.

Joint sampling of the word and phoneme sequence can be very costly. To reduce the computational demand, the phoneme sequence is first sampled from the speech input according to $P(\mathbf{Y}|\mathbf{X}, \mathbf{H})$ and then a word sequence from that phoneme sequence according to $P(\mathbf{W}|\mathbf{Y}, \mathbf{G})$ [5, 4]. For the sampling of the phoneme sequence, an additional phoneme HPYLM $\mathbf{H}$, which includes the word end symbol, is employed. To incorporate knowledge of the learned words, the phoneme HPYLM is sampled from $P(\mathbf{H}|\mathbf{W})$ using the sampled word sequence and their corresponding word sequence.

3 Experiments

We evaluate the segmentation algorithm on datasets provided for the 2015 ZeroSpeech challenge [16]. The datasets consist of an English dataset containing conversational speech from the Buckeye corpus [10] and a second dataset containing prompted speech in Xitsonga, a south African Bantu language, from the NCHLT Xitsonga corpus [2]. Our goal is to demonstrate the possibility of using existing acoustic models from another language to perform the word segmentation, we use acoustic models trained on prompted English speech for both datasets. The English dataset is used to demonstrate the segmentation performance when using acoustic models of the same language. The Xitsonga corpus serves as the low resource language for which we assume to only have audio data available but no transcriptions.

We use the tools provided for the 2015 ZeroSpeech challenge for the evaluation and to be able to compare our results to previous publications. We focus on the type and token scores. The type scores are a measure for the quality of the discovered lexicon and therefore the set of discovered words. The token scores are a measure for the quality of the discovered word tokens and therefore the transcription of the speech, also called parsing quality. A detailed description of the evaluation framework and evaluation measures can be found in [16].

3.1 Setup

For the acoustic model we use a p-norm DNN-HMM triphone speech recognizer [20] trained on English speech from the WSJ0+1 corpus [9]. We build the recognizer using the nnet2 p-norm recipe for WSJ provided with the Kaldi [11] speech recognition toolkit. The recipe was modified to enable phoneme recognition without a word lexicon by building a simple lexicon, mapping each triphone to its middle phoneme.

The recognizer uses LDA transformed 13 dimensional MFCC feature vectors extracted with a frame rate of 10 ms and a context of ± 3 frames at a target dimensionality of 40. FMLLR speaker adaptation of the LDA transformation is performed by a two pass decoding scheme where we assume the speaker ID to be known.

The recognizer is used to create phoneme lattices for both datasets which are processed by the segmentation algorithm. We varied the word- and character language model order in the segmentation algorithm from 1 to 2 (WLM) and 1 to 8 (CLM) to evaluate the performance with different model complexities. Gibbs sampling is performed until iteration 150 to generate the segmentation of a sentence and to update the language model. From iteration 151 Viterbi decoding is performed to generate a segmentation. From iteration 176 the fallback probability to the character model is set to zero to disable the discovery of new words and clean up the language model by removing infrequently, especially uniquely, discovered words. The thresholds were chosen so that in each step the algorithm converged.

3.2 Results

Evaluating the performance of the segmentation algorithm on the Xitsonga dataset delivers insight into its usefulness for low resource language processing. We treat the Xitsonga language as a low resource language by assuming that only audio data is available but no transcriptions. We also assume that no acoustic model is available and instead use the English acoustic model to create phoneme graphs for the segmentation algorithm. This concept is also called cross language transfer, where knowledge from one language is transfered to another.

Figure 3 shows the type F scores for different language model orders and decoding settings. It can be seen that the performance increases with increasing character language model order. The overall scores are fairly low though. This is

mainly due to the mismatch in acoustic models and the resulting errors and noisiness of the phoneme latices. Viterbi decoding delivers a little lower performance although for higher character language model orders it matches the performance with Gibbs sampling. This might partly be due to the noisy characteristics of the input phoneme Lattices. Viterbi decoding is supposed to find the result with the highest probability. Due to the noise this might not be the optimal result. While Gibbs sampling delivers samples from the distribution of segmentations and language models seems to result in better performance. Deactivating the character language model deteriorates the results. Most likely the input data is too noisy resulting in many infrequent words which are being removed in this case. Increasing the word language model order from one to two also does not change the results significantly. The scores are a little higher for the lower order character language models but almost the same for the higher order language models. It seems that word context improves the performance for lower order character language models and noisy input but not for more complex models, contrary to previous results on less noisy data [5].

Figure 4 shows the token F scores. The behavior is similar to the type F score. The result deteriorates with Viterbi decoding and deactivating the character language model. Increasing the word language model order from one to two results in marginally better results. The biggest issue in this low resource setup seems to be the noisy input data making it difficult to learn appropriate models at higher word language model orders.

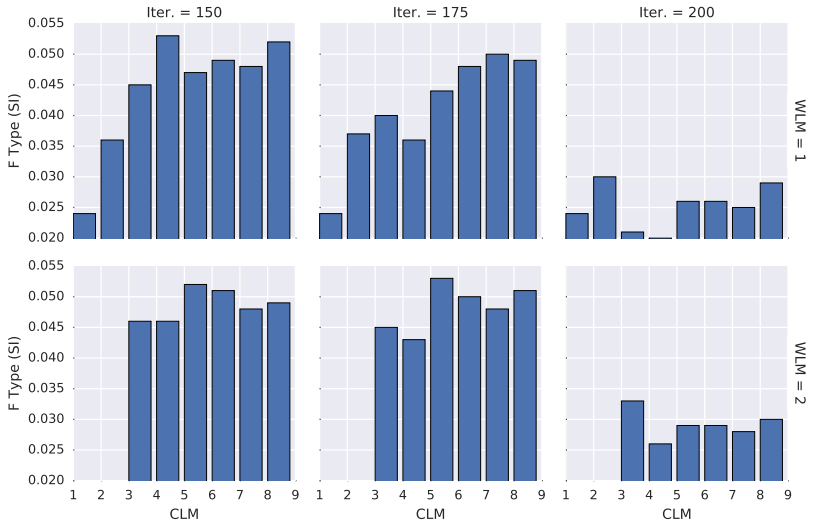


Fig. 3. Type F-score with varying word- and character language model order for Xitsonga dataset. Iter.: 150 (Gibbs), 175 (Viterbi), 200 (No character model fallback)

Fig. 4. Token F-score with varying word- and character language model order for Xitsonga dataset. Iter.: 150 (Gibbs), 175 (Viterbi), 200 (No character model fallback)

3.3 Comparison with previous results

In the 2015 ZeroSpeech challenge two types of systems participated. The two systems can be classified into segmentation systems that segment, cluster and label the complete utterance. Our system also falls into this category. On the other hand Spoken Term Discovery (STD) based systems discover similar segments and only clusters and labels those, leaving segments not discovered as similar to others unlabeled. In Table 1 we compare our results to the two types of systems. For the challenge only two systems were submitted [3] and we compare to the best setups of each.

Osc. Seg. is based on a simple segmentation algorithm finding minima in a particular oscillation frequency of the speech similar to the theta-rhythm brain oscillations and segment according to those. Fixed length representations of the Discovered segments are then clustered, labeled and n-grams of those clusters, sorted in ascending order from longest to shortest, labeled as words [12].

STD is a system based on finding similar segments, building a graph with edges connecting those similar segments with weights proportional to their similarity and clustering them using graph clustering algorithms [6].

For our system we compare the best setups with word language model order one and highest type F score (NHPYLM 1) and word language model oder two and highest token F score (NHPYLM 2) to the other systems. The system performs best in both settings on the English dataset, since we are using English acoustic models. On the Xitsonga dataset our system performs best on the token precision and F score and second best in all three token performance measures.

It also performs better then the Osc. Seg. system. For the type performance our system performs second best in all measures after the STD system. Since our system is a segmentation system it performs better on the token measures while the STD system is able to discover a better lexicon but not label all segments, resulting in higher type measures on the Xitsonga dataset.

Since we are using English acoustic models, the comparison on the English is to be understood as a baseline in case of known and partly matching models.

Table 1. Precision (P), Recall (R), F-score (F) for Type and Token on English and Xitsonga dataset with different algorithms. Red: best score, blue: second best score.

	English						Xitsonga					
	Type			Token			Type			Token		
System	P	R	F	P	R	F	P	R	F	P	R	F
Osc. Seg.	14.1	12.9	13.5	22.6	6.1	9.6	2.2	6.2	3.3	2.3	3.4	2.7
STD	3.1	9.2	4.6	2.4	3.5	2.8	4.9	18.8	7.8	2.2	12.6	0.8
NHPYLM 1	18.1	38.7	24.6	28.8	19.0	22.9	3.9	8.2	5.3	4	2.7	3.2
NHPYLM 2	17.8	36.7	24.0	24.5	25.5	25.0	3.7	8.5	5.1	4.1	3.4	3.7

4 Conclusion

Our system demonstrated a higher performance over a comparable segmentation system while still suffering from noisy input data. Although we achieved better performance than the STD system on the tokes, type quality is still behind STD systems. It is still an open question how to deal with noisy input data. Future research will investigate the integration of the acoustic model into the learning process and how to extend the system to deal with errors in the phoneme lattices and pronunciation variants.

References

1. Chaudhuri, S., Harvilla, M., Raj, B.: Unsupervised learning of acoustic unit descriptors for audio content representation and classification. In: Proc. of Interspeech (2011)
2. De Vries, N.J., Davel, M.H., Badenhorst, J., Basson, W.D., De Wet, F., Barnard, E., De Waal, A.: A smartphone-based asr data collection tool for under-resourced languages. Speech communication 56, 119–131 (2014)
3. Dupoux, E.: The Zero Resource Speech 2015 Challenge Results (2015 (accessed July 14, 2016)), http://www.lscp.net/persons/dupoux/bootphon/zerospeech2014/website/page_5.html
4. Heymann, J., Walter, O., Haeb-Umbach, R., Raj, B.: Unsupervised Word Segmentation from Noisy Input. In: Automatic Speech Recognition and Understanding Workshop (ASRU) (Dec 2013)

5. Heymann, J., Walter, O., Haeb-Umbach, R., Raj, B.: Iterative bayesian word segmentation for unspuervised vocabulary discovery from phoneme lattices. In: 39th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014) (may 2014)
6. Lyzinski, V., Sell, G., Jansen, A.: An evaluation of graph clustering methods for unsupervised term discovery. In: Proceedings of Interspeech (2015)
7. Mochihashi, D., Yamada, T., Ueda, N.: Bayesian unsupervised word segmentation with nested Pitman-Yor language modeling. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1 (2009)
8. Neubig, G., Mimura, M., Kawaharak, T.: Bayesian learning of a language model from continuous speech. IEICE TRANSACTIONS on Information and Systems 95(2) (2012)
9. Paul, D., Baker, J.: The design for the Wall Street Journal-based CSR corpus. In: Proceedings of the workshop on Speech and Natural Language (1992)
10. Pitt, M.A., Dilley, L., Johnson, K., Kiesling, S., Raymond, W., Hume, E., Fosler-Lussier, E.: Buckeye corpus of conversational speech (2nd release). Columbus, OH: Department of Psychology, Ohio State University (2007)
11. Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al.: The kaldi speech recognition toolkit. In: IEEE 2011 workshop on automatic speech recognition and understanding. No. EPFL-CONF-192584, IEEE Signal Processing Society (2011)
12. Räsänen, O., Doyle, G., Frank, M.C.: Unsupervised word discovery from speech using automatic segmentation into syllable-like units. In: Proc. Interspeech (2015)
13. Siu, M.h., Gish, H., Chan, A., Belfield, W., Lowe, S.: Unsupervised training of an hmm-based self-organizing unit recognizer with applications to topic classification and keyword discovery. Computer Speech & Language 28(1), 210–223 (2014)
14. Teh, Y.W.: A Bayesian interpretation of interpolated Kneser-Ney (2006)
15. Teh, Y.W.: A hierarchical Bayesian language model based on Pitman-Yor processes. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics. Association for Computational Linguistics (2006)
16. Versteegh, M., Thiolliere, R., Schatz, T., Cao, X.N., Anguera, X., Jansen, A., Dupoux, E.: The zero resource speech challenge 2015. In: Proceedings of Interspeech (2015)
17. Walter, O., Despotovic, V., Haeb-Umbach, R., Gemmeke, J., Ons, B., Van hamme, H.: An evaluation of unsupervised acoustic model training for a dysarthric speech interface. In: INTERSPEECH 2014 (2014)
18. Walter, O., Haeb-Umbach, R., Chaudhuri, S., Raj, B.: Unsupervised Word Discovery from Phonetic Input Using Nested Pitman-Yor Language Modeling. ICRA Workshop on Autonomous Learning (2013)
19. Walter, O., Korthals, T., Haeb-Umbach, R., Raj, B.: A Hierarchical System For Word Discovery Exploiting DTW-Based Initialization. In: Automatic Speech Recognition and Understanding Workshop (ASRU) (Dec 2013)
20. Zhang, X., Trmal, J., Povey, D., Khudanpur, S.: Improving deep neural network acoustic models using generalized maxout networks. In: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 215–219 (May 2014)

Goal Babbling with Direction Sampling for simultaneous exploration and learning of inverse kinematics of a humanoid robot

Rania Rayyes and Jochen Steil

Research Institute for Cognition and Robotics, Bielefeld University,
Universitätsstr, 33615 Bielefeld, Germany
{rrayyes,jsteil}@cor-lab.uni-bielefeld.de

Abstract. Goal Babbling is a recently introduced method for direct learning of the inverse kinematics within few hundred movements even in high-dimensional sensorimotor spaces. This paper investigates if random selection of movement directions in goal space can be used for Goal Babbling without pre-specifying goals, instead, the goals will be generated along the chosen direction. This so-called Direction Sampling was previously developed for a 2D workspace with a simple planar arm model, whereas we scale it to full 3D and a complex 9-DOF humanoid robot (COmpliant huMANoid - COMAN) integrating simplified walking behavior by means of a simulated robot-floating base. The paper evaluates how much of the workspace can be discovered, what the performance of the learned inverse model is, and how the different degrees of freedom can be constrained by changing the exploration noise model. The results show that the combination of Goal Babbling and Direction Sampling works even under these difficult conditions, but has limitations in performance if the workspace is not fully explored.

Keywords: Exploratory learning, Goal Babbling, Humanoid robot

1 INTRODUCTION

With the advent of humanoid and other robots with many degrees of freedom, motion control and in particular movement skill learning has attracted renewed attention recently. Historically, movement skill learning has been a topic in machine learning, robotics and neuroscience since the 90th, where it is widely accepted that human motor control is organized on the basis of forward and inverse models [1]. A number of schemes have been developed for learning of such internal models, among them the seminal work on distal teachers [2] and on feedback error learning [3]. However, these models were applied to simple robots only and assume that first a forward model is learned or is already available which converts actions into predicted outcomes, before learning an inverse model, that converts goals, e.g. positions to reach, into motor commands. These models cannot describe how to learn from scratch, i.e the first phase of motor learning when a good

body coordination is not yet established. Therefore, a number of works have proposed an initial learning phase to obtain a forward model by random exploration of motor commands under the notion of motor babbling [4], [5]. This appears unrealistic, however, for robots with many degrees of freedom. The respective high-dimensional spaces for motor commands cannot be explored randomly or systematically because of a combinatorial explosion. Furthermore, there is an evidence from infant studies that already neonates perform goal directed action from the very beginning of learning [6]. Apparently, they learn how to reach by trying to reach, and they adapt their motion by iterating their tries [7]. These insights motivated researchers to turn to the idea of direct learning of inverse models [5], [7], [8]. Such models directly yield a motor command to achieve a goal and do not depend on a previously learned forward model. But they have to deal with both the problem of redundancy, which is the problem that a redundant robot has many possible ways to achieve a goal and needs to make a selection from these. And they need to assure the scalability in high dimensions. A particularly efficient has been introduced under the notion of Goal Babbling [9]. Goal Babbling follows the approach to explore rather the low-dimensional space of goals, e.g. target positions in space to be achieved for a robot hand. This is in contrast to exploring the much higher dimensional action space of motor commands that motor babbling explores. Goal Babbling systematically generates consistent samples for supervised learning of the inverse model, for which typically a local linear map [7] or a neural network [10] is employed as learner. It has been shown that Goal Babbling scales to high dimensions (up to 50 DoF for a planar arm [7]), it has been applied to learn the body coordination of the humanoid robot ASIMO [9], and its online version [7] has for instance been applied to learn the inverse kinematics of an soft elephant trunk robot [11] in a truly "learning-while-behaving" fashion.

One limitation of Goal Babbling is that the algorithm needs a predefined set of goals to achieve, for instance a grid of positions to reach in the task space. If the workspace is not fully known a priori or unreachable goals are devised, either only parts of the work space are explored or it can be time consuming to ask the robot to achieve unreachable goals. To overcome this drawback, in [12] an extension of Goal Babbling to discover and determine the reachable workspace while learning the inverse model was introduced as "Direction Sampling". The algorithm is based on random selection of movement directions to explore while learning the inverse kinematic mapping along the way. A planar arm was used for evaluation the effectiveness of this direct sampling. In this case, the workspace is 2D and thus very limited, whereas random directions in 2D are easy to follow. The current paper investigates, if direction sampling can be used for a realistic humanoid robot by simulating the robot COMAN (Compliant Humanoid) that can move in space in order to discover its 3D workspace autonomously. This obviously is a harder problem, which is further complicated by the fact that the robot has very different types of movement available. It can 'walk', which we simulate by means of a simple linear x-y translation in space, and reach with its full upper body with nine degrees of freedom.

Algorithm 1 Online Goal Babbling

INPUT: home postures q_{home}, targets X^*, and forward kinematic function FK.

for number of iteration
  for each target x^*
    generate a temporary path
    for each temporary point along the path x_t^*
      estimate joints' value $\hat{q}_t^*$
      add exploratory noise E: $q_t^+ = \hat{q}_t^* + E(x_t^*, t)$
      $x_t^+ = FK(q_t^+)$
    end for
  end for
end for

OUTPUT: $learner \longleftarrow (q_t^+, x_t^+)$

2 The Goal Babbling Algorithm

The algorithm is given in Algo. 1. Goal babbling starts with an initial inverse estimate g, which has parameters θ adaptable by learning, and is initialized in t = 0 such that it always suggests some comfortable home posture: g(x^* ,θ_0) = const = q^{home} . Then, continuous paths of target positions x_t^* are iteratively chosen by interpolating between the K representative points located on the grid of predefined goals. The system then tries to reach for these targets, which roughly corresponds to infants' early goal-directed movement attempts. For that purpose, the current inverse estimate is used to generate a motor command q_t^*.

The command q_t^* is sent to the robot and executed, the outcomes (q_t^+, x_t^+) are observed, and the parameters θ_t of the inverse estimate are updated online before the next example is generated. It is crucial to make the distinction between q_t^* and q_t^+ at this point: the command q_t^* might not be executable, or might not yet be reached at the time of measurement. Hence, only (q_t^+, x_t^+) but not (q_t^*, x_t^*) represents a sample of the ground truth forward function that is useful for learning. The perturbation term $E(x_t^*, t)$ adds exploratory noise in order to discover new positions or more efficient ways to reach for the targets. This allows to unfold the inverse estimate from the home posture and finally find correct solutions for all positions in the volume of targets X^* spanned by the predefined goals [11]. The most efficient movement will be learned by using the weighting scheme, which helps out to solve the redundancy problem.

For learning, a regression mechanism is needed in order to represent and adapt the inverse estimate $g(x^*)$. The goal directed exploration itself does not require particular knowledge about the functioning of this regressor, such that in principal any regression algorithm can be used. For an incremental online learning, a local-linear map has been chosen. The inverse estimate consists of different linear functions $g^k(x)$, which are centered around prototype vectors and active only in its close vicinity which is defined by a radius d. The function $g(x^*)$ is a linear combination of these local linear functions, weighted by a Gaussian responsibility function [7].

2.1 Direction Sampling

Discovering the workspace could be done by using Motor Babbling, i.e. random motor commands are executed, and their outcomes are observed. However, the robot will discover the workspace without learning it. In contrast, the Goal Babbling uses inverse model which suggests a motor command necessary to achieve a desired outcome and learns it. However, a limitation of Goal Babbling is the need to pre-specify the goals. To this aim, targets must be known beforehand or there is a risk to waste time and to distort the learned inverse model by trying to achieve unreachable targets. To tackle this issue, in [12] Direction Sampling was presented, which is an approach to discover the reachable workspace while learning the inverse kinematic mapping during the discovery. It employs Goal Babbling while generating targets in the workspace instead of predefining them. A random direction $\triangle x$ will be chosen, and the targets will be generated along this path as given in (1):

$$x_t^* = x_{t-1}^* + \frac{\varepsilon}{\|\Delta x\|} \cdot \Delta x, \tag{1}$$

where ε is a step-width, t is a time-step, x_t^* is a generated target, and x_{t-1}^* is the previous one. The robot starts exploration from its home position x^{home}, which is corresponding to some initial joints' values q^{home}. It tries to explore along the desired direction until it reaches an unachievable target i.e. the current position deviates from the desired goal by more than 90 degrees, given in (2):

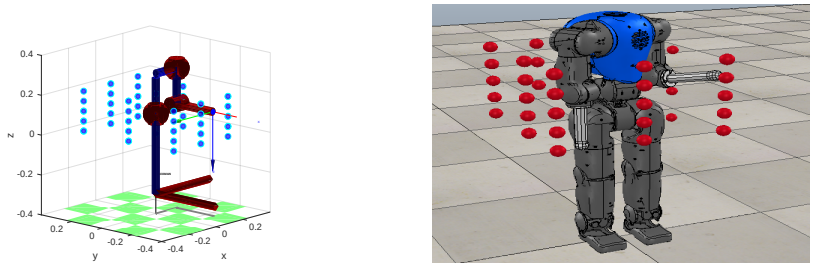
$$(x_t^* - x_{t-1}^*)^T (x_t - x_{t-1}) < 0, \tag{2}$$

where x_t is the current position, and x_{t-1} is the previous observed movement. In this case, a new direction will be chosen and the agent will try to follow it again [12]. Every 100 times the initial position q^{home} is used as a target to avoid drifting. While this mechanism is simple and worked well to explore a 2D workspace, it is not apparent that in full 3D and with a complex robot this mechanism is sufficient to explore a reasonable part of the workspace.

2.2 Noise Scaling

In this section, we introduce a further extension of the Goal Babbling, which is motivated from the idea that not all degrees of freedom should be employed equally much. E.g. walking for a robot can be considered more costly than moving its hand or arm. The previous approach of Goal Babbling already used an efficiency factor to value samples more if they feature more efficient movements. This, however, was purely geometry based, e.g. a shoulder joint needs a smaller deviation to achieve a significant hand movement than an elbow because of the longer lever. But in principle, more factors should be considered such as equilibrium, balance, and motors' synchronization. We therefore try to constrain the learning dynamics to favor solutions that use or avoid certain joints by scaling the exploratory noise for the joints' movement as

$$q_t = g(x_t^*, \theta_t) + E_t(x_t^*) w. \tag{3}$$



(a) (b)

Fig. 1: Compliant humanoid (COMAN) with floating base model in Matlab Robotics toolbox (a) and in VREP (b)

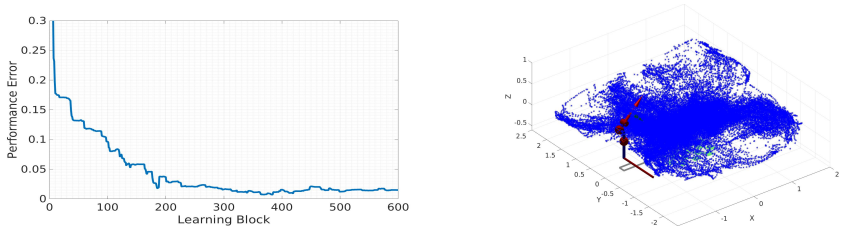
E_t is the exploratory noise weighted by a coefficient vector w. The larger the exploratory noise is in one joint variable i, i.e. the larger the respective w_i, the more likely the learning dynamics will discover a solution for reaching to a point that employs this joint. This implements an implicit, soft constraint. We give highest efficiency for the arm movement, less weight for the torso motion, and the least for the lateral displacement "walking".

3 Setup with the COMAN robot

Unlike standard manipulators, humanoid robots are not physically fixed to a base, there is a so-called floating base. Therefore, the workspace for the humanoid robot is in theory unlimited. However, if we limit the movement to some amount forward and sidewards (in the experiments: $\pm 1.5\ m$), there is a limited reachable workspace around the robot where we can expect interaction of moving, leaning with the upper body and arm motion. We target to discover this reachable workspace with the 3D Direction Sampling approach. Technically, we simulate walking by replacing the actual lower body by two additional degrees of freedom (linear forward, linear sidewards). Therefore, the floating base for the COMAN robot is simplified to move in X-Y plane. The remaining model has 7 DOF: the torso has 3 DOF, the shoulder has 3 DOF, the elbow has 1 DOF. Together with the two virtual DOF for the floating base this is in total a nine dimensional joint space. Note that the types of movement here are very different: linear in the floating base, rotational in the torso and in the arm. The kinematic model has been setup in MATLAB using the Robotic Toolbox [13] and in V-REP for visualization as shown in Fig. 1(a) and Fig. 1(b) respectively.

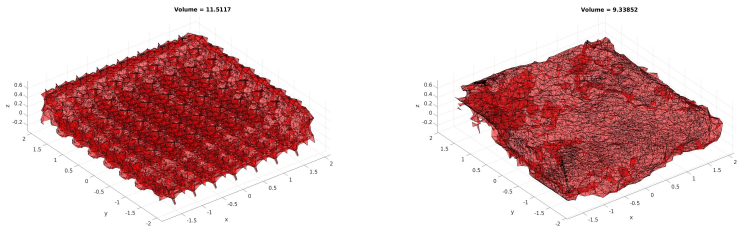
4 Evaluation

In a first step, we verify that Goal Babbling can deal with the complex robot setup and learn to reach 45 targets arranged in a regular 3D grid as illustrated in Fig. 1(a): 15 targets in front of the robot at distance 30 cm, 15 at the coronal plane, and 15 in the back of the robot at distance 30 cm as well. The vertical distance between targets is 5 cm. Fig. 2(a) shows a typical learning curve, the



(a) (b)

Fig. 2: (a) Goal Babbling error in meter, (b) discovered workspace using Direction Sampling



(a) (b)

Fig. 3: Reachable workspace (a) vs Discovered workspace (b)

reaching error drops very fast and already after 200 learning epochs a decent performance on the targets is achieved, i.e. after 800 movements the error drops to 2 mm. The robot leans to use the lateral movement of the floating base to reach to targets behind its body and combines it with the torso and arm movement. Next we turn to Direction Sampling. To obtain a ground truth of the reachable workspace, we use extensive sampling in simulation with a kind of motor babbling to collect 3×10^6 samples. Then the volume of the reachable workspace is estimated using the `alphavol` MATLAB function with radius $R = 0.01$. The estimated volume is 11.5117 m^3 and is illustrated in Fig. 3(a). However, the robot learns nothing about reachable targets in this way. Now, we apply Direction Sampling to explore, discover, and learn the workspace simultaneously. Although the direction sampling is very simple, the robot manages to discover most of the workspace in few thousand steps. Fig. 2(b) illustrates the discovered workspace after 60000 samples. The Direction Sampling algorithm is evaluated after 10^4, 5×10^4, 6×10^4, 10^5, and 10^6 samples. The discovered workspace is again estimated using `alphavol` function. The results are illustrated in Table.1, and the discovered workspace after 10^6 samples is illustrated in Fig. 3(b). As expected, the robot visits an increasing portion of the workspace with more learned samples, and it performs well on the grid targets which were previously used to evaluate the efficiency of standard Goal Babbling, as shown in Table 1.

To gain more insight about the performance relative to the distance from the body, two further target grids for reaching are presented in front of the robot with distance 1 m, and 0.5 m. Then targets are presented in the coronal plane, i.e. some are inside the robot such that it must “walk”, i.e. the lateral movement

Table 1: Volume of discovered workspace averaged over 5 runs

Number of Samples	Average Volume Discovered	Percentage Volume Discovered	Average Error for 45 targets
10^4	0.715 ± 0.07	6.211%	0.377 m
5×10^4	2.17 ± 0.2	18.85%	0.0284 m
6×10^4	3.18 ± 0.02	27.62%	0.0484 m
10^5	3.59 ± 0.01	31.816%	0.047 m
10^6	9.338	81.18%	0.036 m
Goal Babbling	-	-	0.02

Table 2: Testing Error Measured for Different No. of Samples.

	Distance				
	Front		On	Behind	
No. of Samples	-1 m	-0.5 m	0 m	0.5 m	1 m
10^4	0.2091 m	0.16 m	0.17 m	0.42 m	0.2517 m
5×10^4	0.2315 m	0.0234 m	0.02 m	0.074 m	0.1256 m
6×10^4	0.14 m	0.127 m	0.03 m	0.158 m	2.37 m
10^6	0.1020 m	0.0123 m	0.0181 m	1.0625 m	7.17 m

Table 3: Discovered workspace after adding noise scaling

Factor of the scaling noise	Percentage Volume of the Discovered Workspace
[1 1 1 1 1 1 1 1 1]	27.62%
[0.15 0.15 0.5 0.5 0.5 1 1 1 1]	12.5%
[0.1 0.1 0.5 0.5 0.5 1 1 1 1]	10.2%
[0.01 0.01 0.5 0.5 0.5 1 1 1 1]	3.3%

in x-y direction. Finally, they are behind the robot at a distance 0.5 m, and 1 m. The performance error is illustrated in Table. 2. Apparently, the targets behind are much more difficult to reach and in the final row, some of the targets were out of the discovered workspace and produced large errors, as the learner extrapolated rather badly because it is a local linear.

The final experiment is on modulating the learning dynamics to use particular joints more or less. The noise is weighted as shown in Table. 3, which scales down exploration with the floating base (i.e. walking) systematically. The discovered workspace after adding the constrains was evaluated after 60000 samples. The robot discovered less workspace, because of the constrains. For example, 0.01 limit the joint movement exploration more than 0.15 illustrated in Table 3.

5 Conclusion

We have shown that Goal Babbling with or without combination with Direction Sampling can be used even in a complex scenario where a 9 DOF humanoid robot discovers its 3D workspace. There were no indications of local minima or

of the algorithm being captured in already explored areas, which is quite remarkable given the complexity of the mapping to be learned. The results also show, however, that a large number of direction changes are needed and the learner naturally performs badly for goals in the undiscovered areas. It is interesting that indirectly, through scaling of the noise, certain degrees of freedom can be preferred. Future work shall improve the direction sampling. A more active choice of directions towards undiscovered areas should yield better performance, however, at the cost of an increased complexity of the algorithm.

ACKNOWLEDGMENT

R. Rayyes received funding from the German Academic Exchange Service (DAAD)- "Research Grants-Doctoral Programme in Germany" scholarship.

References

1. D. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends Cognit. Sci.*, vol. 2, pp. 338–347, 1998.
2. M. I. Jordan and D. E. Rumelhart, "Forward models: Supervised learning with a distal teacher," *Cognitive Science*, vol. 16, pp. 307–354, 1992.
3. M. Kawato, "Feedback-error-learning neural network for supervised motor learning," *in Advanced Neural Computers. Elsevier*, 1990.
4. Y. Demiris and A. Meltzoff, "The robot in the crib: A developmental analysis of imitation skills in infants and robots," vol. 17, 2008, pp. 43–53.
5. A. Baranes and P. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robot. Auton. Syst.*, vol. 61, no. 1, pp. 49–73, 2013.
6. C. von Hofsten, "An action perspective on motor development," *Trends in CogSci*, vol. 8, p. 266–272, 2004.
7. M. Rolf, J. J. Steil, and M. Gienger, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *IEEE Int. Conf. Development and Learning and on Epigenetic Robotics*, 2011, pp. 1–8.
8. S. V. D'Souza and S. Schaal, "Learning inverse kinematics," *Int. Conf. Intelligent Robots and Systems (IROS)*, vol. 1, pp. 298 – 303, 2001.
9. M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics." *IEEE Trans. Autonomous Mental Development*, vol. 2, no. 3, pp. 216–229, 2010.
10. G. bin Huang, Q. yu Zhu, and C. kheong Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.
11. M. Rolf and J. Steil, "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk," in *IEEE Trans. Neural Networks and Learning Systems*, 2014, pp. 1147–1160.
12. M. Rolf, "Goal babbling with unknown ranges: A direction-sampling approach," in *IEEE Int. Conf. on Development and Learning and on Epigenetic Robotics (ICDL)*, 2013, pp. 1–7.
13. P. Corke, "A robotics toolbox for matlab," *IEEE Robotics & Automation Magazine*, vol. 3, no. 1, pp. 24–32, March 1996.

Virtual optimisation for improved production planning

J. Brinkrolf[1], T. Mittag[2], R. Joppen[2], A. Dröge[3],
K.-H. Pietsch[3], and B. Hammer[1]

1 – University of Bielefeld - CITEC centre of excellence, Germany
2 – University of Paderborn - HNI, Germany
3 – TK Oberfläche, Bielefeld, Germany
{jbrinkro|bhammer}@techfak.uni-bielefeld.de

Abstract. Surface treatment constitutes a particularly energy consuming process, such that its careful planning offers promising optimisation potential. One crucial quantity, which characterises the production process, is given by the number of racks, which are required to mount construction parts for subsequent anodisation or powder coating. Typical orders consist of a list of construction parts together with a specification of their treatment, and an estimate of the overall square meters within the order; orders do not directly reveal the required number of racks. In this contribution, we develop a pipeline, which phrases the mounting of construction parts on racks as a bin packing problem (BPP). After a slight adaptation, it can approximately be solved by the so-called first fit decreasing algorithm. We compute virtual optimisations of exemplary orders, to generate a training set of pairs of orders' square meters and required number of racks. By a simple linear regression, we can infer a mapping of these quantities. As a result, these quantities can be computed from available digital information for subsequent efficient planning. In addition, the proposed BPP solution can serve as a suggestion how to optimally mount a given set of construction parts.

Keywords: Surface treatment, bin packing problem, production planning, optimization

1 Introduction

The buzzword 'industry 4.0' refers to the ever increasing digitalisation of industrial automation processes and production pipelines, including cyber-physical systems, and the internet of things [4]. It carries the promise of smart factories, where intelligent data exchange and adaptive process optimisation enable robust and efficient pipelines for highly individualised manufacturing [5]. While classical process automation and planning already reveals a large optimisation potential, highly individualised processes or products as often faced in modern industry require its efficient data-driven adaptation and optimisation on demand [7, 3]. Recent success stories of data-driven optimisation and modelling range from

high-quality process modelling for ultrasonic wire bonding up to an improved process control based on local sensors [8, 6].

Powder coating and anodisation constitute modern environmentally friendly surface treatments, which require specialised manufacturing lines: construction parts are grouped according to the type of treatment and mounted on racks, which, bundled in small groups, undergo a sequence of processes such as pre-treatment, anodising, etc. The number of required racks directly influences the required processing time, the overall process planing, and the overall energy consumption of the manufacturing pipeline. However, the number of required racks is not available in current orders, rather orders comprise construction parts, their numbers, and the required surface coating only.

In this contribution, we address the particular problem how to mount a given multiset of construction parts on a minimum number of racks. For this purpose, we formalise the process by abstracting from the exact geometrical shape, first, and phrase it as a BPP [2]. We investigate its approximate solution, thereby aiming for the answer of two questions:

1. Do simple efficient approximation schemes of the BPP problem enable good mountings for practically relevant settings? We answer this question by realising an adaptation of the so-called first fit decreasing algorithm, and by a comparison of the achieved results to lower bounds for exemplary orders.
2. Does there exist a simple connection of the square meters of a typical order and the required number of racks? If so, this crucial number could be estimated based on the available digital information, and further production planning could be based thereon. We answer this question by investigating the relation of these quantities for exemplary results, which are obtained by virtual optimisation based on (1).

Now, we first formalise the mounting problem as a BPP. We investigate approximate solutions and their quality for typical benchmarks. Finally, we infer a linear relation of square meters and required number of racks for typical orders using linear regression for the observed examples.

2 Formalisation of the mounting problem

An *order* consists of a multiset S of construction parts, characterised by their unique identifier and the multiplicity of their occurrence in S, corresponding to multiple identical parts being contained in one order. These parts are mounted on racks with suitable width and fixed height. We assume that parts are already ordered in groups according to their surface treatment, i.e. construction parts in the same multiset can be assigned to the same rack, if space permits. The lengths of the parts is irrelevant for the number of batches which occur, since it is accounted for by two or more racks and a fixation of the construction parts at two or more points with different mutual distance. Similarly, the exact shape of the profile of the parts is irrelevant, rather their maximum width and height determines the required space. Hence it is sufficient to describe every part by

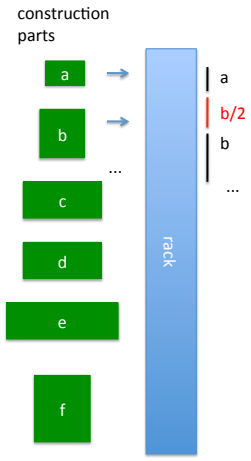


Fig. 1: Example of a mounting problem. Parts are represented via their bounding box only. The distance in between two parts is based on the height of the parts. The maximum width of the parts determines the required width of the rack. The height of the parts determines the number of parts which fit on a given rack.

its bounding box. Mounting parts corresponds to the problem of stacking the parts on the racks while respecting a suitable distance in between parts; see Fig.1 for a schematic display. Thereby, the exact width also becomes irrelevant – the maximum width per rack determines the required width of the entire rack. Hence this problem constitutes a variant of the BPP as follows:

Definition 1. *Assume a multiset $S = \{h_i \mid i = 1, \ldots, p\}$ where $h_i \geq 0$ is the height of part i is given. Since S is a multiset, the element h_i can be contained more than once – corresponding to the fact that a number of identical parts are coated. Assume a distance Δh_i, which is required in between two parts, is fixed. Assume a height $H > 0$ for a rack is fixed. We say that a multiset $S' \subset S$* fits *on H, iff*

$$\sum_{h_i \in S'} (h_i + \Delta h_i) - \max_{\Delta h_i : h_i \in S'} \Delta h_i \leq H \tag{1}$$

The mounting problem *is the problem, to find a number $k \in \mathbb{N}$ and a decomposition of S into multisets*

$$S = S_1 \dot\cup S_2 \dot\cup \ldots \dot\cup S_k \tag{2}$$

such that every S_i fits on H, and k is a minimum number with this property.

In practice, we chose $\Delta h_i := h_i/2$; this choice can easily be changed if the application requires it. Obviously, equation (1) refers to the fact that the parts in S' fit on the rack with height H provided a distance Δh has to be maintained in between two parts. By ordering the part with maximum height h at an end, we safe this distance Δh exactly once. Then the mounting problem simply searches for the minimum number of racks which are required to mount all parts in the multiset S on a rack.

Algorithm 1 FFD algorithm

1: sort $h_1 + \Delta h_1 \geq h_2 + \Delta h_2 \geq \ldots \geq h_{|S|} + \Delta_{h_{|S|}}$
2: **for** $j \leftarrow 1 \ldots |S|$ **do**
3: insert h_j and its space into the first rack which has enough space left

This problem constitutes a variant of the BPP, i.e. the problem is NP hard. We aim for an upper bound on the number of racks in terms of a good assignment, and a lower bound to estimate the quality of the approximation in concrete examples. Note that we can easily determine a lower bound of the number of racks by the minimum required space of all parts together with their distance. Assume distances are sorted in decreasing order $\Delta h_{i_1} \geq \ldots \geq \Delta h_{i_{|S|}}$. Then the lower bound

$$k_{\min} := \min_k \left\{ k \mid \sum_{h_i \in S} h_i + \Delta h_i \leq k \cdot H - \Delta h_{i_1} - \ldots - \Delta h_{i_k} \right\} \tag{3}$$

results – this corresponds to k racks where the largest k distances can be saved by placing the respective parts on the bottom.

There do exist efficient approximation algorithms for the BPP [9, 1]. The *First Fit Decreasing algorithm* (FFD) proceeds as described in (Algorithm 1). It provides a 11/9 approximation for the classical BPP. However, it does not take into account that the largest distance can be omitted per rack. For the latter, we propose a slight variation, the FFD with spaces (FFDS) algorithm, as shown in (Algorithm 2). We will use these algorithms to efficiently generate upper bounds as well as concrete good assignments for the constructions parts to racks.

3 Results

For an evaluation, we investigate 11 concrete orders which were processed in the last year in a company for surface treatment. The orders constitute typical examples of diverse profiles which are required for windows and doors. Characteristics of these orders (number of elements and estimated square meters) are displayed in Tab.1. As a first step, it is necessary to extract bounding boxes for the involved profiles. The profiles and their measures are not contained in the

Algorithm 2 FFDS algorithm

1: sort $\Delta h_{i_1} \geq \ldots \geq \Delta h_{i_{|S|}}$
2: compute $k_{\min}$ as in (3)
3: **for** $j \leftarrow 1 \ldots k_{\min}$ **do**
4: insert h_{i_j} in rack j
5: sort $h_1 + \Delta h_1 \geq h_2 + \Delta h_2 \geq \ldots \geq h_{|S|} + \Delta_{h_{|S|}}$, thereby omitting $i_1 \ldots i_{k_{\min}}$
6: **for** $j \leftarrow 1 \ldots |S| - k_{\min}$ **do**
7: insert h_j and its space into the first rack which has enough space left

order	1	2	3	4	5	6	7	8	9	10	11
number of parts	1303	12	89	194	89	17	46	12	87	18	6
m^2	2405	22	165	356	153	28	51	23	171	40	8
$k_{\min}$	77	1	6	12	6	2	3	1	6	2	1
k_{FFD}	79	1	6	12	6	2	4	1	6	2	1
k_{FFDS}	78	1	6	12	6	2	4	1	6	2	1

Table 1: Exemplary orders, its characteristics, and the obtained lower and upper bounds for the number of racks. For almost all examples the bounds are tight.

orders itself. Rather, every construction part is characterised by an ID which can be accompanied by a digital file displaying its profile in pdf format. Hence we devised a technology to extract bounding boxes of profiles from these pdf files by relying on suitable image processing and pattern recognition technology. The process has been automated such that a correct bounding box could be found for 97% (out of 559) parts. The results have been manually curated, in addition. Due to the required image processing and partial manual curation, that the process of an extraction of the bounding boxes for the given parts is quite costly and not suited for an integration into the online planning and manufacturing process. Rather, it serves as an intermediate step to generate a number of example orders with a provably correct minimum number of required racks.

Afterwards, the height $H = 154$ cm of the rack has been chosen and upper and lower bounds have been computed using the proposed algorithms. Interestingly, in all but two cases the obtained bounds are tight (see Tab.1). That means that the simple FFD algorithm constitutes an excellent heuristic for the considered problem. We verify the feasibility of this modelling by a comparison of the result to the number of racks which have been used in the actual production for the largest order (order number 1). In the manufacturing plan provided by the company, 91 racks have been reported for this order, which corresponds to an increase of about 14% as compared to the optimum. This finding allows two conclusions: the modelling of the mounting process is correct, since a correct size has been obtained for a large order; this result is significant due to the number of involved parts, since random effects for single elements do hardly change the overall result. Further, this finding indicates that there exists the potential to optimise the mounting process in the company by following the proposed optimisation scheme.

4 Interpolation

These results provide a set of 11 example pairs of square meters and their required numbers of racks. We investigate whether these virtual examples enable us to *predict* the required number of racks from the digital information which is available within an order. Each order contains the IDs of the parts which have to be coated as well as the overall number of parts and an estimate of the overall

square meters. Note that, in general, it is impossible to compute the number of required racks from the square meters or the number of parts of an order only, unless additional constraints are present for the order: while square meters are influenced by the length and width of the parts, the required number of racks only depends on the height of the involved parts. That means, in general, there can exist orders with the same number and square meters, but a different number of required racks, provided arbitrary bounding boxes are present in the data. In practice, this is not the case: parts stem from typical series, i.e. orders possess a typical characteristics. Based on the given samples, we investigate whether a relation in between square meters and required number of racks exist for such typical orders.

Due to the small number of available samples, we restrict the model to a simple linear fit, which results from a least squares regression for the given data. A linear regression model trained for all data reveals the function as displayed in Fig.2. An excellent fit can be reached with a mean squared error of 0.3021 and a mean absolute error of 0.376. For all but one sample the absolut error is less than 1 rack.

The feasibility of the obtained model is supported by an evaluation of its generalisation ability in a leave-one-out cross-validation: the mean squared test error equals 0.669 and the mean absolute test error equals 0.578. When rounding the respective output of the linear regression to the next natural number, since no partial racks can be used, the mean absolute test error becomes 0.546. For all but two samples, this absolute error is less than one rack. For two samples, the cross-validation reveals an error of 1.8 racks (see Fig.2): one order with 51 square meters requires a number of 4 instead of predicted 2.29 racks. Further, a large order requires 78 racks instead of predicted 76.21 – note that an error of less than two racks corresponds to less than 3% error in this case, i.e. the model shows excellent extrapolation capabilities for this sample, hence we expect that the inferred function is approximately correct for a wide range of typical orders.

5 Conclusions

We have investigated an optimisation problem within the context of digital factories. In the present case, digital data are available in the form of tables within orders, which change from week to week. The question occurs, whether the process of mounting these orders on racks can be optimised such that the resulting algorithm is easy enough to be useful in practical applications. Further, we have addressed the question whether a prediction of the number of required racks is possible for typical orders based on the available (extremely sparse) digital information only. We have answered both questions positively: by modelling the mounting problem as an instance of the BPP, we have efficient approximation algorithms at our disposal, which can be adapted to the given task, with excellent results. These require the availability of the height for the given parts (which is currently not the case by simple means). Further, we have enhanced typical samples to the full digital information, including bounding boxes, by

means of automated image processing methods (since these are specific for the given parts, we did not further explain this part). This information allows for the virtual optimisation of exemplary orders. As a result, a (small) training set becomes available, based on which a linear regression model can be learned. This establishes a simple relation of the squared meters to the number of racks, with excellent performance also in a leave-one-out cross-validation. Hence the resulting model enables a simple rule of thumb for the required number of racks based on which planning and process optimisation becomes possible for the whole manufacturing process. We would like to stress that the resulting technique is efficient such that their use in online scenarios with changing (but typical) orders per day is possible.

ACKNOWLEDGEMENTS

Funding in the frame of the BMBF leading edge cluster 'it's owl' is gratefully acknowledged.

References

1. G. Dósa. The tight bound of first fit decreasing bin-packing algorithm is ffd(i) $\leq 11/9$ opt(i) + 6/9. In *Proceedings of the First International Conference on Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, ESCAPE'07, pages 1–11, Berlin, Heidelberg, 2007. Springer-Verlag.
2. B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms*. Springer Publishing Company, Incorporated, 4th edition, 2007.

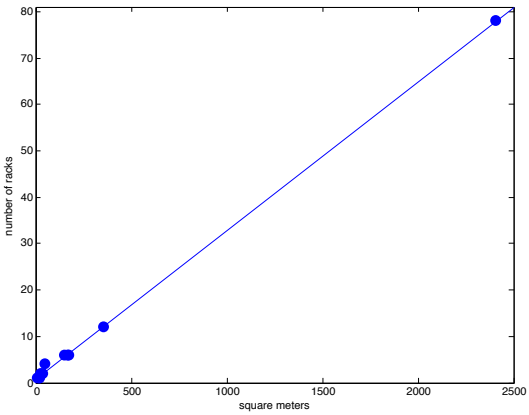


Fig. 2: Linear least squares regression with inputs given by square meters and outputs as the estimated number of racks. The mean square error is 0.3025, the absolute error being smaller than 1 for all but one sample.

3. D. Kreimeier, K.-F. Seitz, and P. Nyhuis. 5th conference on learning factories cyber-physical production systems combined with logistic models – a learning factory concept for an improved production planning and control. *Procedia CIRP*, 32:92 – 97, 2015.
4. J. Lee, B. Bagheri, and H.-A. Kao. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 3:18–23, 2015.
5. O. Niggemann and C. W. Frey. Data-driven anomaly detection in cyber-physical production systems. *Automatisierungstechnik*, 63(10):821–832, 2015.
6. F. Oestersötebier, P. Traphoener, F. Reinhart, S. Wessels, and A. Trächtler. Design and implementation of intelligent control software for a dough kneader. In *Proceedings of the 3rd International Conference on System-Integrated Intelligence*, Paderborn, 2016. Elsevier.
7. J. J. Solberg. Production planning and scheduling in CIM. In *IFIP Congress*, pages 919–925, 1989.
8. A. Unger, W. Sextro, S. Althoff, T. Meyer, M. Brökelmann, K. Neumann, R. F. Reinhart, K. Guth, and D. Bolowski. Data-driven modeling of the ultrasonic softening effect for robust copper wire bonding. In *Proceedings of 8th International Conference on Integrated Power Electronic Systems*, volume 141, pages 175–180, 2014.
9. M. Yue and L. Zhang. A simple proof of the inequality $mffd(l) \leq 71/60opt(l) + 1, l$ for the mffd bin-packing algorithm. *Acta Mathematicae Applicatae Sinica*, 11(3):318–330, 1995.

The Artificial Mind's Eye

Resisting Adversarials for Convolutional Neural Networks using Internal Projection

Harm Berntsen[1], Wouter Kuijper[1], and Tom Heskes[2]

[1] Nedap N.V., Security Management
Groenlo, The Netherlands
{harm.berntsen,wouter.kuijper}@nedap.com
[2] Radboud University, Institute for Computing and Information Sciences
Nijmegen, The Netherlands
t.heskes@science.ru.nl

Abstract. We introduce a novel artificial neural network architecture that integrates robustness to adversarial input in the network structure. The main idea of our approach is to force the network to make predictions on what the given instance of the class under consideration would look like and subsequently test those predictions. By forcing the network to redraw the relevant parts of the image and subsequently comparing this new image to the original, we are having the network give a "proof" of the presence of the object.

1 Introduction

Convolutional Neural Networks (CNNs) have been shown to work well on image classification tasks [19]. However, CNNs are vulnerable to adversarial images [16, 23]. In this paper we introduce a novel type of network structure and training procedure that results in classifiers that are provably, quantitatively more robust to adversarial samples. Adversarial images can be found by perturbing a normal image in such a subtle way that the change is usually imperceptible by the naked eye [7, 23].

The main idea of our approach is to force the network to make predictions on what the given instance of the class under consideration would look like and subsequently test those predictions. Technically we achieve this by chopping the classifier network into three stages: estimation, projection and comparison.

The first stage estimates a vector of parameters (displacement, rotation, scale and, possibly, various object specific internal deformations) from the image. The second stage generates an image based on the estimated parameters. The third stage compares the projected image with the actual image and delivers a likelihood value which can be turned into a verdict using a threshold. The working hypothesis is that this network structure improves robustness against adversarial samples.

There are two intuitions behind this working hypothesis. The first is that parameter estimation is a *smoother* task than classification. Meaning that an

orbit through the multidimensional output space can be expected to have a smooth corresponding orbit through the input space. In other words: it is possible to meaningfully interpolate parameters *for a model of a given class* but it is much harder to meaningfully interpolate between *models of two or more different classes*.

The second intuition behind our working hypothesis is that, by forcing the network to draw a new image using only the estimated parameter vector and subsequently comparing this new image to the original, we are having the network give a "proof" of the presence of the object. By carrying out this comparison only through myopic, local features we ensure that, in order to get enough probability mass to make the threshold, the network must be fairly precise in reproducing the internal details of the objects. In effect we force the network to learn much more than just a discerning set of features, we force it to learn also the detailed internal structure of the object, thereby making it inherently more robust against adversarial input.

In this paper we lay the conceptual groundwork and give initial experimental results. We hope that this will enable further research on combining our approach with other, orthogonal, approaches like adversarial training [7] and on applying this method, or refinements inspired by it, on real- world tasks.

The source code for training the networks and to generate adversarial images is available at `https://github.com/hberntsen/resisting-adversarials`.

1.1 Related Work

Neural networks recognise objects in a different way than humans. As Ullman et al. [26] point out: "...the human recognition system uses features and learning processes, which are critical for recognition, but are not used by current models". They show that where humans can recognise internal components of the objects in the image, current neural networks do not. With knowledge about the internal representation of the objects, false detections can be rejected when it is not consistent with the internal representation of the object. This corresponds with the sensitivity to adversarial images with an imperceptible change that have been shown in [23] and various work since [16]. They show that the smoothness assumption does not hold for neural networks; an imperceptible change in the query image can flip the classification. Goodfellow et al. argue that the primary cause for this is the linear behaviour of the networks in high-dimensional spaces [7] as opposite to the nonlinearity suspected in [23]. The adversarial images are not isolated, spurious points in the pixel space but appear in large regions of the space [24]. Moreover, adversarial images can be efficiently computed using gradient ascent, starting from any input [7].

Though the existence of adversarial examples is universal [23], neural networks can be made more robust against them. One way is to include adversarial examples in the training data [7, 10, 14, 23], e.g. by assigning them to an additional rubbish class. Apart from increasing the robustness it can also increase the accuracy on non-adversarial examples. Another approach is to adapt the model of the network to improve robustness [4, 8]. In [4] the authors identify features

that are causally related with the classes. Their learning procedure could be seen as a way to train a classifier that is robust against adversarial examples. In [8] the authors test several denoising architectures to reduce the effects of adversarial examples. They conclude that the sensitivity is more related to the training procedure and objective function than to model topology and present a new training procedure.

We use 3D models to train the classifier. Though this is artificial data, it can be used as training material for real data, e.g. for object detection [18, 22] or even aligning 3D models within an 2D image [1, 15, 21]. The work of [1] does this using HOG descriptors, while [15, 21] use neural networks. They have trained a CNN to predict the viewpoint of 3D models and were successful in applying this model to real-world images.

2 Network Architectures

In this section we describe the network architectures that we use to test the robustness of our approach. The task of each network is the same: classify the image. Our data consists of greyscale ImageNet images where a part of the image is overlaid with an alpha-blended instance of a 3D model. We use three 3D models that are parametrised by their Euler rotation. The neural network has to recognise the 3D models in all those rotations and emit which 3D model, if any, is visible in the query image. We compare the robustness against adversarial images using three concrete network structures. We will refer to the three 3D models as positive classes and refer to the 'None' class as the negative class.

2.1 Networks

Direct Classification To set a baseline, we train a network to map the query images directly to a probability distribution over the classes. This network is based on AlexNet [12], which has been shown to work well in various situations [17, 20, 21, 25]. To adapt AlexNet to a reduced set of classes and smaller query image, we use a reduced version of AlexNet from [2] which uses smaller layers. We replaced the last layer with a softmax classification layer. The softmax layer has four outputs, three for the positive classes and one to indicate the negative class.

Direct Classification + Parameter Estimation The Direct Classification + Parameter Estimation network is a variant of the Direct Classification network that has an additional output: the parameters of the model. This additional output forces the neural network to develop a better understanding of the 3D models it has to recognise. The parameter estimation is only used to guide the training process and is not used after the network has been trained.

Triple-staged We will first describe the triple-staged network as if it is specific to one single class. We expand this design later to a configuration for multiple classes. The triple-staged network contains three stages: (1) estimation, (2) projection, (3) comparison, shown in Fig. 1. Each stage was trained separately and finally merged into one network.

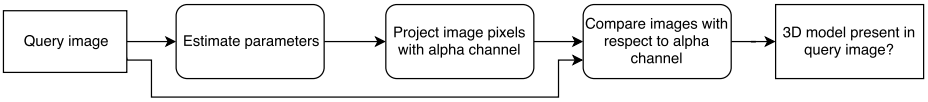


Fig. 1: Data flow diagram of the triple-staged network structure for a single 3D model. The model parameters are estimated from the input image, converted back to an image and then compared to the original image.

The first stage maps the query image to a parameter vector that describes a 3D model. In our running example the parameters describe just the Euler rotation of a 3D model but in general this can also include scale, pan, and internal parameters such as dimensions, and rotational and linear joints. The estimations are clipped to their valid range. The network structure of this stage is the same as the direct classification + parameter estimation network without the task to predict the class.

The second stage of the network projects the parameter vector to a 2D image that contains the rendered 3D model in front of a black background. The alpha channel indicates to which degree each pixel belongs to the 3D model. In [5], it was shown that a deep, deconvolutional neural network can be trained to generate images that are parametrised by a broad set of classes and viewpoints. Due to our smaller set of classes and parameters, we use a downscaled variant of the 1s-S network from [5]. The first and second stage together form an autoencoder where the bottleneck contains an understandable instantiation vector of the input. This concept was already applied in the context of transforming autoencoders [9].

The final stage has to compare each projected image with the query image. Here we follow [27], which shows how to compare image patches using CNNs. We adapted the 2-channel structure to create a network that compares 10×10 pixel image patches with respect to the alpha channel. This network is convoluted over the output of the second stage, giving it only local data to work with. The network was trained to emit a binary output that indicates whether the original and projected image patch should be considered equal. Fig. 2 visualises how this network works.

The combination of the three stages is capable of determining the presence of a single class in the query image. This does not scale well since a separate network has to be trained for each and every class. This issue is addressed by adding the class to generate as a parameter to the projection stage of the network. Each stage can now be trained on data of all the 3D models at once. The

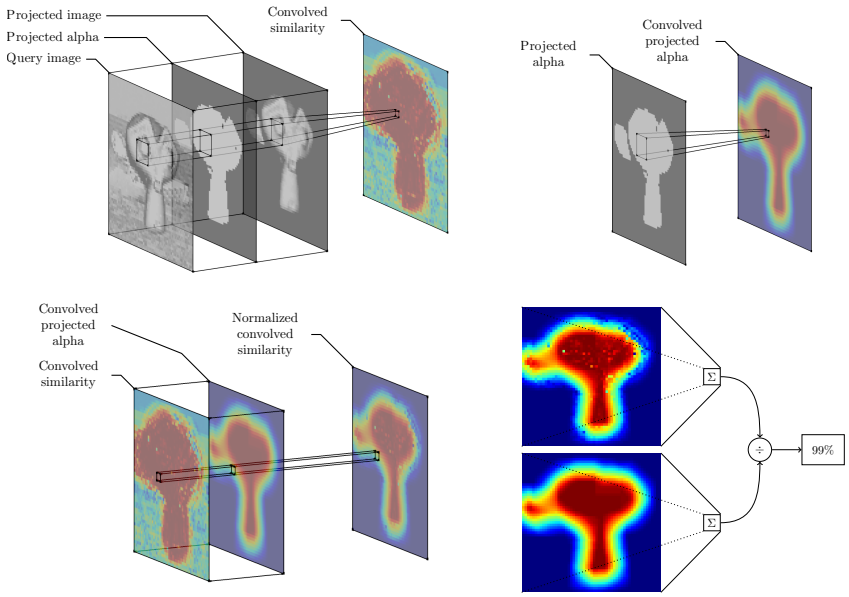


Fig. 2: Visualisation of the comparison stage of the network. We convolute a network that compares a 10×10 pixels patch with stride 1 (top left). This generates the *convoluted similarity map* that shows which areas of the image should be considered as being the same. Next to this similarity map, we apply a 10×10 average pooling layer over the projected alpha channel to generate the *convoluted, projected alpha channel* (top right). We can then directly multiply the convoluted alpha channel and the convoluted similarity map to end up with the *normalized, convoluted similarity map* (bottom left). Next we feed both the normalized, convoluted similarity map and the convoluted, projected alpha channel, as a whole, into a single-output, sum–reduction layer. Finally we obtain the ratio by having a single output multiply the similarity sum with the reciprocal of the total weight sum (bottom right). To obtain a final verdict we apply a threshold Θ over the output (cf. Section 4).

class parameter improves scalability of the triple-staged network since only one network needs to be trained for multiple classes. To generate a classification for a query image, it is provided as an input to the network multiple times with a different class parameter. If there is any class where the output of the network rises above the threshold Θ, we use the class that yields the highest similarity score. Otherwise we judge that none of the classes are visible in the query image.

Similarly, an optimisation we applied is to supply the class parameter to the prediction stage of the network. We add the class as additional binary channels to the query image. Without the class information passed to this network, the network would internally need to determine which class is visible in the query image. We found that supplying the class information to the network increased

the robustness of the triple-staged network. Figure 3 shows the triple-staged network architecture we used.

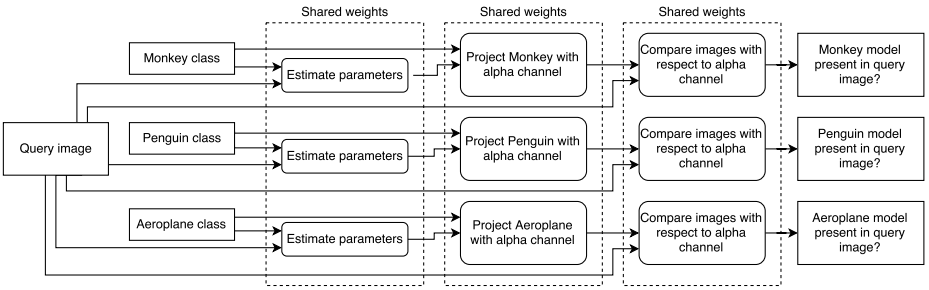


Fig. 3: Usage of the triple-staged network. A classification is obtained by passing the query image three times through a single network, each time with a different reference class given as extra input. Since these passes are completely independent they could also be parallelised.

2.2 Rationale

The main feature of the triple-staged architecture is to be robust against adversarial samples that cause the network to indicate that a certain class is visible when it is not. To let the neural network produce a false positive classification, an adversary needs to perturb the image such that it ultimately fools the comparator stage of the network. However, in order to do so, it must pass through both the estimator and the projector stage. Any attempt to generate a false positive will start drawing another 3D model over the existing one because the comparator compares the query image with the stable internal projection of the class in question. Ultimately then, the 'false positive' class will be evidently visible in the query image.

Since the comparator network directly consumes the query image, this network could still be susceptible to adversarial perturbations of the query image in much the same way as a normal classifier would be. In order to reduce susceptibility, we limited the input space of the network to a single 10×10 pixel patch. This is enough to learn the general concept of two patches being "similar" (modulo some minor deviations and/or artefacts) but it is not enough to learn longer range correlations in the query image (that would give the adversarial a clear gradient to follow in generating adversarial input). We convolve this local network across the whole image, hence an adversary would need to simultaneously fool sufficiently many individual, local patch comparisons to make a significant impact on the overall similarity mass.

3 Experiment Set-up

In this section we describe how we will test the network architectures from the previous section for their robustness.

3.1 Training method

As objects to recognise, we use parametrised 3D models. We rendered 64×64 pixel greyscale images of three 3D models: a Monkey (the Suzanne model from [3]), Penguin [13] and an Aeroplane [6] using Blender [3]. We took the rotation of the 3D model over three axes as our parameter space though our method is not limited to this. The rotations were uniformly sampled from the range of $[-0.5, 0.5]$ radians. To give the 3D models a 'natural' background, we use alpha-composition to blend the 3D model in front of randomly sampled images from the ImageNet dataset [19]. This reduces overfitting of the network on the otherwise black background. We generated 4×10^4 samples for each class. The None class simply consists of random ImageNet images.

We used Caffe [11] for the network implementations. The direct classification networks were trained using all 16×10^4 samples. We left the predicted parameter vector for the negative samples undefined. Each stage of the triple-staged was trained separately. The estimator stage was only trained on the subset of positive samples. The second stage was trained on the original data as rendered through Blender. The input of this stage consists of the binary encoding of the class and the rotation parameters.

The data for the third stage of the network was generated by passing data through the first two stages of the network. This resulted in a new dataset with the query image, ground truth class, projected image and projected alpha mask. From this data we generated a balanced dataset where half of the samples should be considered the same and the other half of the samples is not. The samples that are considered different compare the query image, which is either a random ImageNet image or one of the 3D models in front of an ImageNet background, against one of the projected images by the second stage of the network. From this training set we sampled 10×10 pixel patches where the projected alpha mask indicated that at least 1% of the pixels belonged to the model. The other samples do not matter since their comparison is cancelled out by the multiplication with the projected alpha (visualised in Fig. 2).

3.2 Adversarial Image Generation

When we want to generate an adversarial query image $\tilde{x}$, we search for a minimal perturbation of the original image x that is sufficient to flip the classifier towards a chosen adversarial target class value y. To do this we adopt the fast gradient sign method of [7]. The fast gradient sign method can efficiently generate adversarial images using backpropagation. Our aim is to generate adversarial

samples that flip the classification to another positive class value y. Specifically, we perturb an image by computing

$$\tilde{x} = \text{clip}(x - \text{sign}(\nabla_x J(\theta, x, y)), 0, 255),$$

with J the loss function over the query image x and network parameters θ. Since we use 8-bit greyscale images with a range of $[0, 255]$, each pixel of the image will be minimally perturbed. This function is applied as often as needed to flip the classification of the network to the target y.

4 Results

We test our networks on a separate test set which consists of 10000 samples of each class. The backgrounds are sourced from the ImageNet validation set. We first measure the classification performance of the networks on non-adversarial images, see Table 1 for the results. The direct classification networks have the lowest error rate, followed by the triple-staged network. With $\Theta = 0.2$ or $\Theta = 0.7$, the 'None' class is chosen more/less often. Although choosing Θ somewhere in the middle seems to be the best option purely in terms of minimizing classification error, our data clearly shows that there is a trade-off to consider concerning robustness to adversarial samples versus classification error.

Table 1: Classification error rate of the networks.

Network	Error rate
Direct classification	0.01%
Direct classification + Parameter Estimation	0.01%
Triple-staged, $\Theta = 0.2$	1.34%
Triple-staged, $\Theta = 0.45$	0.57%
Triple-staged, $\Theta = 0.7$	3.09%

To compare the networks under adversarial conditions, we measure how many iterations of adversarial perturbation it takes to change the classification and how much the image was changed. Here we follow [23] who measure the amount of perturbation in adversarial sample for original sample as distortion which is defined as:

$$\frac{1}{255}\sqrt{\frac{\sum_i (\tilde{x}_i - x_i)^2}{n}},$$

where x is the original image, $\tilde{x}$ is the distorted image and n is the number of pixels.

We performed experiments with both false positive and false negative adversarial images. To generate a false positive adversarial image, we start from a test image containing one of our 3D objects, say a Monkey. Following the procedure

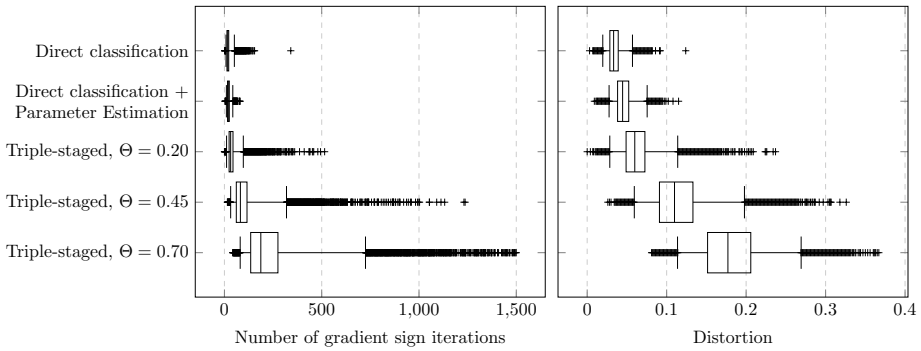
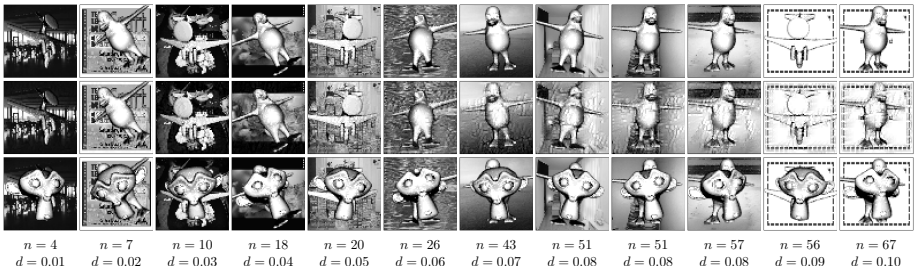


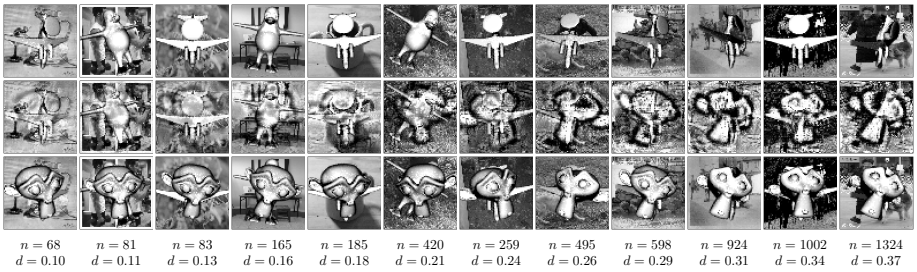
Fig. 4: The effort required to convert the test images that contain one of the 3D models to an image where the network judges that another class of the 3D models is visible. There were no cases with 0 steps. For readability we clipped the number of steps in the graph to a maximum of 1500. The whiskers are placed at the 2^{nd} and 98^{th} percentile.

explained in Section 3.2, we then construct an adversarial image that makes the network believe that the image belongs to the other class, once for a Penguin and once for an Aeroplane. We repeat this procedure for all test images. Figure 4 shows the results for the false positive adversarial images. The figure shows that for the direct classification networks the required changes are limited: the median of their distortion is still below 0.1 which indicates that the adversarial image is still very similar to the original one. The examples in Fig. 5a show this. In contrast, the triple-staged network requires significantly more effort to change the classification. The higher the threshold, the more the query image needs to look like the internally projected image. Figure 5b shows false positive adversarial samples for the $\Theta = 0.70$ network. The triple-staged network structure requires the adversary to generate images that really start to look like the adversary class. As Table 1 shows, the error rate on normal samples is still reasonable at this threshold.

When we generate false positives, we start with an image that contains one of the classes and perturb it to an image that is classified as 'None'. For the direct classification networks this is the 4^{th} class they can predict. In the case of the triple-staged network, the output for every class has to be below the threshold Θ. Figure 6 shows that the number of required iterations is significantly higher for $\Theta = 0.20$ compared to the other networks. Note that in contrast to the false positives, the false negative adversarial samples are better resisted using a lower threshold. By lowering the threshold, the triple-staged network is less likely to switch to the 'None' class, requiring more work from the adversary.



(a) Direct classification + parameter estimation



(b) Triple-staged network with $\Theta = 0.7$

Fig. 5: Generated adversarial images, ordered by distortion. The top row contains the original image, the second row the adversarial variant. The bottom row contains the original image with the Monkey 3D model alpha-blended, rendered by Blender using the predicted parameters. Every column is annotated with the distortion d and number of iterations n. All adversarial samples are classified as the Monkey class. The adversarial images in (a) do not look like the Monkey class at all while in (b) the shape of the monkey is clearly visible.

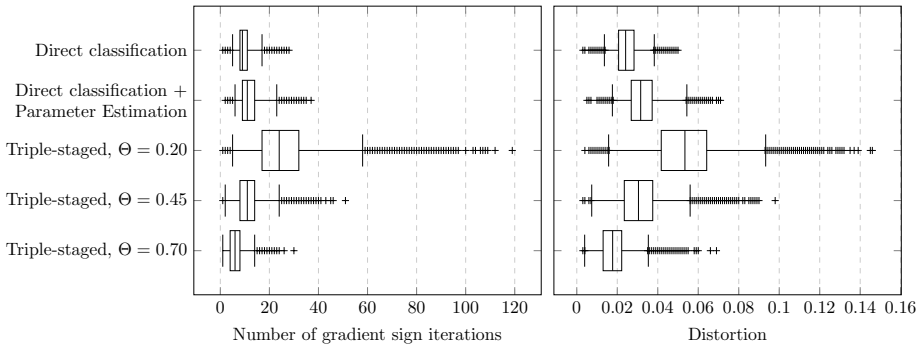


Fig. 6: The effort required to convert the test images that contain one of the 3D models to an image where the network judges that none of the 3D models is visible. This was possible for all the samples in the test set. The instances that were misclassified in the first place were filtered out. This is $0.00\%, 0.01\%, 0.17\%, 0.77\%, 4.12\%$ of the data respectively from top to bottom. The whiskers are placed at the 2^{nd} and 98^{th} percentile.

5 Discussion

We have adapted the classical network structure for classifier tasks and shown significant improvements in robustness against adversarial samples. In order not to pollute the results we have not taken into account other types of solutions against adversarial samples such as adversarial training. This does not however mean these techniques would not be useful also in our setting. As future work we therefore plan to incorporate adversarial training into our approach.

Future work could also apply our technique to include more parameters including internal deformations, using joints etc. We have only tested three 3D models with a limited parameter space. In [15, 21] it was already shown that it is possible to estimate viewpoints of 3D models in real-world images. Dosovitskiy et al. [5] have shown that a deconvolutional neural can generate images based on many classes and viewpoints. This opens up possibilities to expand our work to a real-world situation.

For the present work we opted to train the network in three separate stages. This allowed us quite a bit of control over the network architecture which, in turn, allowed us a shorter route to testing the working hypothesis. Nevertheless, as future work, it would be interesting to develop end-to-end training methods for which the architecture would be more emergent and less explicit. As an obvious first step we could conceive of training the first two stages end-to-end, as an autoencoder, instead of using manually parameterized 3D models.

Acknowledgements

We would like to thank Daan van Beek for the lively discussions during the preconceptual stage of this work, and also for his detailed comments on a draft of this paper.

References

1. Aubry, M., Maturana, D., Efros, A., Russell, B., Sivic, J.: Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In: CVPR (2014), http://www.di.ens.fr/willow/research/seeing3Dchairs/
2. Berntsen, H.: Adversarial background augmentation improves object localisation using convolutional neural networks. Master's thesis, Radboud University (2015), http://www.ru.nl/publish/pages/769526/z_thesis_harm_berntsen.pdf
3. Blender Online Community: Blender - a 3D modelling and rendering package. Blender Foundation, Blender Institute, Amsterdam (2015), http://www.blender.org
4. Chalupka, K., Perona, P., Eberhardt, F.: Visual causal feature learning. In: UAI (2015), http://auai.org/uai2015/proceedings/papers/109.pdf
5. Dosovitskiy, A., Springenberg, J.T., Tatarchenko, M., Brox, T.: Learning to generate chairs, tables and cars with convolutional neural networks. CoRR (2015), http://arxiv.org/abs/1411.5928v3

6. Fraga, P.: Antonov an - 71 - 3d model - .obj, .mb (2015), http://tf3dm.com/3d-model/antonov-an-71-no-texture-77342.html
7. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. CoRR (2015), http://arxiv.org/abs/1412.6572
8. Gu, S., Rigazio, L.: Towards deep neural network architectures robust to adversarial examples. CoRR (2014), http://arxiv.org/abs/1412.5068v4
9. Hinton, G.E., Krizhevsky, A., Wang, S.D.: Transforming auto-encoders. In: ICANN (2011)
10. Huang, R., Xu, B., Schuurmans, D., ri, C.S.: Learning with a strong adversary. CoRR (2015), http://arxiv.org/abs/1511.03034
11. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. CoRR (2014), http://arxiv.org/abs/1408.5093
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
13. Legaz, K.: Tux (armatured) - 3d models - kator legaz (2015), http://web.archive.org/web/20151014015438/http://www.katorlegaz.com/3d_models/miscellaneous/0167/index.php
14. Lyu, C., Huang, K., Liang, H.N.: A unified gradient regularization family for adversarial examples. In: ICDM (2015), http://arxiv.org/abs/1511.06385
15. Massa, F., Russell, B.C., Aubry, M.: Deep exemplar 2d-3d detection by adapting from real to rendered views. CoRR (2015), http://arxiv.org/abs/1512.02497
16. Nguyen, A., Yosinski, J., Clune, J.: Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. IEEE (2015), http://www.evolvingai.org/fooling
17. Norouzi, M., Mikolov, T., Bengio, S., Singer, Y., Shlens, J., Frome, A., Corrado, G., Dean, J.: Zero-shot learning by convex combination of semantic embeddings. In: International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.5650
18. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning deep object detectors from 3d models. In: ICCV (2015), http://www.cv-foundation.org/openaccess/content_iccv_2015/html/Peng_Learning_Deep_Object_ICCV_2015_paper.html
19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115(3), 211–252 (2015), http://arxiv.org/abs/1409.0575
20. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: Workshop at International Conference on Learning Representations (2014), http://arxiv.org/abs/1312.6034
21. Su, H., Qi, C.R., Li, Y., Guibas, L.J.: Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In: The IEEE International Conference on Computer Vision (ICCV) (2015), http://arxiv.org/abs/1505.05641
22. Sun, B., Saenko, K.: From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains. In: Proceedings of the British Machine Vision Confer-

ence. BMVA Press (2014), http://www.bmva.org/bmvc/2014/papers/paper062/index.html
23. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (2014), http://research.google.com/pubs/pub42503.html
24. Tabacof, P., Valle, E.: Exploring the space of adversarial images. CoRR (2015), http://arxiv.org/abs/1510.05328
25. Tang, K.D., Paluri, M., Fei-Fei, L., Fergus, R., Bourdev, L.D.: Improving image classification with location context. In: ICCV (2015), http://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Tang_Improving_Image_Classification_ICCV_2015_paper.pdf
26. Ullman, S., Assif, L., Fetaya, E., Harari, D.: Atoms of recognition in human and computer vision. Proceedings of the National Academy of Sciences (2016), http://www.pnas.org/content/early/2016/02/09/1513198113
27. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015), http://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Zagoruyko_Learning_to_Compare_2015_CVPR_paper.pdf

Handcrafting vs Deep Learning: An Evaluation of NTraj+ Features for Pose Based Action Recognition

Martin Garbade, Juergen Gall

University of Bonn
{garbade,gall}@iai.uni-bonn.de

Abstract. We evaluate the capabilities of the recently introduced NTraj+ features for action recognition based on 2d human pose on a variety of datasets. Inspired by the recent success of neural networks for computer vision tasks like image classification, we also explore their performance on the same action recognition tasks. Therefore we introduce two new neural network architectures which both show competitive performance in comparison to the state-of-the-art. We show that handcrafted features are still useful in the context of action recognition but as the amount of training data keeps on growing the era of neural networks might soon reach the realm of pose based action recognition.

1 Introduction

Action recognition is the task of inferring an action label for a short video clip where a human performs a single action, e.g. clap hands, sit down and shoot bow. Due to the progress of 2d human pose estimation [1], the position of most important body parts like head, hands and feet can be inferred by various techniques. According to the Gestalt principle, the movement of these body parts are enough for the human brain to recognize what action a human is performing. Inspired by this principle we try to infer action labels for short video clips by using only the 2d pose coordinates of the acting person as input to our algorithms.

In [6] the Joint-annotated Human Motion Data Base (Jhmdb) was proposed to study the impact of human pose for action recognition on a challenging dataset consisting of videos taken from the Internet. The authors also proposed a feature descriptor, termed NTraj+, that concatenates many simple descriptors like relative joint positions, distances between joints, angles defined by triplets of joints and their first order temporal derivatives. The features, however, have never been compared with other descriptors. We evaluate NTraj+ on five action datasets (sub-Jhmdb [6], Jhmdb [6], Hdm05 [9], Florence 3D [10], and PennAction [17]) and compare it with the state-of-the-art.

Since NTraj+ are hand-crafted features, we also investigate two neural network architectures that learn pose features in an end-to-end fashion directly from the 2d pose data. The first architecture is based on the AlexNet model [8] which has been proposed for image classification. It comprises a convolution layer that

is applied to the input pose data across seven consecutive time frames. The output is max pooled and followed by three fully connected layers. The second architecture uses a hierarchical body part model and is inspired by an approach for action recognition from 3d pose data [4]. It applies a convolution and max pooling to each of the five body parts separately, i.e. trunk, right arm, left arm, right leg, left leg. Afterward the individual body part layers are successively combined to form a full body layer topped by two fully connected layers.

2 Related Work

Until now current state-of-the-art action recognition baselines for RGB-videos rely on low-level features such as dense trajectories, which is a feature vector encoding the movement of interest points tracked using optical flow and augmented by appearance features such as HOG [14, 15]. CNN architectures used to extract high quality appearance features and trained on optical flow further helped to enhance action recognition performance [11, 3]. Due to stronger and CNN based features, human pose estimation has also made significant advances [12]. Given stronger pose estimates, the paradigm of using low level features for action recognition might soon draw to a close. As Jhuang et al. [6] have shown, high quality pose estimates have the potential of outperforming low-level features on the task of action recognition. The area of pose based action recognition is partly decoupled from pose estimation since pose information can be retrieved in multiple ways (e.g. kinect sensor, motion capturing). Remarkably dissimilar approaches achieve state-of-the-art results on data from RGB-D sensors. Vemulapalli et al. [13] introduce a pose feature representation as points in a Lie group and achieve state-of-the-art results on datasets such as Florence 3D-Action. Du et al. [4] designed a hierarchical recurrent neural network that performs inner product operations on separate body parts which are subsequently fused to a full body model. They use bidirectional recurrent neural networks and LSTM units to combine the frames of each action sequence temporally. Zhang et al. [17] introduce a volumetric, x-y-t, patch classifier to recognize and localize actions.

3 Methodology

We evaluate three different approaches for action recognition. The first approach is the method proposed in [6]. It extracts NTraj+ features and uses a non-linear SVM for classification. The approach is described in Section 3.1. In Section 3.2, we introduce a neural network with fully connected layers and in Section 3.2 we introduce a neural network that models the hierarchical structure of the human body. The neural networks can be applied to pose data directly or to the NTraj+ features. For all neural network computations we used the publically available caffe library [7].

3.1 NTraj+, Bag-of-Words and SVM (BOW)

In the work [6], NTraj+ features have been introduced for action recognition. It combines a variety of descriptors that are extracted from a sequence of 2d human pose. In the original paper, the pose is scale normalized where the scale is given by the annotation tool. In general, the scale is unknown and we therefore do not normalize the pose by scale. The descriptors can be extracted from an arbitrary skeleton. In the following, we describe the features for a skeleton with 15 joints.

1. The first part consists of a 30 dimensional vector containing the x and y positions of the 15 joints relative to the root joint, i.e., the head.
2. The distance between each pair of joints (i, j), i.e., $\|(x_i, y_i) - (x_j, y_j)\|$, yields $\binom{15}{2} = 105$ descriptors.
3. The orientation of each pair of joints (i, j), i.e., $\arctan(\frac{y_i - y_j}{x_i - x_j})$, yields $\binom{15}{2} = 105$ descriptors.
4. For each triplet of joints (i, j, k), an angle is computed for each joint i, j, and k by $\arccos(\mathbf{u}_{ji} \cdot \mathbf{u}_{ki})$, $\arccos(\mathbf{u}_{ij} \cdot \mathbf{u}_{kj})$, and $\arccos(\mathbf{u}_{jk} \cdot \mathbf{u}_{ik})$ with $\mathbf{u}_{ij} = \frac{(x_i, y_i) - (x_j, y_j)}{\|(x_i, y_i) - (x_j, y_j)\|}$. This results in $3 \times \binom{15}{3} = 1,365$ descriptors.

This gives a $1,605$ dimensional feature vector f. In addition first order temporal derivatives are computed over a trajectory of length T, which is subsampled with step size s:

$$(f_{t+s} - f_t, \ldots, f_{t+ks} - f_{t+(k-1)s}) \tag{1}$$

with $k \in [1, \ldots, \lfloor \frac{T}{s} \rfloor]$. This results in $3,210$ feature descriptors. In addition, $(\arctan(\frac{dy_{t+s}}{dx_{t+s}}), \ldots, \arctan(\frac{dy_{t+ks}}{dx_{t+ks}}))$ is added where $dx_{t+ks} = x_{t+ks} - x_{t+(k-1)s}$. This gives additional 15 descriptors summing up to $3,225$ descriptors.

For each descriptor, a codebook is generated by running k-means 10 times on all training samples and choosing the codebook with maximum compactness. These codebooks are used to extract a histogram for each descriptor type and video. For classification, an SVM classifier in a multi-channel setup is used. To this end, for each descriptor type f, a distance matrix D_f is computed that contains the χ^2-distance between the histograms (h_i^f, h_j^f) of all video pairs (v_i, v_j). The kernel matrix for classification is then given by

$$K(v_i, v_j) = \exp\left(-\frac{1}{L} \sum_f \frac{D_f(h_i^f, h_j^f)}{\mu^f}\right) \tag{2}$$

where μ^f is the mean of the distance matrix D_f. For classification, an SVM is trained in a one-vs-all setting.

3.2 Fully Connected Neural Network (FC)

We concatenate the pose of $T = 7$ consecutive frames with a step size of 3 between the frames. Figure 1 a) shows a sketch of the network architecture. The

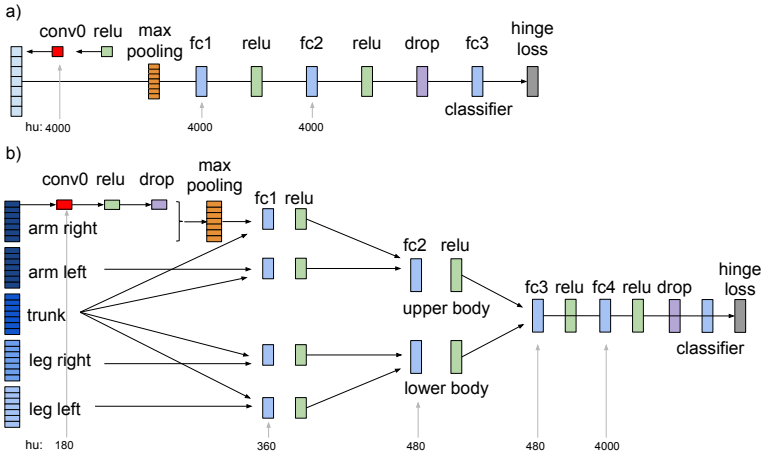


Fig. 1. Architecture of a) the "fully connected" (FC) and b) the hierarchical neural network (HR). Specifications of both neural network architectures: Input is a sequence of either pose coordinates or NTraj+ features computed from 7 consecutive frames of the video sequence with a step size of 3 between the frames. "conv" is a convolution filter applied to every of the 7 input frame separately. Max pooling with kernel size 7 is performed after each convolution. "fc" signifies a fully connected layer. "relu" is a rectified linear unit. "drop" stands for dropout which is set to 50 % chance. "hu" signifies the number of hidden units used in the respective layer.

convolution layer (conv) is applied to all 7 input frames separately. The following max pooling forwards only the values of those frames that have maximum values. The max pooling is followed by three fully connected layers (fc) with a rectified unit (relu) as nonlinearity. The last fully connected layer is the classification layer. As loss layer we use hinge loss with L2 regularization. A dropout layer in front of the classification layer serves for further regularization. All fully connected layers are initialized using the Xavier heuristic [5] and the convolution is initialized with random numbers drawn from a Gaussian distribution.

3.3 Hierarchical Neural Network (HR)

For the hierarchical architecture, we structure the joints by body parts as outlined in Figure 1 b). The convolution is applied to each body part separately followed by a dropout layer. Subsequently the body parts are hierarchically combined while applying a fully connected layer after every combination. In the case of NTraj+ as input feature the features are computed for every body part individually reducing the dimensionality of NTraj+ features substantially. The numbers of hidden units used in both neural network architectures can be found in Figure 1 in the bottom row.

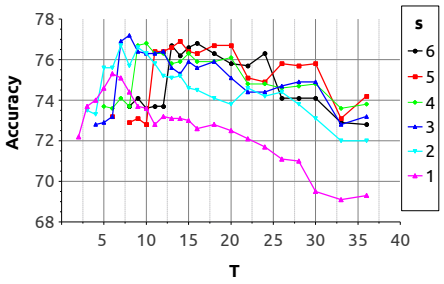


Fig. 2. Impact of the parameters trajectory length (T) and subsampling step size (s) (legend) on the action recognition accuracy for Jhmdb.

	sub-Jhmdb	Jhmdb	Hdm05	Florence 3D	Penn Action
a)	anno-tated	anno-tated	mocap	kinect	anno-tated
b)	12	21	65	9	15
c)	316	928	2337	215	2037
d)	70 / 30	70 / 30	90 / 10	90 / 10	50 / 50
e)	3	3	10	10	1

Table 1. Specifications of the datasets. a) Pose coordinate source. b) Number of action categories. c) Number of action sequences. d) Training / testing ratio. e) Number of splits.

4 Experiments

4.1 Datasets

We perform action recognition on five datasets, namely sub-Jhmdb [6], Jhmdb [6], Hdm05 [9], Florence 3D [10], and PennAction [17]. All datasets are transformed into a uniform skeleton consisting of 13 joint locations + neck and belly. The other thirteen joints are head, shoulders, elbows, wrists, hips, knees and ankles. Although, Hdm05 and Florence3D provide 3d pose, we only use 2d projections of the poses. Penn-Action provides 13 joints which we augment with the locations of neck and belly. The latter are computed as center of mass of shoulders or hips and shoulders, respectively.

Table 1 summarizes the specifications of each dataset. In the case of Hdm05, we follow the protocol proposed in [4] and randomly subsample sequences from the entire dataset. Thus videos of the same actor performing the same action can occur in both the training and the testing set. This makes the results of Hdm05 especially prone to overfitting.

4.2 Evaluation of NTraj+ Parameters

Using the SVM as described in Section 3.1, we perform an evaluation of the two parameters trajectory length T and step size s. In general, the performance has its peak when the trajectory is subdivided once and the differences are computed from start middle and the end frame as can be seen from Figure 2. The best configuration is obtained for $T = 7$ and $s = 3$, which is used for the rest of the experiments. In general, the NTraj+ features are not sensitive to a particular parameter choice.

4.3 Different Feature Combinations

For the neural networks, we evaluate different fusion schemes. For all frames in a video sequence, we extract either a) the feature layer corresponding to the last fully connected layer before the classification layer, denoted by *feats*, or b) the

scores of the classification layer using an additional softmax layer, denoted by *scores*. For version a) we train a linear SVM (one-vs-all) using Lib-SVM [2].

To aggregate all frames belonging to the same sequence, we evaluated three different methods, namely the average, max and min computed for each of the K outputs f of the neural network over all N frames belonging to the same video:

$$a_k = \frac{1}{N} \sum_{n=1}^{N} f_n(k) \tag{3}$$

$$M_k = \max_n f_n(k) \tag{4}$$

$$m_k = \min_n f_n(k) \tag{5}$$

This is then concatenated to obtain one feature vector per video:

$$(a_1, ..., a_K) \tag{6}$$

$$(M_1, ..., M_K) \tag{7}$$

$$(M_1, ..., M_K, m_1, ..., m_K, a_1, ..., a_K). \tag{8}$$

In case of a), it is then used to train the linear SVM.

The same aggregation schemes are applied for b) for each class score $c_n \in [1, 2, ..., C]$. The action label $\hat{c}$ for a sequence is then obtained by

$$\hat{c} = \arg\max_{c=1...C} a(c) \tag{9}$$

$$\text{or} \quad \hat{c} = \arg\max_{c=1...C} M(c). \tag{10}$$

Table 2 and 3 show that generally using the CNN features combined with max aggregation scheme performs best. Only in the case of pose input data the min-max-average aggregation performs slightly better. But since pose generally performs worse than NTraj+ features, we stick to the max aggregation scheme. It is interesting to note that the neural networks perform better with the hand crafted NTraj+ features than with the raw pose data.

4.4 Comparisons

Finally we compare the performance of all three methods on all datasets comparing pose vs. NTraj+ features as input (see Table 4). We see that in every experiment the NTraj+ helps to achieve top performance compared to using pose only. Depending on the dataset, SVM with NTraj+ or FC with NTraj+ performs best. Although most of the results perform slightly less than state-of-the-art performance, we see that all three approaches are quite robust for a variety of datasets and perform competitively when they are used with NTraj+ features. It needs to be noted that P-CNN [3] uses the annotation scale, which is usually not available, and the methods [4] and [13] use 3D pose. Given that we use only 2d pose, the results obtained by the NTraj+ features are impressive. On the Penn-Action dataset, the features outperform the state-of-the-art.

			sub-Jhmdb	Jhmdb	Hdm05	Florence 3D	Penn Action	Total Acc
FC	Max	feats	76.7	69.8	**95.3**	89.8	93	84.9
		scores	73.6	70.8	93.5	86.3	87.6	82.4
	Average	feats	73.6	70.4	95.1	86.4	90	83.1
		scores	**74.7**	70.8	94.3	87.2	89.1	83.2
	Min-Max-Average	feats	74.3	70.7	95.8	86.3	91.9	83.8
HR	Max	feats	73.9	71.8	94.5	**88.3**	**94.1**	**84.5**
		scores	71.3	**71.9**	84.4	87.4	89	80.8
	Average	feats	74.3	71.3	93.3	87	92.4	83.7
		scores	72	72	86.5	88	90.3	81.8
	Min-Max-Average	feats	73.9	71.7	94.6	**88.3**	**94.1**	**84.5**

Table 2. Comparison of different feature combination schemes for NTraj+ computed on individual body parts. The best performance is achieved when a 4000 dimensional feature vector is retrieved from the previous last fully connected layer of a neural network for every frame in a video sequence. The frames are then combined

			sub-Jhmdb	Jhmdb	Hdm05	Florence 3D	Penn Action	Total Acc
FC	Max	feats	71.1	65.6	**90.4**	82.7	**92.9**	80.5
		scores	68.6	66.1	80.8	81.5	88.3	77.1
	Average	feats	69.3	65.7	88.1	83.7	90.9	79.5
		scores	69.4	65.4	83.4	82.3	89.2	77.9
	Min-Max-Average	feats	71.5	66.6	90	82.7	92.8	**80.7**
HR	Max	feats	**71.9**	66.2	85.3	82.9	92.8	79.8
		scores	67.6	**66.9**	66.8	83.5	79.5	72.9
	Average	feats	70.7	65	82.9	**83.9**	90.5	78.6
		scores	66.5	65.3	68.7	83	82.1	73.1
	Min-Max-Average	feats	70.7	65.5	85.2	83	92.8	79.4

Table 3. Comparison of different feature combination schemes for pose input data

		sub-Jhmdb	Jhmdb	Hdm05	Florence 3D	Penn Action	Total Acc
BOW	NTraj+	75.6 ± 2.7	**76.9 ± 4.1**	**95.4 ± 1.1**	88.5 ± 6.3	**98**	**86.9**
FC	NTraj+	**76.7 ± 6**	69.8 ± 2.6	95.3 ± 0.9	**89.8 ± 7.4**	93	84.9
HR	NTraj+ *	73.9 ± 1	71.8 ± 1.2	94.5 ± 1.3	88.3 ± 8.1	94.1	84.5
FC	Pose	71.1 ± 3.4	65.6 ± 1.4	90.4 ± 2.3	82.7 ± 7.3	92.9	80.5
HR	Pose	71.9 ± 4.5	66.2 ± 2.8	85.3 ± 2	82.9 ± 13.7	92.8	79.8
		78.2 [3]	77.8 [3]	96.9 [4]	90.9 [13]	85.5 [16]	

Table 4. Comparison of all three methods: bag-of-words (BOW), fully connected neural network (FC), and hierarchical neural network (HR). The frames for each video sequence are aggregated using the max-feats scheme. (*): For HR, the NTraj+ features are computed for each body part individually

5 Conclusion

We demonstrated that NTraj+ is a robust pose feature descriptor that enhances action recognition performance across a variety of datasets. Further we could show that relatively shallow neural network architectures already achieve performances close to the state-of-the-art suggesting the need for further investigation into that domain.

The work was partially supported by the ERC grant ARCA (677650).

References

1. Andriluka, M., Pishchulin, L., Gehler, P., Schiele, B.: 2d human pose estimation: New benchmark and state of the art analysis. In: CVPR (2014)
2. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology
3. Chéron, G., Laptev, I., Schmid, C.: P-CNN: Pose-based CNN Features for Action Recognition. In: ICCV (2015)
4. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: CVPR (2015)
5. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)
6. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.J.: Towards understanding action recognition. In: ICCV. pp. 3192–3199 (Dec 2013)
7. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. p. 2012
9. Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., Weber, A.: Documentation mocap database Hdm05. Tech. Rep. CG-2007-2, University of Bonn (2007)
10. Seidenari, L., Varano, V., Berretti, S., Del Bimbo, A., Pala, P.: Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In: CVPR, Workshop 12 - Human Activity Understanding from 3D Data . pp. 479–485 (2013)
11. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS (2014)
12. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks pp. 1653–1660 (2014)
13. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: CVPR (2014)
14. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action Recognition by Dense Trajectories. In: CVPR. pp. 3169–3176 (2011)
15. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. In: ICCV 2013. pp. 3551–3558 (2013)
16. Xiaohan Nie, B., Xiong, C., Zhu, S.C.: Joint action recognition and pose estimation from video. In: CVPR (2015)
17. Zhang, W., Zhu, M., Derpanis, K.G.: From actemes to action: A strongly-supervised representation for detailed action understanding. In: ICCV. pp. 2248–2255 (2013)

Quality Prediction for a Road Detection System

J. Kreger[1,3], L. Fischer[2,3], S. Hasler[3], U. Bauer-Wersing[1], T. H. Weisswange[3]

[1] Frankfurt University of Applied Sciences, [2] Bielefeld University,
[3] Honda Research Institute Europe GmbH

Abstract. Machine learning (ML) algorithms are used for autonomous driving systems, but they cannot always offer reliable results. Our approach predicts the credibility of such a system without knowing its internal details. Based on the prediction, poor results can be rejected to raise the overall system quality. We show that our approach outperforms a rejection strategy based on classifier confidences for road detection.

ML is one prominent method for autonomous driving systems [2]. Typically ML methods perform very well on general scenes, but in some rare scenes they show inferior performance or even fail completely. A system relying on ML methods must be able to differentiate situations where its output is safe and scenes where failure is expected. In the latter case, the result should be rejected and the user could be alerted to take back control. A common approach to tackle this issue is to reject a classification if the classifier is unconfident on the result [1]. To design a proper confidence measure, detailed knowledge of the classifier itself is required. Another approach is meta-learning [4], where an external algorithm predicts the performance of the original classifier. In this way no knowledge of the original classifier has to be at hand. We propose a new meta-learning approach based on holistic scene information to reject error-prone scenes. We compare our results to a rejection method based on classifier confidences and to a baseline using ground truth data. We demonstrate our approach exemplary at a road terrain detection system (RTDS, [3]) used for advanced driver assistance systems.

We propose to predict the RTDS performance and only use its results if they are predicted to be reliable (Fig. 1). Otherwise a fallback strategy is triggered. We use a boosting predictor[1] based on global image descriptors typically used for scene recognition tasks [5]. We chose centrist features containing statistical information on edges, and a color histogram to compensate missing color information in the centrist features. The RTDS output is a confidence map from which we compute the output quality by comparing it to ground truth information, and a confidence measure dependent on the number of pixels that do not match a certainty threshold. Note that we use RTDS information during training only.

We use a base data set with 204 color images collected with a vehicle platform in German cities and on rural roads, for which RTDS ground truth is available. We extend this data set by modifying some images with global noise, e. g. γ-correction ($\gamma \in [0.125, 8]$), Gaussian blur ($\sigma \in \{3, 6\}$) and salt noise (probability $p \in \{0.1, 0.2\}$). RTDS fails only on few scenes of the base data, while the extended

[1] Since different parameter settings only had minor effects no details are reported.

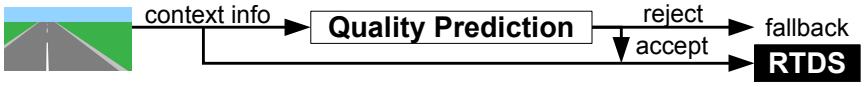


Fig. 1. System overview.

data set is more challenging. Figure 2 shows the test results on the extended set, using leave one out cross validation. The left plot shows the mean system quality change when rejecting the worst sample one by one dependent on ground truth quality, predicted quality, or RTDS confidence. Our approach performs better than the confidence based rejection and we reach the performance of the baseline for small rejection rates. Hence our approach is a reasonable choice to improve the quality of the overall system. The right plot shows the prediction results and an example rejection threshold (marked point at left plot): many predictions are accepted or rejected correctly (green) but many poor RTDS outputs are accepted and some good results are rejected (red).

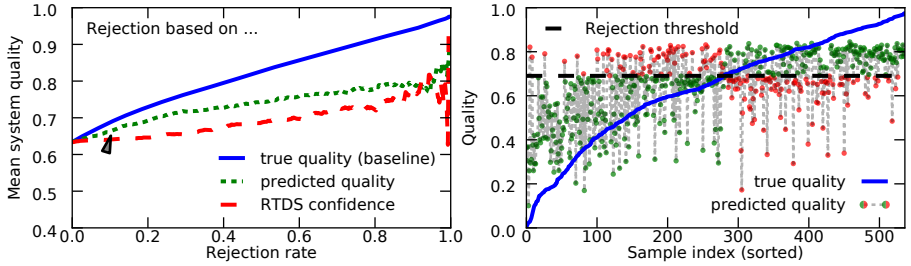


Fig. 2. Results on the extended data. Left: Rejection performance. We omit a rejection rate of one (undefined quality). Right: Single predictions, sorted by true quality.

Our work shows the capacities of rejection based on meta-learning. It turned out that finding discriminative features to predict the RTDS output quality is a challenging task (Fig. 2, right) that needs further analysis. In the future we will (i) incorporate other types of global features and (ii) apply our approach to other application domains in ML and computer vision e. g. image segmentation.

References

1. Fischer, L., Hammer, B., Wersing, H.: Efficient rejection strategies for prototype-based classification. Neurocomputing 169, 334–342 (2015)
2. Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, R.G.: Making Bertha See. In: IEEE International Conference on Computer Vision Workshops (ICCVW 2013). pp. 214–221. IEEE, Sydney, Australia (2013)
3. Fritsch, J., Kühnl, T., Kummert, F.: Monocular Road Terrain Detection by Combining Visual and Spatial Information. IEEE Transactions on Intelligent Transportation Systems 15(4), 1586–1596 (2014)
4. Lemke, C., Budka, M., Gabrys, B.: Metalearning: a survey of trends and technologies. Artificial Intelligence Review 44(1), 117–130 (2015)
5. Wu, J., Rehg, J.M.: CENTRIST: A visual descriptor for scene categorization. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(8), 1489–1501 (2011)

Object Detection Based on Deep Learning and Context Information

Paulin Pekezou Fouopi, Gurucharan Srinivas, Sascha Knake-Langhorst, and Frank Köster

German Aerospace Center, Institute of Transportation Systems, Braunschweig, Germany

{paulin.pekezoufouopi,gurucharan.srinivas,sascha.knake-langhorst,frank.koester}@dlr.de

Abstract. In order to avoid collision with other traffic participants, automated vehicles need to understand the traffic scene. Object detection, as part of scene understanding, remains a challenging task mostly due to the highly variable object appearance. In this work, we propose a combination of convolutional neural networks and context information to improve object detection. To accomplish that, context information and deep learning architectures, which are relevant for object detection, are chosen. Different approaches for integrating context information and convolutional neural networks are discussed. An ensemble system is proposed, trained, and evaluated on real traffic data.

Keywords: Object Detection, Convolutional Neural Networks, Context Information, Bayesian Models

1 Introduction

Automated driving is one of the most important research topics in automotive area. In recent years, many projects like PROMETHEUS, the DARPA Grand/Urban challenge, and CityMobil as well as different research groups and institutions have addressed this topic with promising results. To plan a collision free trajectory, automated driving vehicles must be able to detect objects. Although many solutions are available in the literature, this remains a challenging task due to huge variation in object appearance and scene complexity. Object appearance can change according to occlusion, noise, variation in pose and illumination [1], and background clutter. Convolutional Neural Networks (CNN) show the best classification results, but still have some classification errors because they are mostly appearance-based classifiers. Context information can be used to improve object detection [1]. In this paper we propose an object detection system, which uses the advantages of CNN and context-based classifiers. We discuss different approaches for combining both classifiers. The proposed system is trained and evaluated on real traffic data.

The next sections of this work are divided as follows: in section 2, we present the state of the art. The proposed system as well as training and evaluation results are discussed in section 3. In section 4, we conclude and give an outlook on future work.

2 Related Work

Object detection consists of localizing object instances (hypotheses generation) in an image and classifying those into semantic classes (hypotheses classification). Hypotheses are generated using features like symmetry, aspect ratio, expected position, color, and motion. Hypotheses classification methods can be separated into shape- and feature-based approaches. In this work we focus on the second one.

Feature-based approaches first transform hypotheses into features and classify them. Features can be generated manually or learned directly from the data using e.g. Deep Learning (DL). Manually generated features like Histogram of Oriented Gradients and Deformable Parts Model [2] are used with Shallow Learning (SL) classifiers like Support Vector Machine for vehicle and pedestrian classification. While these SL-classifiers show promising results, they suffer from human errors made during the feature engineering task. DL approaches solve this problem by learning the specific features inherently from large training data set. Since 2012, many DL-classifiers like Faster R-CNN and Yolo outperform SL-classifiers for object detection, but suffer from wrong detections mostly due to the appearance variation drawback depicted above.

In [1, 3] spatial (interposition, support, and position), semantic (co-occurrence), and scale (familiar size) context information between objects, scenes und situations were combined with SL-classifiers to improve object detection. It is difficult to explicitly model the contextual dependencies described above into CNN because CNN just reason about spatial dependencies between object parts. The simplest solution is to integrate context information as pre- or post-processing step. Chu et al. [4] used an ensemble system, which combined the Faster R-CNN, local and global context for object detection. Some efforts to integrate context information directly into the CNN were shown in [5] (time constraint) and [6] (global image-level and local super-pixel context). Liang et al. [7] argued that Recurrent CNN (RCNN) were more suitable for integrating contextual relations, but RCNN can just reason about spatial dependencies between objects and their parts. Contextual dependencies on object and scene levels were still missed and will be address in this work.

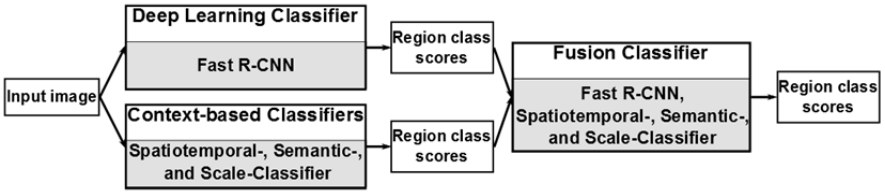


Fig. 1. Overview of the system for integrating DL and context-based classifiers

3 "Our Approach" with Current Results

In this work, we focus on the integration of DL and context-based classifiers using an ensemble system (see **Fig. 1**). We follow the idea proposed in [4], but use different context information and graphical model. As DL-classifier, we choose the pre-trained Fast R-CNN [8]. The semantic (se_{cf}), spatiotemporal (st_{cf}) and scale (sc_{cf}) context proposed in [1] are used for generating context-based features. The context-based classifiers estimate the conditional class probability $p(C|X_{\mathrm{cf}})$ of an object hypothesis given the context-based feature $X_{\mathrm{cf}} \in \{se_{\mathrm{cf}}, st_{\mathrm{cf}}, sc_{\mathrm{cf}}\}$ using the naïve Bayes classifier

$$p(C|X_{\mathrm{cf}}) = \frac{p(X_{\mathrm{cf}}|C)p(C)}{\int_C p(X_{\mathrm{cf}}|C)p(C)}, \tag{1}$$

where $p(X_{\mathrm{cf}}|C)$ is the likelihood function. $C \in \{ped., non\_ped.\}$ is the semantic class set and $p(C)$ the prior class probability. The fusion classifier combines the Fast R-CNN and the context-based classifiers scores S_{f_rcnn} and $S_{cb\_c} = (S_{se\_c}, S_{st\_c}, S_{sc\_c})$ using a Bayesian network and the assumption that the scores are conditionally independent given C. The conditional class probability is

$$p(C|S_{cb\_c}, S_{f\_rcnn}) = \frac{p(S_{cb\_c}|C)p(S_{f\_rcnn}|C)p(C)}{\int_C p(S_{cb\_c}|C)p(S_{f\_rcnn}|C)p(C)}. \tag{2}$$

$p(S_{cb\_c}|C)$, and $p(S_{f\_rcnn}|C)$ are the likelihood functions. S_{se_c}, S_{st_c}, and S_{sc_c} are the semantic, spatiotemporal and scale context-based classifiers scores.
For evaluating the proposed system, we used the Caltech Pedestrian Data Set (CPDS) [9]. Only the aspect ratio $a_r = w/h$ was used as context-based feature, since it belonged to the scale context proposed in [1] and the CPDS didn't contain depth information. h and w were the height and the width of a given bounding box. The likelihood functions $p(X_{\mathrm{cf}}|C)$, $p(S_{cb\_c}|C)$, and $p(S_{f\_rcnn}|C)$ were modeled as Gaussian distributions and the Maximum Likelihood Estimator (MLE) were used to estimate theirs parameters. The prior probability $p(C)$ was the ratio of pedestrians respectively non-pedestrians present in the training dataset. **Fig. 2** presents from left to right the ground truth as well as the Fast R-CNN, the aspect ratio-based classifier (A_R-classifier), and the fusion classifier results with scores greater than 0.5. We observed that the Fast R-CNN detected the most of pedestrians. Just a few objects were missed probably because of the low resolution and occlusion. Although the A_R-classifier had many false positive, the fusion classifier improved the Fast R-CNN and A_R-classifier detecting more pedestrians. The fusion classifier false positive could be explained by the fact that aspect ratio was not powerful enough to model the context.

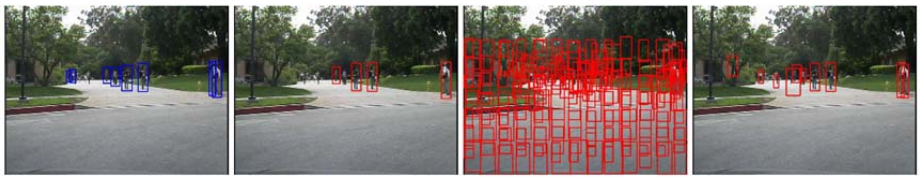


Fig. 2. Detection results on CPDS ([9]). See text for more information.

4 Conclusion and Future Work

In this work, we addressed the problem of integrating context information and DL architectures into a system for object detection. A fusion system combining DL and context-based classifiers was proposed. We modeled the context-based classifiers using the naïve Bayes method. The DL and the context-based classifiers scores were fused using a Bayes model. For training and evaluating our system, we used the DL-classifier called Fast R-CNN. The context-based features were generated using aspect ratio. The Likelihood functions parameters were learned with the MLE on the CPDS dataset. First results on real data revealed that the proposed system was able to improve the detection in some cases, but also had some false positive. Integrating more context information may compensate this effect.
In our future work, we will integrate more context information (e.g. explicitly reasoning about occlusion) and evaluate the system on large data set. The problem of integrating context directly into the DL architecture will be addressed. Another key aspect will be to investigate the possibility of learning context information directly from the data without explicit modelling.

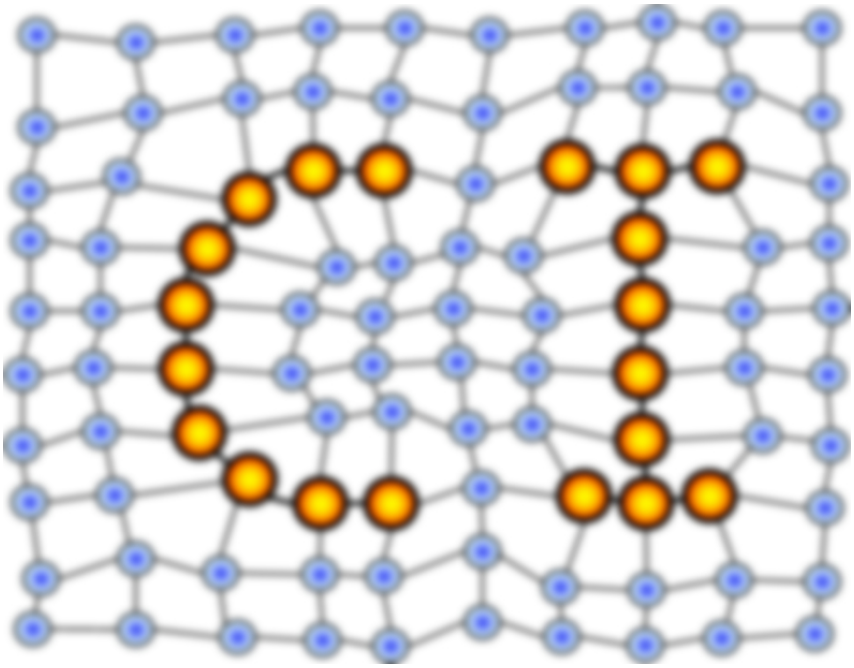
References

References

1. Galleguillos, C., Belongie, S.: Context Based Object Categorization: A Critical Survey. Comput. Vis. Image Underst. 114, 712–722 (2010)
2. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1627–1645 (2010)
3. Biederman, I., Mezzanotte, R.J., Rabinowitz, J.C.: Scene perception. Detecting and judging objects undergoing relational violations. Cognitive Psychology 14, 143–177 (1982)
4. Chu, W., Cai, D.: Deep Feature Based Contextual Model for Object Detection. CoRR abs/1604.04048 (2016)
5. Kang, K., Ouyang, W., Li, H., Wang, X.: Object Detection from Video Tubelets with Convolutional Neural Networks. CoRR abs/1604.04053 (2016)
6. Liang, X., Xu, C., Shen, X., Yang, J., Tang, J., Lin, L., Yan, S.: Human Parsing with Contextualized Convolutional Neural Network. IEEE Transactions on Pattern Analysis and Machine Intelligence PP (2016)
7. Liang, M., Hu, X. (eds.): Recurrent convolutional neural network for object recognition. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
8. Girshick, R.B.: Fast R-CNN. CoRR abs/1504.08083 (2015)
9. Caltech Pedestrian Detection Benchmark, http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

MACHINE LEARNING REPORTS

Report 04/2016



Impressum

Machine Learning Reports ISSN: 1865-3960

▽ Publisher/Editors

Prof. Dr. rer. nat. Thomas Villmann
University of Applied Sciences Mittweida
Technikumplatz 17, 09648 Mittweida, Germany
• http://www.mni.hs-mittweida.de/

Dr. rer. nat. Frank-Michael Schleif
University of Bielefeld
Universitätsstrasse 21-23, 33615 Bielefeld, Germany
• http://www.cit-ec.de/tcs/about

▽ Acknowledgments

We would like to thank the reviewers for their time and patience.