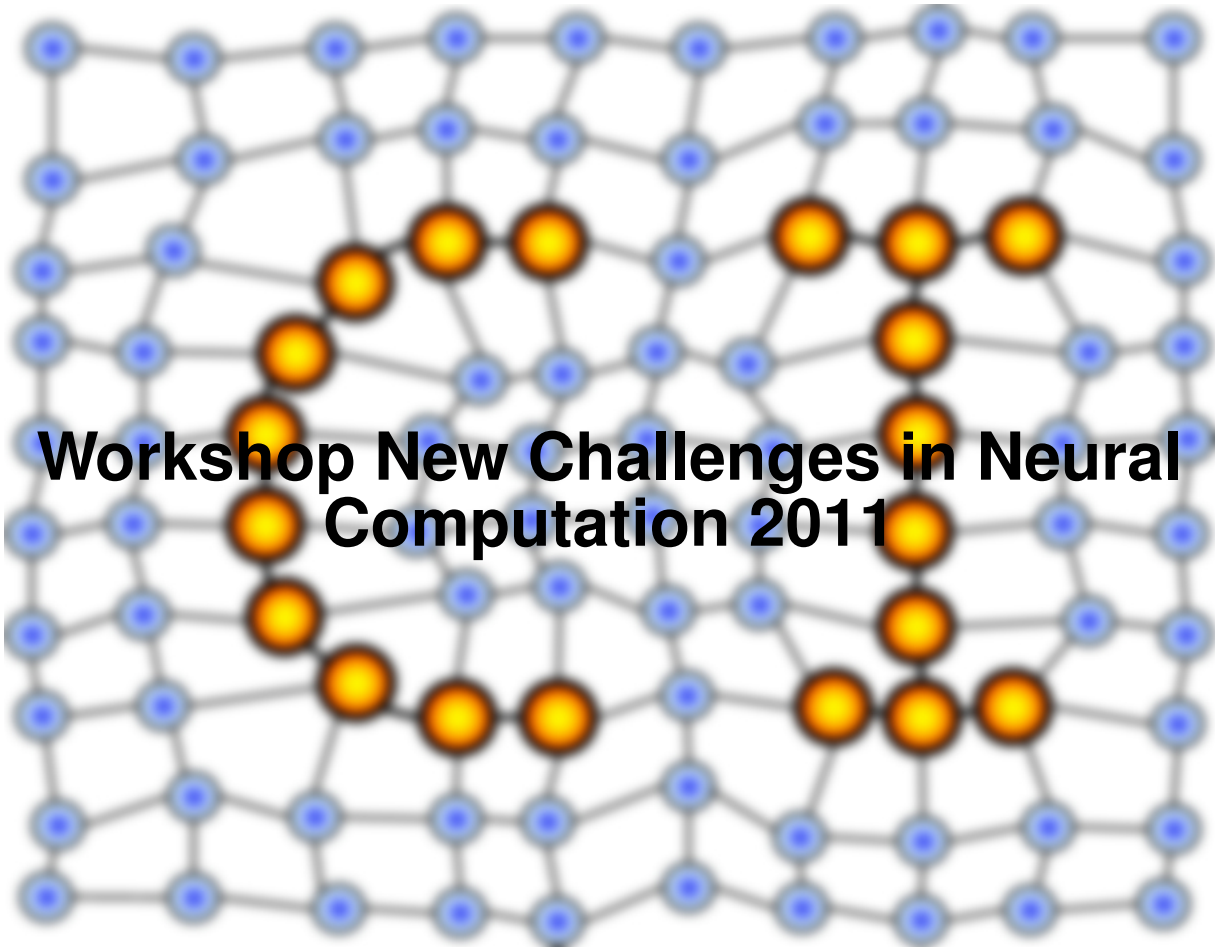
MACHINE LEARNING REPORTS



Workshop New Challenges in Neural Computation 2011

Report 05/2011
Submitted:26.08.2010
Published: 30.08.2011

Barbara Hammer[1] and Thomas Villmann[2] (Eds.)
(1) University of Bielefeld, Dept. of Technology CITEC - AG Computational Intelligence, Universitätsstrasse 21-23, 33615 Bielefeld
(2) University of Applied Sciences Mittweida, Technikumplatz 17, 09648 Mittweida, Germany

Table of contents:

New Challenges in Neural Computation NC^2 – 2011

Barbara Hammer[1] and Thomas Villmann[2]

1 – Cognitive Interaction Technology – Center of Excellence,
Bielefeld University, Germany
2 – 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 second time, this year in connection to the prestigious DAGM conference in Frankfurt am Main. Again, the workshop centers around exemplary challenges and novel developments of neural systems covering recent research concerning theoretical issues as well as practical applications of neural research. This year, among general contributions, a special focus topic was chosen: autonomous learning, which deals with the central problem of how machines can learn as autonomously as humans in unknown environments without the necessity of dedicated focussed tasks or a teacher with shapes the problems such that the machine can solve it easily. We are happy to have two well-known invited speakers in this area: Marc Toussaint, who is also one of the main investigators of a corresponding priority program of the German Research Foundation, presents an overview about recent developments to autonomously learn by means of relational representations in statistical environments. Jochen Steil, managing director of the CoR-Lab Research Institute for Cognition and Robotics, presents novel ways in which robots can learn autonomously inspired by cognitive learning processes. The invitation of invited speakers became possible due to the generous sponsoring of the European Neural Networks Society (ENNS). Correspondingly, the workshop was not only supported by the working group Neural Networks of the German Computer Society but also by the German chapter of the ENNS, the GNNS.

Besides these invited talks, a large number of regular contributions demonstrates the active research in the field of neural networks. Interestingly, all contributions can be linked to complex learning problems beyond simple classical supervised learning, demonstrating the relevance of the special focus topic. A number of contributions centers around the question of how to represent complex signals in a sparse cognitively plausible way: Sven Hellbach et al. present a very general approach how to decompose motion into basic constituents by means of non-negative matrix factorization. Jens Hocke et al. use similar principles to represent image data. The contributions by Xibin Zhu et al. and Marika Kästner and Thomas Villmann deal with a sparse prototype based representation of data, thereby focussing on different complex non-vectorial data formats. A second set of papers centers around the question how learning paradigms beyond simple supervised classification can be canonically formalized, focussing on semi-supervised learning in the contribution by Oliver Beyer and Philipp

Cimiano, perceptual grouping in the approach of Martin Meier et al. and data visualization in the proposal by Wouter Lueks et al. Time plays an essential role in learning processes and, therefore, should be treated explicitly in the frame of autonomous learning. Michael Glodek et al. extend classical hidden Markov models to advanced models which can reliably deal with complex activities. Similarly, the approaches of Alexandra Barchunova et al. and Matthias Rolf and Jochen Steil deal with motion trajectories of the hand, autonomously recognizing and producing, respectively, complex hand trajectories. Being one of our most powerful senses, vision plays a central role in learning processes. The last three contributions of Johannes Lohmann and Martin V. Butz, Hannes Schulz and Sven Behnke, and Marco K. Müller et al. deal with different facets of how to connect this sense to other modes, or to solve complex tasks such as segmentation and recognition with cognitively plausible architectures.

Altogether, these contributions constitute promising steps into the direction of complex autonomous information processing with neural systems by providing new paradigms, concepts, and models.

Challenges of exploration, learning and goal-directed behavior in relational worlds

Keynote talk: Prof. Dr. Marc Toussaint, Machine Learning and Robotics Lab, FU Berlin

Abstract: Natural environments composed of many manipulable objects can be described in terms of probabilistic relational models. Autonomous learning, exploration and planning in such environments is generally hard, but can be tackled when exploiting the inherent relational structure. I will first cover some basic research of our lab in the area of planning by inference before I address in more detail our recent advances in relational exploration, learning and planning, with emphasis on robotics applications. The question of how neurons could do such kind of "inference in relational representations" is rather puzzling to me - but I conjecture that animals and humans in some way or another have to do such kinds of computations.

Neurons Driving Cognitive Robots

Keynote talk: Prof. Dr. Jochen J. Steil, CoR-Lab, Bielefeld University

Abstract: Cognitive Robotics is one major application domain for neural learning methods, whereas robustness to environmental conditions, learning in interaction with human partners, and developmental learning are ideal and challenging playgrounds. We will discuss recent progress using brain-inspired learning and architecture with focus on three important questions: how to get from simple movement to rich motor skills? What do human inspired computational architectures contribute? How shall interaction with human users be shaped? Application examples will include the child-like iCub, the commercial humanoid Nao and the Honda humanoid robot. Finally, we will illustrate that the developed methods are also highly relevant for tomorrow's much more flexible automation technology.

Learning Motion Primitives using Spatio-Temporal NMF

Sven Hellbach[1], Christian Vollmer[1], Julian P. Eggert[2], and Horst-Michael Gross[1]

[1] Ilmenau University of Technology, Neuroinformatics and Cognitive Robotics Labs,
POB 10 05 65, 98684 Ilmenau, Germany
christian.vollmer@tu-ilmenau.de
[2] Honda Research Institute Europe GmbH, Carl-Legien-Strasse 30,
63073 Offenbach/Main, Germany
julian.eggert@honda-ri.de

1 Introduction

The understanding and interpretation of movement trajectories is a crucial component in dynamic visual scenes with multiple moving items. Nevertheless, this problem has been approached very sparsely by the research community. Most approaches for describing motion patterns, like [1], rely on a kinematic model for the observed human motion. This causes the drawback that the approaches are difficult to adapt to other objects. Here, we aim at a generic, model-independent framework for decomposition, classification and prediction.

Consider the simple task for a robot of grasping an object which is handed over by the human interaction partner. To avoid a purely reactive behaviour, which might lead to 'mechanical' movements of the robots, it is necessary to predict the further movement of the human's hand.

In [2] an interesting concept for a decomposition task is presented. Like playing a piano a basis alphabet – the different notes – are superimposed to reconstruct the observation (the piece of music). Regarding only the information, when a base primitive was active, gives rise to an instance of the so called 'piano model' which is a very low-dimensional and sparse representation and which can be exploited for further processing. While the so-called piano model relies on a set of given basis primitives, our approach is able to learn these primitives from the training data.

We use [3], a blind source separation approach in concept similar to PCA and ICA. The system of basis vectors which is generated by the NMF is not orthogonal. This is very useful for motion trajectories, since one basis primitive is allowed to share a common part of its trajectory with other primitives and to specialize later.

2 Non-negative Matrix Factorization

Like other approaches, e. g. PCA and ICA, non-negative matrix factorization (NMF) [3] is meant to solve the source separation problem. Hence, a set of training data is decomposed into basis primitives $\mathbf{W}$ and activations thereof $\mathbf{H}$:

$$\mathbf{V} \approx \mathbf{W} \cdot \mathbf{H} \tag{1}$$

Each training data sample is represented as a column vector $\mathbf{V}_i$ within the matrix $\mathbf{V}$. Each column of the matrix $\mathbf{W}$ stands for one of the basis primitives. In matrix $\mathbf{H}$ the element H_i^j determines how the basis primitive $\mathbf{W}_j$ is activated to reconstruct training sample $\mathbf{V}_i$.

For generating the decomposition, optimization-based methods are used. Hence, an energy function E has to be defined:

$$E(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \left\| \mathbf{V} - \mathbf{T} \cdot \mathbf{W} \cdot \mathbf{H} \right\|^2 + \lambda \sum_{i,j} H_i^j \tag{2}$$

By minimizing the energy equation, it is now possible to achieve a reconstruction using the matrices $\mathbf{W}$ and $\mathbf{H}$. This reconstruction is aimed to be as close as possible to the training data $\mathbf{V}$. In

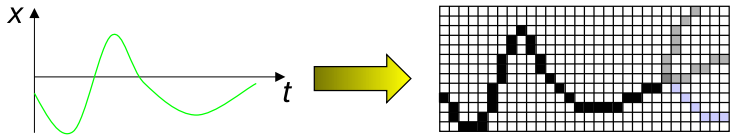


Fig. 1. Motion Trajectories are transferred into a grid representation. A grid cell is set to 1 if it is in the path of the trajectory and set to zero otherwise. Each dimension has to be regarded separately. During the prediction phase multiple hypotheses can be gained by superimposing several basis primitives. This is indicated with the grey trajectories on the right side of the grid.

addition the basis primitives are intended to be allowed to move, rotate and scale freely. This is achieved by adding a transformation matrix $\mathbf{T}$ to the decomposition formulation [4]. For each allowed transformation the corresponding activity has to be trained individually. To avoid trivial or redundant solutions a further sparsity constraint is necessary. Its influence can be controlled using the parameter λ [5].

The minimization of the energy function can be done by gradient descent. The factors H and W are updated alternately with a variant of exponentiated gradient descent until convergence.

3 Decomposing Motion Trajectories

For being able to decompose and to predict the trajectories of the surrounding dynamic objects, it is necessary to identify them and to follow their movements. For simplification, a tracker is assumed, which is able to provide such trajectories in real-time. A possible tracker to be used is presented in [6]. The given trajectory of the motion is now interpreted as a time series $\mathcal{T}$ with values $\boldsymbol{s}_i = (x_i, y_i, z_i)$ for time steps $i = 0, 1, \ldots, n-1$:

$$\mathcal{T} = (\mathbf{s}_0, \mathbf{s}_1, \ldots, \mathbf{s}_{n-1}). \tag{3}$$

It is now possible to present the vector $\mathcal{T}$ directly to the NMF approach. But this could result in an unwanted behaviour, while trying to reconstruct the motion by use of the basis primitives. Imagine two basis primitives, one representing a left turn and another representing a right turn. A superposition of those basis primitives would result in a straight movement.

The goal is to have a set of basis primitives, which can be concatenated one after the other. Furthermore, it is necessary for a prediction task to be able to formulate multiple hypotheses. For achieving these goals, the x-t-trajectory is transferred into a grid representation, as it is shown in figure 1. Then, each grid cell (x_i, t_j) represents a certain state (spatial coordinate) x_i at a certain time t_j. Since most of the state-of-the-art navigation techniques rely on grid maps, the prediction can be integrated easily. Grid Maps were first introduced in [7]. This 2D-grid is now presented as image-like input to the NMF algorithm. Using the grid representation of the trajectory also supports the non-negative character of the basis components and their activities.

It has to be mentioned, that the transformation to the grid representation is done for each of the dimensions individually. Hence, the spatio-temporal NMF has to be processed on each of these grids. Regarding each of the dimensions separately is often used to reduce the complexity of the analysis of trajectories (compare [8]). However, the algorithm's only limitations to handle multi-dimensional grid representation is the increase of computational effort.

While applying an algorithm for basis decomposition to motion trajectories it seems to be clear that the motion primitives can undergo certain transformations to be combined to the whole trajectory. For example, the same basis primitive standing for a straight move can be concatenated with another one standing for a left turn. Hence, the turning left primitive has to be moved to the end of the straight line, and transformation invariance is needed while decomposing motion data. For our purposes, we concentrate on translation. This makes it possible to reduce the complexity of the calculations and to achieve real time performance.

The sparse coding constraint helps to avoid trivial solutions. Since the input can be compared with a binary image, one possible solution would be a basis component with only a single grid cell filled. These can then be concatenated one directly after another. So, the trajectory would simply be copied into the activities.

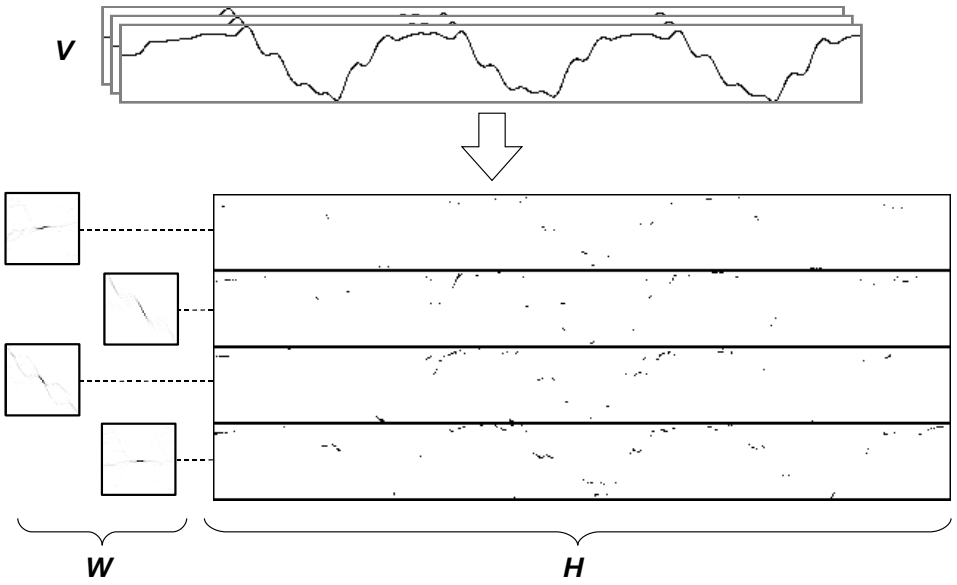


Fig. 2. Training with Spatio-Temporal NMF. Given is a set of training samples in matrix **V**. The described algorithm computes the weights **W** and the corresponding activities **H**. Only the weights are used as basis primitives for further processing.

Training Phase: The goal of the training phase is to gain a set of basis primitives which allow to decompose an observed and yet unknown trajectory (see Fig. 2). As it is discussed in section 3, the training samples are transferred into a grid representation. These grid representations are taken as input for the NMF approach and are therefore represented in matrix **V**. On this matrix **V** the standard NMF approach, extended by the sparsity constraint and by translation invariance, is applied. The algorithm is summarized in [9].

Beside the computed basis primitives, the NMF algorithm also provides the information of how each of the training samples can be decomposed by these basis primitives.

Application Phase: As it is indicated in Fig. 3, from the training phase a set of motion primitives is extracted. During the application phase, we assume that the motion of a dynamic object (e. g. a person) is tracked continuously. For getting the input for the NMF algorithm, a sliding window approach is taken. A certain frame in time is transferred into the already discussed grid like representation. For this grid the activation of the basis primitives is determined by trying to reconstruct the input.

The standard approach to NMF implies that each new observation at the next time step demands a new random initialization for the optimization problem. Since an increasing column number in the grid representation stands for an increase in time, the trajectory is shifted to the left while moving further in time. For identical initialization, the same shift is then reflected in the activities after the next convergence. To reduce the number of iterations until convergence, the shifted activities from the previous time step are used as initialization for the current one.

To fulfil the main goal discussed in this paper – the prediction of the observed trajectory into the future – the proposed algorithm had to be extended. Since the algorithm contains the transformation invariance constraint, the computed basis primitives can be translated to an arbitrary position on the grid. This means that they can also be moved in a way that they exceed the borders of the grid. Up to now, the size of reconstruction was chosen to be the same size as the input grid. Hence, using the standard approach means that the overlapping information has to be clipped. To be able to solve the prediction task, we simply extend the reconstruction grid to the right – or into the future (see Fig. 3). So, the previously clipped information is available for prediction.

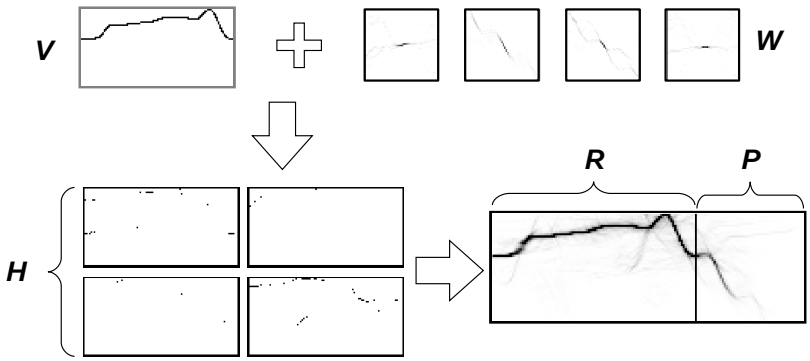


Fig. 3. The basis primitives **W**, which were computed during the training, are used to reconstruct (matrix **R**) the observed trajectory **V**. This results in a set of sparse activities – one for each basis primitive – which describe on which position in space and time a certain primitive is used. Beside the reconstruction of the observed trajectory (shown in Fig. 3), it is furthermore possible to predict a number of time steps into the future. Hence, the matrix **R** is extended by the prediction horizon **P**.

References

1. Hoffman, H., Schaal, S.: A computational model of human trajectory planning based on convergent flow fields. In: 37st Meeting of the Society of Neuroscience. (2007)
2. Cemgil, A., Kappen, B., Barber, D.: A generative model for music transcription. IEEE Transactions on Speech and Audio Processing **14** (2006) 679–694
3. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing **13** (2001) 556–562
4. Eggert, J., Wersing, H., Körner, E.: Transformation-invariant represenatation and NMF. In: IJCNN. (2004) 2535 – 2539
5. Eggert, J., Körner, E.: Sparse Coding and NMF. In: IJCNN. (2004) 2529 – 2533
6. Otero, N., Knoop, S., Nehaniv, C., Syrdal, D., Dautenhahn, K., Dillmann, R.: Distribution and Recognition of Gestures in Human-Robot Interaction. ROMAN (2006) 103–110
7. Elfes, A.: Using Occupancy Grids for Mobile Robot Perception and Navigation. Computer **12**(6) (June 1989) 46–57
8. Naftel, A., Khalid, S.: Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space. MM Syst. **12**(3) (2006) 227–238
9. Hellbach, S., Eggert, J., Koerner, E., Gross, H.M.: Basis decomposition of motion trajectories using spatio-temporal nmf. In: ICANN. (2009) 804–814

Image Deconvolution with Sparse Priors

Jens Hocke[1], Thomas Martinetz[1], and Erhardt Barth[1]

[1]Institute for Neuro- and Bioinformatics, University of Lübeck

August 16, 2011

Abstract

Optical systems used for image acquisition are usually not perfect and leading to degraded images. A typical degradation is image blur. Building perfect optics is not always possible due to physical limitations, cost, size or weight. Therefore, there is interest in computational solutions to remove these degradations. By knowing the sources of distortion it is possible to remove them.

Image blur can be removed by deconvolution, however, the problem which has to be solved is underdetermined. For solving these ill-posed problems additional assumptions have to be considered. Recently, many advances were made in the investigation of underdetermined systems of equations [1] in cases where the solution can be sparsely encoded. The sparseness constraint is used to select a plausible solution out of an infinite set of possible solutions. This method is applied to the deconvolution problem.

Similar to other approaches to deconvolution based on sparse coding, for speed and memory efficiency we apply the fast Fourier transform and the fast wavelet transform to model the convolution and provide a sparse basis [2]. For the convolution, boundary areas are cut to avoid wrong modelling due to the cyclic nature of the Fourier transform. By cutting the boundary areas the system of equations becomes underdetermined.

We apply this approach to a pinhole camera setting. Using a simulated pinhole camera, we look at the influence of sparseness and the robustness to noise. First tests have also been made using a real pinhole camera.

References

[1] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[2] M. A. T. Figueiredo and R. D. Nowak, "A bound optimization approach to wavelet-based image deconvolution," in *IEEE International Conference on Image Processing – ICIP'2005*, (Genoa, Italy), pp. 782–785, 2005.

Learning vector quantization for classification of dissimilarity data

Xibin Zhu, Frank-Michael Schleif, Barbara Hammer
{xzhu,fschleif,bhammer}@techfak.uni-bielefeld.de
Bielefeld University, CITEC-Centre of Excellence
D-33594 Bielefeld, Germany

August 17, 2011

Abstract

Prototype models offer an intuitive interface to given data volumes because they represent the model in terms of typical prototypes which can directly be inspected by the user. Popular supervised prototype methods such as learning vector quantization suffer from its restriction to Euclidean vectors. Thus, they are not suited to deal with general dissimilarity data which occurs more and more often in applications. In this contribution two extensions of supervised prototype based methods to deal with general dissimilarity data are proposed.

1 Introduction

Machine learning techniques have revolutionized the possibility to deal with large electronic data sets. Prominent methods like the support vector machine provide highly accurate models, but they often constitute black box mechanisms such that their decision process can hardly be inspected by humans. In contrast, prototype-based methods represent decisions in terms of typical representatives, which can easily be inspected by humans in the same way as data points.

Different methods have been proposed to infer prototypes from given data, such as variants of k-means and topographic mapping and statistical counterparts [3, 2, 5]. One of the most popular supervised prototype based method is given by learning vector quantization (LVQ). Modifications and extensions thereof relate the model to explicit cost functions or statistical models [2, 8, 4], with excellent generalization ability [6, 7].

In modern applications, data are often addressed using non-Euclidean dissimilarities such as dynamic time warping for time series or alignment for symbolic strings. In such cases, a Euclidean representation of data is not possible. Rather, data are given implicitly in terms of pairwise dissimilarities or relations. Standard LVQ and its variants cannot be used in such cases.

In this contribution, we provide relational extensions of generalized LVQ (GLVQ) and robust soft LVQ (RSLVQ) [8, 4] such that supervised prototype based classification for dissimilarity data becomes possible. Thereby, we refer to techniques which have recently been established in unsupervised settings [9, 10]: prototypes are represented implicitly as linear combinations of data in the so-called pseudo-Euclidean embedding. The relevant distances of data and prototypes can be computed without an explicit reference to the vectorial data representation. This principle holds for every symmetric dissimilarity matrix and thus, allows us to formalize a valid objective of RSLVQ and GLVQ for relational data.

In this contribution, we review GLVQ and RSLVQ first, which are subsequently extended to relational data and tested on several benchmarks.

2 Prototype based clustering and classification

Assume data $\vec{x}^i \in \mathbb{R}^n, i = 1, \ldots, m$, are given. Prototypes are contained in the same space $\vec{w}^j \in \mathbb{R}^n, j = 1, \ldots, k$. They decompose data into receptive fields $R(\vec{w}^j) = \{\vec{x}^i \ : \ \forall k \ d(\vec{x}^i, \vec{w}^j) \leq d(\vec{x}^i, \vec{w}^k)\}$ based on a dissimilarity measure, e.g. the Euclidean distance $d(\vec{x}^i, \vec{w}^j) = \|\vec{x}^i - \vec{w}^j\|^2$.

In supervised settings, $\vec{x}^i$ and $\vec{w}^j$ are equipped with class labels $c(\vec{x}^i) \in \{1, \ldots, L\}$ and $c(\vec{w}^j)$, respectively. A data point is assigned to the class of its closest prototype. The classification error is given by $\sum_j \sum_{\vec{x}^i \in R(\vec{w}^j)} \delta(c(\vec{x}^i) \neq c(\vec{w}^j))$ with the standard delta function δ. Since this cannot efficiently be optimized directly, LVQ and its extensions rely on a heuristic or alternative cost function which relates to the classification error [2]. The cost function of Generalized LVQ (GLVQ) [8] is given by

$$E_{\mathrm{GLVQ}} = \sum_i \Phi \left(\frac{d(\vec{x}^i, \vec{w}^+(\vec{x}^i)) - d(\vec{x}^i, \vec{w}^-(\vec{x}^i))}{d(\vec{x}^i, \vec{w}^+(\vec{x}^i)) + d(\vec{x}^i, \vec{w}^-(\vec{x}^i))} \right)$$

where Φ is a differentiable monotonic function such as the hyperbolic tangent, and $\vec{w}^{\pm}(\vec{x}^i)$ refers to the closest, equally $(+)$ or differently $(-)$ labeled prototype to $\vec{x}^i$. The error of a point $\vec{x}^i$ is smallest if $d(\vec{x}^i, \vec{w}^+) < d(\vec{x}^i, \vec{w}^-)$, leading to a correct classification. The cost function emphasizes the hypothesis margin of the classifier by summing over the differences of the distances. Usually, the cost function is optimized by a stochastic gradient descent with random initialization of the prototypes. Given a data point $\vec{x}^i$, the udate of $\vec{w}^{\pm}$ is given by:

$$\Delta \vec{w}^{\pm}(\vec{x}^i) \quad \sim \quad \mp \Phi'(\mu(\vec{x}^i)) \cdot \mu^{\pm}(\vec{x}^i) \cdot \nabla_{\vec{w}^{\pm}(\vec{x}^i)} d(\vec{x}^i, \vec{w}^{\pm}(\vec{x}^i))$$

where

$$\mu(\vec{x}^i) = \frac{d(\vec{x}^i, \vec{w}^+(\vec{x}^i)) - d(\vec{x}^i, \vec{w}^-(\vec{x}^i))}{d(\vec{x}^i, \vec{w}^+(\vec{x}^i)) + d(\vec{x}^i, \vec{w}^-(\vec{x}^i))}, \ \mu^{\pm}(\vec{x}^i) = \frac{2 \cdot d(\vec{x}^i, \vec{w}^{\mp}(\vec{x}^i))}{(d(\vec{x}^i, \vec{w}^+(\vec{x}^i)) + d(\vec{x}^i, \vec{w}^-(\vec{x}^i)))^2}.$$

For the squared Euclidean norm, we get $\nabla_{\vec{w}^j} d(\vec{x}^i, \vec{w}^j) = -2(\vec{x}^i - \vec{w}^j)$.

Robust soft LVQ (RSLVQ) [4] is an alternative statistical approach, which in the limit of small bandwidth leads to updates similar to LVQ. For non-vanishing bandwidth, soft assignments of data points to prototypes take place. Each $\vec{w}^j$ and induces a Gaussians $p(\vec{x}^i|\vec{w}^j) = K \cdot \exp(-d(\vec{x}^i, \vec{w}^j)/2\sigma^2)$ with variance $\sigma \in \mathbb{R}$ and normalization constant $K = (2\pi\sigma^2)^{-n/2}$. Assuming equal prior for each $\vec{w}^j$, we obtain the overall and class dependent probability, respectively, of $\vec{x}^i$ by

$$p(\vec{x}^i) = \sum_{\vec{w}^j} p(\vec{x}^i|\vec{w}^j)/K, \qquad p(\vec{x}^i, c(\vec{x}^i)) = \sum_{\vec{w}^j : c(\vec{w}^j)=c(\vec{x}^i)} p(\vec{x}^i|\vec{w}^j)/K\,.$$

The cost function of RSLVQ is induced by the quotient of these probabilities

$$E_{\mathrm{RSLVQ}} = \log \prod_i \frac{p(\vec{x}^i, c(\vec{x}^i))}{p(\vec{x}^i)} = \sum_i \log \frac{p(\vec{x}^i, c(\vec{x}^i))}{p(\vec{x}^i)}$$

E is optimized by means of a stochastic gradient descent, i.e. for given $\vec{x}^i$:

$$\Delta\vec{w}^j \sim -\frac{1}{2\sigma^2} \cdot \left(\frac{p(\vec{x}^i|\vec{w}^j)}{\sum_{j:c(\vec{w}^j)=c(\vec{x}^i)} p(\vec{x}^i|\vec{w}^j)} - \frac{p(\vec{x}^i|\vec{w}^j)}{\sum_j p(\vec{x}^i|\vec{w}^j)} \right) \cdot \nabla_{\vec{w}^j} d(\vec{x}^i, \vec{w}^j)$$

if $c(\vec{x}^i) = c(\vec{w}^j)$ and

$$\Delta\vec{w}^j \sim \frac{1}{2\sigma^2} \cdot \frac{p(\vec{x}^i|\vec{w}^j)}{\sum_j p(\vec{x}^i|\vec{w}^j)} \cdot \nabla_{\vec{w}^j} d(\vec{x}^i, \vec{w}^j)$$

if $c(\vec{x}^i) \neq c(\vec{w}^j)$. In the limit of small bandwidth, the soft assignments become crisp values, leading to the standard LVQ update in case of mistakes of the classifier.

3 Dissimilarity data

In typical applications, often, data are described by means of a dedicated dissimilarity measures to account for the complexity of the data. Standard supervised prototype techniques are restricted to Euclidean vector spaces. Recently unsupervised prototype methods have been extended to more general formats [9]. Following this approach, we extend GLVQ and RSLVQ to relational variants.

We assume that data $\vec{x}^i$ are characterized by pairwise symmetric dissimilarities $d_{ij} = d(\vec{x}^i, \vec{x}^j)$, with $d_{ii} = 0$. D refers to the corresponding dissimilarity matrix[1]. We do not require that d refers to a Euclidean data space, i.e. D does not need to be Euclidean embeddable, nor does it need to fulfill the conditions of a metric.

As argued in [10, 9], every such data can be embedded in a pseudo-Euclidean vector space the dimensionality of which is limited by the number of points.

[1] It is easy to transfer similarities to dissimilarities and vice versa, see [10].

The pseudo-Euclidean vector space is a real-vector space with the bilinear form $\langle \vec{x}, \vec{y} \rangle_{p,q} = \vec{x}^t I_{p,q} \vec{y}$ where $I_{p,q}$ is a diagonal matrix with p entries 1 and q entries -1. The tuple (p, q) is the signature of the space; q determines how far the standard Euclidean norm has to be corrected by negative eigenvalues to arrive at the given dissimilarity measure. The data are Euclidean if and only if $q = 0$. For a given D, its pseudo-Euclidean embedding can be computed by means of an eigenvalue decomposition of the related Gram matrix. It yields explicit vectors $\vec{x}^i$ such that $d_{ij} = \langle \vec{x}^i - \vec{x}^j, \vec{x}^i - \vec{x}^j \rangle_{p,q}$ holds for every pair of data points.

Based on this observation, we embed prototypes in this pseudo-Euclidean vector space. We restrict prototypes to linear combination of data points of the form

$$\vec{w}^j = \sum_i \alpha_{ji} \vec{x}^i \text{ with } \sum_i \alpha_{ji} = 1 \,.$$

In this case, dissimilarities can be computed implicitly by means of the formula

$$d(\vec{x}^i, \vec{w}^j) = [D \cdot \alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j$$

where $\alpha_j = (\alpha_{j1}, \ldots, \alpha_{jn})$ is a vector of coefficients describing $\vec{w}^j$ implicitly [9]. Based on this observation, we can transfer the Euclidean cost function to the relational case.

The costs of relational GLVQ (RGLVQ) are:

$$E_{\text{RGLVQ}} = \sum_i \Phi \left(\frac{[D\alpha^+]_i - \frac{1}{2} \cdot (\alpha^+)^t D \alpha^+ - [D\alpha^-]_i + \frac{1}{2} \cdot (\alpha^-)^t D \alpha^-}{[D\alpha^+]_i - \frac{1}{2} \cdot (\alpha^+)^t D \alpha^+ + [D\alpha^-]_i - \frac{1}{2} \cdot (\alpha^-)^t D \alpha^-} \right) ,$$

where the closest correct and wrong prototype coefficients are referred to using α^+ and $\alpha-$, respectively. Adaptation of the coefficients $\alpha^\pm$ in RGLVQ is given by:

$$\Delta \alpha_k^\pm \quad \sim \quad \mp \Phi'(\mu(\vec{x}^i)) \cdot \mu^\pm(\vec{x}^i) \cdot \frac{\partial \left([D\alpha^\pm]_i - \frac{1}{2} \cdot (\alpha^\pm)^t D \alpha^\pm \right)}{\partial \alpha_k^\pm}$$

where $\mu(\vec{x}^i)$, $\mu^+(\vec{x}^i)$, and $\mu^-(\vec{x}^i)$ are as above. The partial derivative yields

$$\frac{\partial \left([D\alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j \right)}{\partial \alpha_{jk}} = d_{ik} - \sum_l d_{lk} \alpha_{jl}$$

Similarly, the costs of RSLVQ can be extended

$$E_{\text{RRSLVQ}} = \sum_i \log \frac{\sum_{\alpha_j : c(\alpha_j) = c(\vec{x}^i)} p(\vec{x}^i | \alpha_j)/k}{\sum_{\alpha_j} p(\vec{x}^i | \alpha_j)/k}$$

where

$$p(\vec{x}^i | \alpha_j) = \frac{\exp\left(- \left([D\alpha_j]_i - \frac{1}{2} \cdot \alpha_j^t D \alpha_j \right) / 2\sigma^2 \right)}{K}$$

Stochastic gradient descent leads to the updates

$$\Delta\alpha_{jk} \sim -\frac{1}{2\sigma^2} \cdot \left(\frac{p(\vec{x}^i|\alpha_j)}{\sum_{j:c(\alpha_j)=c(\vec{x}^i)} p(\vec{x}^i|\alpha_j)} - \frac{p(\vec{x}^i|\alpha_j)}{\sum_j p(\vec{x}^i|\alpha_j)} \right) \cdot \frac{\partial \left([D\alpha_j]_i - \frac{1}{2}\alpha_j^t D\alpha_j \right)}{\partial \alpha_{jk}}$$

if $c(\vec{x}^i) = c(\alpha_j)$ and

$$\Delta\vec{w}^j \sim \frac{1}{2\sigma^2} \cdot \frac{p(\vec{x}^i|\alpha_j)}{\sum_j p(\vec{x}^i|\alpha_j)} \cdot \frac{\partial \left([D\alpha_j]_i - \frac{1}{2}\alpha_j^t D\alpha_j \right)}{\partial \alpha_{jk}}$$

if $c(\vec{x}^i) \neq c(\alpha_j)$.

For both, RGLVQ and RRSLVQ, each adaptation is followed by a normalization: $\sum_i \alpha_{ji} = 1$. The prototypes are initialized randomly with small values for α_{ij} with $\sum_i \alpha_{ji} = 1$. It is possible to take class information into account by setting all α_{ij} to zero which do not correspond to the class of the prototype.

An out of sample extension of the classification to novel data points is immediate based on an observation made in [9]: given a novel data point $\vec{x}$ characterized by its pairwise dissimilarities $D(\vec{x})$ to the data used for training, the dissimilarity of $\vec{x}$ to a prototype represented by α_j is $d(\vec{x}, \vec{w}^j) = D(\vec{x})^t \cdot \alpha_j - \frac{1}{2} \cdot \alpha_j^t D\alpha_j$.

4 Experiments

We evaluate the algorithms for several benchmark data sets where data are characterized by pairwise dissimilarities. We consider eight data sets used also in [1]: Amazon47, Aural-Sonar, Face Recognition, Patrol, Protein and Voting. Further we consider the Cat Cortex from [13], the Copenhagen Chromosomes data [11] and one own data set, the Vibrio data. The last one consists of 1,100 samples of vibrio bacteria populations characterized by mass spectra. The spectra contain approx. 42,000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The preprocessing of the Vibrio data is described in [12] and the underlying similarity measures in [14, 12].

Since some of these matrices correspond to similarities rather than dissimilarities, we use standard preprocessing as presented in [10]. For every data set, a number of prototypes which mirrors the number of classes was used, representing every class by one or two prototypes, see Tab. 1. Initialization of LVQ is done randomly; training takes place for 100 epochs with learning rate 0.1. The parameter σ is optimized on the training set. The evaluation of the results is done by means of the classification accuracy as evaluated on the test set in a ten fold repeated cross-validation with ten repeats The results are reported in Tab. 1. In addition, we report the best results obtained by SVM after diverse preprocessing techniques as reported in the article [1].

Interestingly, in most cases, results which are comparable to the best SVM as reported in [1] can be found by relational GLVQ, while relational RSLVQ leads to a slightly worse accuracy. Note that GLVQ is used directly for the respective dissimilarity matrix, while SVM requires preprocessing to guarantee positive definiteness, see [1].

	#pt	L	**RGLVQ**	**RRSLVQ**	**SVM [1]**	$\lvert\{\vec{w}^j\}\rvert$
Amazon47	204	47	0.810(0.014)	0.830(0.016)	0.82	94
Aural Sonar	100	2	0.884(0.016)	0.609(0.048)	0.87	10
Face Rec.	945	139	0.964(0.002)		0.96	139
Patrol	241	8	0.841(0.014)	0.850(0.011)	0.88	24
Protein	213	4	0.924(0.019)	0.530(0.011)	0.97	20
Voting	435	2	0.946(0.005)	0.621(0.010)	0.95	20
Cat Cortex	65	5	0.930(0.010)	0.910(0.022)	n.d.	12
Vibrio	4200	22	1.000(0.000)	0.941(0.077)	n.d.	49
Chromosome	1100	49	0.927(0.002)		n.d.	63

Table 1: Mean results of prototype based classification in comparison to SVM, the standard deviation is given in parenthesis.

5 Conclusions

We have presented an extension of prototype-based techniques to general possibly non-Euclidean data sets by means of an implicit embedding in pseudo-Euclidean data space and a corresponding extension of the cost function of GLVQ and RSLVQ to this setting. As a result, a very powerful learning algorithm can be derived which, in most cases, achieves results which are comparable to the SVM with the respective best preprocessing technique. Unlike the latter, relational LVQ does not require preprocessing of the data since relational LVQ can directly deal with possibly non-Euclidean data whereas SVM requires a positive semidefinite Gram matrix. Similar to SVM, relational LVQ has quadratic complexity due to its dependency on the full dissimilarity matrix. A speed-up to linear techniques e.g. by means of the Nyström approximation for dissimilarity data similar to [15] is the subject of ongoing research.[2]

References

[1] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, Luca Cazzanti; Similarity-based Classification: Concepts and Algorithms, Journal of Machine Learning Research 10(Mar):747–776, 200

[2] T. Kohonen, editor. *Self-Organizing Maps*. Springer-Verlag New York, Inc., 3rd edition, 2001.

[3] Martinetz, T.M., Berkovich, S.G., Schulten, K.J.: 'Neural-gas' network for vector quantization and its application to time-series prediction. IEEE Trans. on Neural Networks 4(4), 558–569 (1993)

[4] Sambu Seo and Klaus Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003.

[2]Acknowledgement: Financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative and from the "German Science Foundation (DFG)" under grant number HA-2719/4-1 is gratefully acknowledged.

[5] C. Bishop, M. Svensen, and C. Williams. The generative topographic mapping. *Neural Computation* 10(1):215-234, 1998.

[6] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[7] P. Schneider, M. Biehl, and B. Hammer, "Adaptive relevance matrices in learning vector quantization," *Neural Computation*, vol. 21, no. 12, pp. 3532–3561, 2009.

[8] A. Sato and K. Yamada. Generalized learning vector quantization. In M. C. Mozer D. S. Touretzky and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference*, pages 423–9, Cambridge, MA, USA, 1996. MIT Press.

[9] B. Hammer and A. Hasenfuss. Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9):2229-2284, 2010.

[10] E. Pekalska and R.P.W. Duin The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore, December 2005.

[11] C. Lundsteen, J-Phillip, and E. Granum, "Quantitative analysis of 6985 digitized trypsin g-banded human metaphase chromosomes," *Clinical Genetics*, vol. 18, pp. 355–370, 1980.

[12] T. Maier, S. Klebel, U. Renner, and M. Kostrzewa, "Fast and reliable maldi-tof ms–based microorganism identification," *Nature Methods*, no. 3, 2006

[13] B. Haasdonk and C. Bahlmann (2004), Learning with distance substitution kernels, in *Pattern Recognition - Proc. of the 26th DAGM Symposium.*

[14] S. B. Barbuddhe, T. Maier, G. Schwarz, M. Kostrzewa, H. Hof, E. Domann, T. Chakraborty, and T. Hain, "Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry," *Applied and Environmental Microbiology*, vol. 74, no. 17, pp. 5402–5407, 2008.

[15] Andrej Gisbrecht, Barbara Hammer, Frank-Michael Schleif and Xibin Zhu, Accelerating dissimilarity clustering for biomedical data analysis. Proceedings of SSCI 2011

Fuzzy Supervised Neural Gas with Sparsity Constraint

M. Kästner and T. Villmann
University of Applied Sciences Mittweida,
Technikumplatz 17, 09648 Mittweida, Germany
{kaestner,villmann}@hs-mittweida.de

Abstract

In this paper we propose a new approach to combine unsupervised and supervised vector quantization for clustering and fuzzy classification using the neural gas. For this purpose the original cost function is modified in such a way that both aspects, unsupervised vector quantization and supervised classification, are incorporated. Additionally, for improved focussing of fuzzy classification a sparsity constraint is introduced. The convergence of the new algorithm is proven by an adequate redefinition of the underlying dissimilarity measure now interpreted as a dissimilarity in the data space combined with the class label space. Finally, a gradient descent learning as known for the original algorithms is obtained togther with an additional sparsity emphasizing term. Thus a semi-supervised learning scheme is achieved, which is examplary applied for classification of different coffee types.

1 Introduction

Unsupervised and supervised vector quantization by neural maps is still an important issue. Neural maps are prototype based algorithms inspired by biological neural systems. Prominent models are the self-organizing map (SOM) and the neural gas network (NG) [6],[8]. These approaches are designed for unsupervised data clustering (NG) and visualization (SOM). Supervised learning vector quantization follows the idea of prototype based classification preserving the concept of data typical representation in contrast to support vector machines, which emphasize the class borders to describe data classes. Well known such models are the family of learning vector quantizers (LVQ) based on a heuristic adaptation scheme [6], or their cost function based counterpart named generalized LVQ (GLVQ) [11].

There exist only a few approaches to combine unsupervised and supervised learning in SOM or NG. The most intuitive one is simple post labeling after unsupervised training. More advanced is the counterpropagation network where a SOM is subsequently combined with a multilayer perceptron (MLP) [2]. However, the follow-up supervised learning of the MLP does not influence the SOM, and, hence prototypes might be suboptimal. An approach based on a modification of the cost function of NG and SOM (in the Heskes variant, [3]) are the Fuzzy Labeled NG (FLNG) and the Fuzzy Labeled SOM (FLSOM) [15, 17]. Both approaches add an extra term to the standard cost function judging the classification accuracy of the prototypes, which are equipped with a class label to be adapted during the learning together with the prototype positions. Yet, their theoretical justifications are tricky and partially unsatisfactory. In the LASSO algorithm fuzzy labels are concatenated to the data and prototype vectors [9]. These new vectors are treated as usual in NG and SOM during learning. However, in the recall phase the data vectors are presented without label information and classification is performed by association. Therefore, the winner determination is different from those during the learning phase and, hence, also unsatisfactory.

In this paper we propose a much more simple ansatz: We incorporate the classification error in the standard cost functions of a neural vector quantizer by a multiplicative factor. Thereby, this factor evaluates the classification accuracy based on a quasi metric [10]. This allows a

redefinition of the data metric in such a way that the problem can be handled in this new quasi-metric space analogously to the original quantizer equipped with the Euclidean metric. Thus the structural framework of standard vector quantizer is preserved and their convergence properties are transferred to the new model. The new approach can be seen as a kind of semi-supervised learning. Moreover, the model can be applied to both crisp and fuzzy labeled data.

The methodology is applied to classify coffee type based on spectral measurements. The results will be presented during the workshop.

2 The Fuzzy Supervised Neural Gas Model

The usual neural gas model assumes data points $\mathbf{v} \in V \subset \mathbb{R}^n$ with the data density $P(\mathbf{v})$ and prototypes $\mathbf{w}_j \in \mathbb{R}^n$, $j = 1 \ldots N$. The cost function to be minimized in NG is

$$E_{\mathrm{NG}} = \sum_j \int P(\mathbf{v})\, h_\sigma^{NG}(k_j(\mathbf{v}, \mathbf{w}_j))\, d(\mathbf{v}, \mathbf{w}_j)\, d\mathbf{v} \tag{1}$$

with a differentiable (in the second argument) dissimilarity measure $d(\mathbf{v}, \mathbf{w}_j)$ usually taken as the Euclidean distance [8]. The function

$$h_\sigma^{NG}(k_j(\mathbf{v}, \mathbf{w}_j)) = \exp\left(-\frac{k_j(\mathbf{v}, \mathbf{w}_j)}{2\sigma^2}\right) \tag{2}$$

is the neighborhood function depending on the winner rank $k_j(\mathbf{v}, \mathbf{w}_j) = \sum_{i=1}^{N} \Theta(d(\mathbf{v}, \mathbf{w}_j) - d(\mathbf{v}, \mathbf{w}_i))$ where

$$\Theta(x) = \begin{cases} 0 & if \quad x \leq 0 \\ 1 & else \end{cases} \tag{3}$$

is the Heaviside function. We remark that $h_\sigma^{NG}(k_j(\mathbf{v}, \mathbf{w}_j))$ is evaluated in the data space V. An input vector $\mathbf{v}$ is mapped onto a prototype s by the winner-take-all mapping rule

$$s(\mathbf{v}) = \mathrm{argmin}_j\, (d(\mathbf{v}, \mathbf{w}_j)) \tag{4}$$

and the learning takes place as stochastic gradient descent $\frac{\partial_S E_{\mathrm{NG}}}{\partial \mathbf{w}_j}$ on E_{NG}.

In the following we develop a new variant of standard NG, which integrates additional class information into the standard model ending up with a (semi-)supervised variant of standard NG.

We suppose C data classes. Each data vector $\mathbf{v}$ is accompanied by a data assignment vector $\mathbf{c_v} = \left(c_{\mathbf{v}}^1, \ldots, c_{\mathbf{v}}^C\right) \in [0,1]^C$ with vector entries taken as class probabilities or possibility assignments. Crisp classification is obtained by the additional requirement of $c_{\mathbf{v}}^j \in \{0, 1\}$. Analogously, we also equip the prototypes $\mathbf{w}_j$ with class labels $\mathbf{y}_j$. For the FSNG model, we now consider the cost function

$$E_{\mathrm{FSNG}} = \sum_j \int P(\mathbf{v})\, h_\sigma^{NG}(k_j(\mathbf{v}, \mathbf{w}_j))\, D_\varepsilon(\mathbf{v}, \mathbf{w}_j, \gamma)\, d\mathbf{v} \tag{5}$$

which is structurally similar to standard NG but with the new deviation measure

$$D_\varepsilon(\mathbf{v}, \mathbf{w}_j, \gamma) = D_\varepsilon^\delta(\mathbf{v}, \mathbf{w}_j, \gamma) \cdot D_\varepsilon^d(\mathbf{v}, \mathbf{w}_j, \gamma) - \varepsilon_\delta \varepsilon_d \tag{6}$$

with

$$D_\varepsilon^\delta(\mathbf{v}, \mathbf{w}_j, \gamma) = (\gamma \cdot \delta(\mathbf{c_v}, \mathbf{y}_j) + \varepsilon_\delta) \text{ and } D_\varepsilon^d(\mathbf{v}, \mathbf{w}_j, \gamma) = ((1-\gamma) \cdot d(\mathbf{v}, \mathbf{w}_j) + \varepsilon_d) \tag{7}$$

and the offset$\varepsilon_\delta \varepsilon_d$ obtained from the parameter vector $\varepsilon = (\varepsilon_\delta, \varepsilon_d)$. This offset is necessary in D_ε to prevent unexpected behavior of the FSNG under certain conditions[5]. The dissimilarity

measure $D_\varepsilon(\mathbf{v}, \mathbf{w}_j, \gamma)$ takes into account both, the usual dissimilarity $d(\mathbf{v}, \mathbf{w}_j)$ between data and prototypes and a dissimilarity measure $\delta(\mathbf{c_v}, \mathbf{y}_j)$ for the class assignment vectors. The dissimilarity $D_\varepsilon(\mathbf{v}, \mathbf{w}_j, \gamma)$ has to be used during learning in the winner determination (4) whereas in the recall phase the data distance $d(\mathbf{v}, \mathbf{w}_j)$ as to be applied realizing an association model [9]. The parameter $\gamma \in [0, 1]$ determines the influence of the class information with $\gamma = 0$ yielding the standard NG. In the simplest case, both measures, $d(\mathbf{v}, \mathbf{w}_j)$ and $\delta(\mathbf{c_v}, \mathbf{y}_j)$, could be chosen as the (quadratic) Euclidean distance.

The FSNG model leads to a prototype adaptation influenced by the class agreement $\delta(\mathbf{c_v}, \mathbf{y}_j)$:

$$\triangle \mathbf{w}_j = -(1-\gamma) \cdot D_\varepsilon^\delta(\mathbf{v}, \mathbf{w}_j, \gamma) \cdot h_\sigma^{NG}(k_j(\mathbf{v}, \mathbf{w}_j)) \cdot \frac{\partial d(\mathbf{v}, \mathbf{w}_j)}{\partial \mathbf{w}_j} \tag{8}$$

accompanied by the label adaptation

$$\triangle \mathbf{y}_j = -\gamma \cdot D_\varepsilon^d(\mathbf{v}, \mathbf{w}_j, \gamma) \cdot h_\sigma^{NG}(k_j(\mathbf{v}, \mathbf{w}_j)) \cdot \frac{\partial \delta(\mathbf{c_v}, \mathbf{y}_j)}{\partial \mathbf{y}_j} \tag{9}$$

such that both, prototype and the respective class assignment vectors are updated at the same time according to a stocastic gradient on E_{FSNG} [5].

It should be mentioned that $D_\varepsilon(\mathbf{v}, \mathbf{w}_i, \gamma)$ is only a quasi-metric [10]. A detailed consideration can be found in [5]. This completes the proof for the desired FSNG dynamic.

The quasi-metric $D_\varepsilon(\mathbf{v}, \mathbf{w}_i, \gamma)$ depends on the balancing parameter γ weighting the unsupervised and supervised aspects. Experiences from earlier models (Fuzzy Label Neural Gas – FLNG, [15]) suggest a careful control of this parameter beginning with $\gamma(0) = 0$ and later (adiabatic) increase up to a final value $\gamma_{\max}$, which should be chosen as $\gamma_{\max} < 1$ to avoid instabilities as known from FLNG. This can be interpreted as a remaining influence of unsupervised learning in the supervised learning phase of FSNG.

3 Sparsity in Fuzzy Labeling

To focus the fuzzy labeling to only a few classes for each prototype additional sparsity constraints can be added to the FSNG model. A common choice is an entropic penalty. Thus, we assume probabilistic fuzzy labels $\mathbf{y}_i$, i.e. the l_1-norm is $||\mathbf{y}_i||_1 = 1$. We suggest the penalty as

$$P = \kappa(t) \sum_i H(\mathbf{y}_i) \tag{10}$$

with $H(\mathbf{y}_i)$ being the Shannon entropy $H(\mathbf{y}_i) = -\sum_j y_i^j \cdot \log\left(y_i^j\right)$ or the Rényi entropy $H(\mathbf{y}_i) = \frac{1}{1-\alpha} \log\left(\sum_j \left(y_i^j\right)^\alpha\right)$, which leads to an additional update term $\triangle_P y_i^j = -\frac{\partial P}{\partial y_i^j}$. The function $\kappa(t) \geqq 0$ is monotonically increasing after convergence of the usual FSNG model. The sparsity learning is stopped, if a significant loss of accuracy is observed.

4 Conclusion

We provide in this paper a new approach for semi-supervised learning in neural gas. The new approach combines in a dissimilarity measure both the dissimilarity between data and prototypes as well as their class dissimilarity in a multiplicative manner, the balancing of which is controlled by the balancing parameter γ. We show for that the mathematical structure of the underlying cost function is equivalent compared to the original NG, if an adequate redefinition of the dissimilarity measure takes place. In consequence, the theoretical framework of the original NG algorithm justifies the new approach. The theoretical assumptions of stochastic gradient descent learning

for an analog SOM modification are lost, however, such that it is only a heuristic scheme in that case (FSSOM).

Obviously, the new approach allows a broad variability of dissimilarity measures d in the data space and δ for the fuzzy labels. Surely, the Euclidean distance is a good choice. However, interesting alternatives are under discussion for different data types at least for the data dissimilarity measures. Prominent examples are the scaled Euclidean metric for relevance learning [1] and their functional counterpart [4], or the Sobolev distance [16] and other functional norms [7], if the data are supposed to be representations of functions. Generalization of the scaled Euclidean metric are quadratic forms used in matrix learning [13]. Divergences are proposed for spectral data as suitable data dissimilarity measures [14], whereas the utilization of differentiable kernel also seems to be a new promising alternative for data dissimilarity judgment [12].

References

[1] B. Hammer and T. Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 2002.

[2] R. Hecht-Nielsen. Counterprogagation networks. *Appl. Opt.*, 26(23):4979–4984, December 1987.

[3] T. Heskes. Energy functions for self-organizing maps. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 303–316. Elsevier, Amsterdam, 1999.

[4] M. Kästner and T. Villmann. Functional relevance learning in generalized learning vector quantization. *Machine Learning Reports*, 5(MLR-01-2011):81–89, 2011. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_01_2011.pdf.

[5] M. Kästner and T. Villmann. Fuzzy supervised neural gas for semi-supervised vector quantization – theoretical aspects. *Machine Learning Reports*, 5(MLR-02-2011):1–12, 2011. ISSN:1865-3960, http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_02_2011.pdf.

[6] T. Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer, Berlin, Heidelberg, 1995. (Second Extended Edition 1997).

[7] J. Lee and M. Verleysen. Generalization of the l_p norm for time series and its application to self-organizing maps. In M. Cottrell, editor, *Proc. of Workshop on Self-Organizing Maps (WSOM) 2005*, pages 733–740, Paris, Sorbonne, 2005.

[8] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten. 'Neural-gas' network for vector quantization and its application to time-series prediction. *IEEE Trans. on Neural Networks*, 4(4):558–569, 1993.

[9] S. Midenet and A. Grumbach. Learning associations by self-organizatiom: the LASSO model. *Neurocomputing*, 6:343–361, 1994.

[10] E. Pekalska and R. Duin. *The Dissimilarity Representation for Pattern Recognition: Foundations and Applications*. World Scientific, 2006.

[11] 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.

[12] F.-M. Schleif, T. Villmann, B. Hammer, P.Schneider, and M. Biehl. Generalized derivative based kernelized learning vector quantization. In C. Fyfe, P. Tino, D. Charles, C. Garcia-Osorio, and H. Yin, editors, *Proceedings of the Conference IDEAL*, volume 6283 of *LNCS*, pages 21–28, Berlin, 2010. Springer.

[13] P. Schneider, B. Hammer, and M. Biehl. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.

[14] T. Villmann and S. Haase. Divergence based vector quantization. *Neural Computation*, 23(5):1343–1392, 2011.

[15] T. Villmann, B. Hammer, F.-M. Schleif, T. Geweniger, and W. Herrmann. Fuzzy classification by fuzzy labeled neural gas. *Neural Networks*, 19:772–779, 2006.

[16] T. Villmann and F.-M. Schleif. Functional vector quantization by neural maps. In J. Chanussot, editor, *Proceedings of First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2009)*, pages 1–4. IEEE Press, 2009. ISBN 978-1-4244-4948-4.

[17] T. Villmann, F.-M. Schleif, E. Merényi, and B. Hammer. Fuzzy labeled self-organizing maps for classification of spectra. In F. Sandoval, A. Prieto, J. Cabestany, and M. Grana, editors, *Computational and Ambient Intelligence – Proceedings of the 9th Work-conference on Artificial Neural Networks (IWANN), San Sebastian (Spain)*, LNCS 4507, pages 556–563. Springer, Berlin, 2007.

Online semi-supervised growing neural gas

Oliver Beyer and Philipp Cimiano

Semantic Computing Group, CITEC, Bielefeld University,
obeyer@cit-ec.uni-bielefeld.de
http://www.sc.cit-ec.uni-bielefeld.de

In this paper we introduce online semi-supervised growing neural gas (OSSGNG), a novel online semi-supervised learning (SSL) approach for growing neural gas (GNG). Semi-supervised learning exploits both labelled and unlabelled data and has been successfully applied to many clustering and classification tasks. Existing semi-supervised approaches for GNG process the labelled and unlabelled training data in two separate phases in order to perform a classification. They are offline in the sense that each neuron of the network gets labelled after the GNG training ended and therefore it is necessary to store the complete training data. We present an approach that is able to simultaneously process labelled and unlabelled examples of the training data, using online labelling and prediction strategies. Both labelled and unlabelled examples are processed during the learning process of GNG without the need to store any of the training examples explicitly. As main contribution we show that our online approach does perform as good as previous semi-supervised learning extensions of growing neural gas.

In particular, we offer the following contributions:

1. We extend the original GNG algorithm by an on-the-fly labelling step and an on-the-fly prediction step, in order to provide the online processing of labelled and unlabelled data.
2. We compare OSSGNG with SSGNG [1] as baseline on a classification task and show that the online extension of GNG does not deteriorate the classification performance compared to SSGNG, but even outperforms SSGNG in 75% of our experiments.
3. We show that OSSGNG is competitive with respect to other semi-supervised classification approaches [2] .

In order to extend growing neural gas to a semi-supervised classifier, we add two steps (step 4 and 5) to the original GNG algorithm, as shown in Figure 1. In the first step (1), the algorithm starts with two neurons, randomly placed in the feature space. (2) The first stimulus $x \in R^n$ of the input space (first training example) is presented to the network. (3) The two neurons s_1 and s_2 which minimize the euclidean distance towards x are determined as first and second winner. In step (4), a label for x is predicted according to a label prediction strategy,

[1] Zaki, S.M. & Yin, H (2008) A Semi-Supervised Learning Algorithm for Growing Neural Gas in Face Recognition. *Journal of Mathematical Modelling and Algorithms*, 7(4):425-435

[2] Chapelle, O. & Schölkopf, B. & Zien, A. (2006) Semi-Supervised Learning, *MITPress*

in case that x belongs to the unlabelled examples. The prediction strategy used is called *single-linkage* prediction [3]. According to this prediction strategy a new datapoint d_{new} is labelled with category c of the neuron n that minimises the distance to this new example:

$$l(d_{new}) = \arg\min_{c}(\arg\min_{n \in N(c)} |n - d_{new}|^2)$$

where $N(c) = \{n \in N \mid l(n) = c\}$ is the set of all neurons labelled with category c according to the used labelling strategy. In step (5), the label of the presented stimulus is assigned to the winner neuron in each iteration of GNG. The label assignment is performed by an online labelling function, which will be described in the following. We denote the winner neuron for a datapoint d by $w(d)$. The labelling strategy itself is local in the sense that it does not consider any neighbouring neurons besides the winner neuron $w(d)$. The labelling is performed during the training process, which means that the label assigned to a neuron can change over time. Thus, the online labelling function is dependent on the number of examples the network has seen and has the following form: $l : N \times T \rightarrow C$. We simply write $l_t(n_i)$ to denote the label assigned to neuron n_i after having seen t datapoints. We use the *relabelling method* as online labelling strategy, as it has been shown that this strategy has a good performance in classification tasks [3]. According to this very simple strategy, the winner neuron $w(d)$ corresponding to d adopts the label of d:

$$l_t(n_i) = l_t(d), \ where \ n_i = w(d)$$

(6) The age of all edges that connect s_1 to other neurons is increased by 1. In step (7), the local error variable w_{s_1} of s_1 is updated. This error variable will be used later in order to set the location for a newly inserted node. In step (8), s_1 and its topological neighbours will be adapted towards x by fractions e_b (for s_1) and e_n. (9) A new connection between s_1 and s_2 is created and the age of the edge is set to 0. (10) All edges with an age greater than a_{max}, as well as all neurons without any connecting edge, are removed. (11) Depending on the iteration and the parameter λ, a new node r is inserted into the network. It will be inserted half-way between the neuron q with the highest local error and its topological neighbour f having the largest error among all neighbours of q. In addition, the connection between q and f is removed and both neurons are connected to r. In step (12), the error variables of all nodes are decreased by a factor β. (13) The algorithm stops, if the stop criterion is met, which is the size of the network in our case.

Table 1 shows the classification results of our algorithm compared to SSGNG [1] and six standard semi-supervised learning benchmark data sets proposed by Chapelle et al. [2]. The results show, that OSSGNG outperforms SSGNG in 75% of all experiments. They further show, that OSSGNG is competitive with respect to state-of-the-art SSL approaches.

[3] Beyer, O. & Cimiano P. (2011) Online labelling strategies for growing neural gas. *In Press: Proceedings of the 12th International Conference on Intelligent Data Engeneering and Automated Learning*

Online semi-supervised learning for growing neural gas (OSSGNG)

1. Start with two units i and j at random positions in the input space.
2. Present an input vector $x \in R^n$ from the input set or according to input distribution.
3. Find the nearest unit s_1 and the second nearest unit s_2.
4. **If the label of x is missing, assign a label to x according to the present prediction strategy.**
5. **Assign the label of x to s_1 according to the present labelling strategy.**
6. Increment the age of all edges emanating from s_1.
7. Update the local error variable by adding the squared distance between w_{s_1} and x.
8. Move s_1 and all its topological neighbours (i.e. all the nodes connected to s_1 $\Delta error(s_1) = |w_{s_1} - x|^2$ by an edge) towards x by fractions of e_b and e_n of the distance:

$$\Delta w_{s_1} = e_b(x - w_{s_1})$$

$$\Delta w_n = e_n(x - w_n)$$

   for all direct neighbours of s_1.
9. If s_1 and s_2 are connected by an edge, set the age of the edge to 0 (refresh). If there is no such edge, create one.
10. Remove edges with their age larger than a_{max}. If this results in nodes having no emanating edges, remove them as well.
11. If the number of input vectors presented or generated so far is an integer or multiple of a parameter λ, insert a new node r as follows:
    Determine unit q with the largest error.
    Among the neighbours of q, find node f with the largest error.
    Insert a new node r halfway between q and f as follows:

$$w_r = \frac{w_q + w_f}{2}$$

    Create edges between r and q, and r and f. Remove the edge between q and f.
    Decrease the error variable of q and f by multiplying them with a constant α. Set the error r with the new error variable of q.
12. Decrease all error variables of all nodes i by a factor β.
13. If the stopping criterion is not met, go back to step (2). (For our experiments, the stopping criterion has been set to be the maximum network size.)

Fig. 1. GNG algorithm with extension for online semi-supervised learning

	TSVM	Cluster-Kernel	Data-Dep. Reg.	LDS	SSGNG	OSSGNG
g241c/10	75.29	51.72	58.75	71.15	41.53	**58.09**
g241c/100	81.54	86.51	79.69	81.96	60.37	**61.85**
g241d/10	49.92	57.95	54.11	49.37	**63.75**	51.16
g241d/100	77.58	95.05	67.18	76.26	63.36	**64.49**
Digit1/10	82.23	81.27	87.51	84.37	**91.84**	87.23
Digit1/100	93.49	96.21	97.56	96.54	96.86	**97.04**
USPS/10	74.80	80.59	82.04	82.43	92.47	**93.99**
USPS/100	90.23	90.68	94.90	95.04	**95.23**	93.93
COIL/10	32.50	32.68	36.35	38.10	71.06	**76.35**
COIL/100	74.20	78.01	88.54	86.28	87.52	**89.61**
BCI/10	50.85	51.69	49.79	50.73	**55.00**	51.38
BCI/100	66.75	64.83	52.53	56.03	69.37	**70.43**
Average/10	60.93	59.32	61.43	62.69	69.28	**69.70**
Average/100	80.63	85.23	80.07	82.02	78.79	**79.56**

Table 1. Classification accuracy of a 12-fold cross-validation for the different SSL algorithms performed on the 6 datasets (g241c, g24d, Digit1, USPS, COIL, BCI), trained with 10 and 100 examples of labelled training data (best SSGNG vs. OSSGNG results are marked in each line).

Hallucinating Image Features to Supplement Perceptual Groups

Martin Meier, Robert Haschke, Helge Ritter
{mmeier,rhaschke,helge}@techfak.uni-bielefeld.de

August 18, 2011

Abstract

In this paper we present an approach towards cognitive reasonable figure amendments utilizing the Gestalt-based dynamics of the Competitive Layer Model.

1 Introduction

When a human perceives incomplete shapes, for example the ones from Fig. 1, no effort is needed to recognize the meant geometric primitives, although they are far from being complete. In this paper, we propose an human-like approach to fill these "gaps". Based on Gestalt Theory (e.g. see [1] for an overview), especially the law of continuity, we strive to amend these sparse informations through modelling missing parts utilizing the neural dynamics of the Competitive Layer Model (CLM).

The CLM [3] has been proven feasible in a wide spectrum of recognition tasks. Previous works successfully applied the CLM to simulate various grouping tasks based on Gestalt Laws like contour grouping in noisy settings [5] or action segmentation [2].

Based on the approaches for contour grouping, we make use of the internal binding dynamics of the CLM to evaluate the quality of hallucinated features with respect to previously grouped contours.

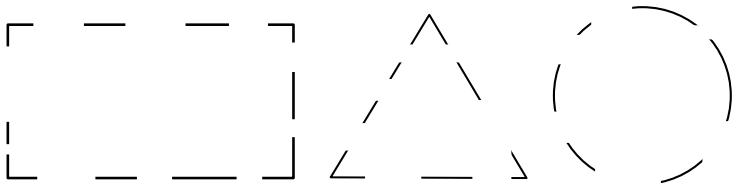


Figure 1: Gestalt Law of continuity: Although the shapes are not complete, they are easily recognized as a rectangle, triangle and circle.

2 The Competitive Layer Model

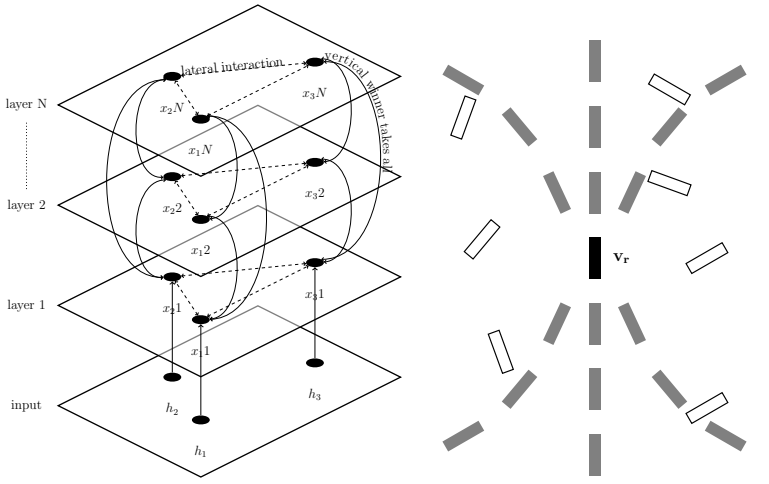
The CLM uses an internal recurrent dynamics to group similar features. To this end, a set of $L \times N$ linear threshold units are arranged in L neuron layers. We denote the activity of a neuron with $x_{r\alpha}$, where $r = 1..N$ denotes the feature index and $\alpha = 1..L$ the layer index. Hence, for each feature r exists a column of neurons across all L layers. The significance of a feature r is determined by the external input h_r (cf. Fig. 2(a)).

Within each layer a lateral interaction $f_{rr'}$ is defined according to the compatibility or similarity of features v_r and $v_{r'}$. If both features are considered similar, a positive connection weight between $x_{r\alpha}$ and $x_{r'\alpha}$ is used, realizing a positive feedback loop. This compatibility measurement is domain specific for the type of used features v and must therefore be explicitly specified in a symmetric interaction function:

$$f_{rr'} = \mathbf{f}(v_r, v_{r'}) = \mathbf{f}(v_{r'}, v_r) \tag{1}$$

This mutually reinforces activity of neurons representing similar features. All layers employ the same lateral interaction weights.

Grouping of features is realized by collecting positive neuronal activity within layers. To enforce activation of a neuron related to a particular feature v_r within a single layer only, the lateral layer-wise interaction is augmented by a column-wise *winner-takes-all (WTA)* interaction. The combination of the vertical WTA



(a) Competitive Layer Model

(b) Compatibility for oriented edges

Figure 2: (a) The Competitive Layer Model with three inputs $h_{1...3}$ and the corresponding neurons $x_{r\alpha}$ in each layer. (b) Compatibility for oriented edges. Emanating from the centered feature v_r, dark filled edges indicate a high compatibility whereas unfilled edges indicate low compatibility.

dynamics and the lateral interactions leads to a linear threshold dynamics of

$$\dot{x}_{r\alpha} = -x_{r\alpha} + \sigma(J(h_r - \sum_{\beta} x_{r\beta}) + \sum_{r'} f_{rr'} x_{r'\alpha}) \tag{2}$$

with $\sigma(x) = max(0, x)$, where $h_r - \sum_\beta x_{r\beta}$ represents the vertical *WTA* interaction, weighted by a (usually small) constant J and $\sum_{r'} f_{rr'} x_{r'\alpha}$ represents the lateral interaction.

Since the lateral interactions $f_{rr'}$ are identical in each layer, they can be calculated once and stored in a symmetric interaction matrix

$$M_{rr'} = \mathbf{f}(v_r, v_{r'}) \tag{3}$$

An exemplary interaction function is shown in Fig. 2(b), displaying the interaction of oriented edges. Starting from the centered feature v_r, features with a similar orientation w.r.t. to their distance have a higher compatibility than nearly perpendicular features in close proximity.

3 Hallucinating Features

We strive to use the CLM binding dynamics to "imagine" well matching amendments for sparse geometric shapes. In order to achieve this goal, we apply the CLM to a set of geometric shapes, let it converge and then induce hallucinated features to evaluate their compatibility using the binding dynamics.

The induction of hallucinated features is currently done without a priori knowledge about the distribution of known features from the CLM grouping. Therefore the search space is narrowed to a finite set and the search for well matching hallucinated features is currently done with a "brute force" approach. For each possible element the compatibility to the existing groups is evaluated.

To evaluate the compatibility of a new feature vector v_{new}, an interaction vector

$$m = (\mathbf{f}(v_{new}, v_0), \mathbf{f}(v_{new}, v_1), \ldots, \mathbf{f}(v_{new}, v_r))^T \tag{4}$$

is created to extend the interaction matrix $M_{rr'}$:

$$M_{new} = \begin{pmatrix} M_{rr'} & m \\ m^T & 1 \end{pmatrix} \tag{5}$$

The support for the hallucinated feature from the existing neurons is then calculated as:

$$x_{v_{new}\alpha} = m^T \cdot \vec{x}_\alpha \tag{6}$$

4 Preliminary Results

To evaluate the proposed approach, we applied a CLM with ten layers to a set of sparse circles composed of oriented edges, as depicted in Fig. 3(a), with an

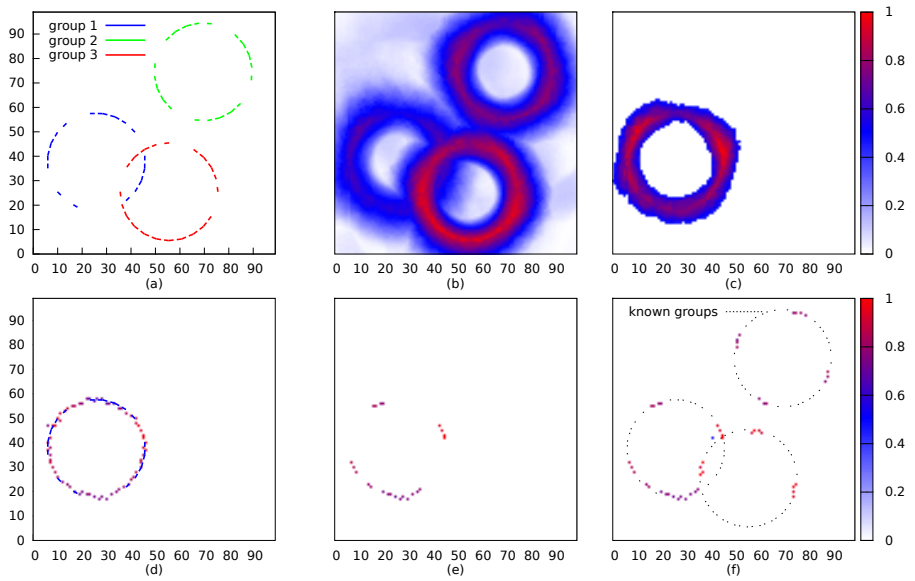


Figure 3: (a) CLM grouping of three sparse circles. (b) Activity of hallucinated features merged over all layers. (c) Activation from hallucinated features for a single layer after applying a threshold of 0.5. (d) Local maximum in a 5×5 neighbourhood with known features from group 1. (e) Known features subtracted from previous maxima. (f) Best matching features in unoccupied areas for all layers with known features from (a).

oriented edge defined by a 2D position (x, y) and orientation θ. Different layers are represented with different colors. For each position in the 100×100 input space 36 features with different orientations in a range from $0°$ to $175°$ were imagined and evaluated for their compatibility with existing groups.

Fig. 3(b) shows the maximal activity at each position (x, y) over all possible orientations θ. Please note that Fig. 3(b) is furthermore a combination of all layers.

To reduce the noise from not well matching hallucinated features, a single layer is selected in Fig. 3(c) and a threshold is applied, which sets every activity smaller than 0.5 to zero.

To narrow down the result of the thresholding, a filter which selects the maximum in a 5×5 neighbourhood is utilized. This new local maximum is then used as point of origin for a new filtering step in which already visited positions are omitted. This enables the filter to "follow" local maxima. Of course hallucinated features in close proximity to already known features are selected by this filter, too. This is shown in Fig. 3(d), where the result of the filtering process is overlaid with group 1 from Fig. 3(a).

In an additional step depicted in Fig. 3(e), hallucinated features in close proximity to existing groups are removed, leaving only good amendments. Fig. 3(f)

shows the above mentioned steps for all groups, including the original CLM grouping results of from Fig. 3(a). In the interests of clarity, all groups are displayed with the same symbols.

These results show the feasibility of using the CLM dynamics in conjunction with hallucinated features to amend sparse informations.

5 Conclusion

Inducing hallucinated features into the CLM opens an interesting foundation to amend sparse informations, which is not only limited to the completion of geometric shapes but can also be generalized to much more complex scenarios. For example given the action segmentation from [2], it is imaginable to use the CLM for action generation, given a set of incomplete action segments.

It also introduces a lot of new questions for research, e.g. how to overcome the current "brute force" approach to initially generate hallucinated features, as well as a more general technique to finally find good amendments in contrast to the feature specific method presented here.

Also of interest will be a combination of learning the lateral interactions as presented in [4] with amendment through hallucinated features to gain a better generalization.

Acknowledgements

This work has been conducted within and funded by the German collaborative research center "SFB 673: Alignment in Communication" granted by Deutsche Forschungsgemeinschaft.

References

[1] A. Desolneux, L. Moisan, and J.M. Morel. *From gestalt theory to image analysis: a probabilistic approach*. Springer Verlag, 2008.

[2] M. Pardowitz, R. Haschke, J. Steil, and H. Ritter. Gestalt-based action segmentation for robot task learning. In *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS Int. Conf. on*, pages 347–352. IEEE, 2008.

[3] H. Ritter. A spatial approach to feature linking. In *Int. Neural Network Conference, Paris*, 1990.

[4] S. Weng, H. Wersing, J.J. Steil, and H. Ritter. Learning lateral interactions for feature binding and sensory segmentation from prototypic basis interactions. *Neural Networks, IEEE Transactions on*, 17(4):843–862, 2006.

[5] H. Wersing, J.J. Steil, and H. Ritter. A competitive-layer model for feature binding and sensory segmentation. *Neural Computation*, 13(2):357–387, 2001.

How to Evaluate Dimensionality Reduction?

Wouter Lueks[1,2], Bassam Mokbel[1],
Michael Biehl[2], Barbara Hammer[1]

[1)] CITEC – Center of Excellence
for Cognitive Interaction Technology
Bielefeld University
D-33501 Bielefeld
Germany
{wlueks|bmokbel|bhammer}
@techfak.uni-bielefeld.de

[2)] Johann Bernoulli Institute
for Mathematics and Computer Science
University of Groningen
P.O. Box 407, 9700 AK Groningen
The Netherlands
m.biehl@rug.nl

1 Introduction

The amount of electronic data available today as well as its complexity are becoming larger and larger in virtually all application domains. In consequence, humans can no longer directly deal with such collections by inspecting the text files. Rather, automated tools are required to support humans to extract the relevant information. One core technology is given by data visualization: relying on one of the most powerful senses of humans, it offers the possibility to visually inspect large amounts of data at once and to infer relevant information based on the astonishing cognitive capabilities of humans in visual grouping and similar.

Dimensionality reduction techniques constitute one important method in understanding high-dimensional data because they directly produce a low-dimensional visualization from high dimensional vectorial data. Consequently, many dimensionality reduction techniques have been proposed in recent years. In the beginning, these methods were primarily linear, like principal component analysis (PCA), corresponding to low cost dimensionality reduction techniques with a well founded mathematical background. However, linear techniques cannot preserve relevant nonlinear structural elements of data. Therefore, recently, more and more non-linear methods like Isomap [1], locally linear embedding (LLE) [2] and stochastic neighbor embedding (SNE) [3] have become popular, see the overview article [4], for example.

With more and more dimensionality reduction techniques being readily available, the user faces the problem which of the methods to choose for the current application. Usually, different techniques can lead to qualitatively very different results. In addition, virtually all recent techniques have parameters to control the mapping. Hence, depending on the parameters of the method, even a single DR technique can lead to qualitatively very diverse results. It is usually not clear whether the different results correspond to different relevant structural as-

pects in the data which are possibly partially contradictory in low dimensions, or whether some of the methods and model parameters are less suited to preserve the relevant structural aspects in the given data set. At the same time, it is very hard for humans to judge the quality of a given mapping and the suitability of a specific technique and choice of parameters by visual inspection: the original data set is not accessible to the user due to its high dimensionality such that a human cannot compare a given visualization to ground truth easily. Therefore, there is a need to develop formal measures which judge the quality of a given mapping of data. Such formal measures should evaluate in an automated and objective way in how far structure of the original data corresponds to the structure observed in the low dimensional representation.

Several quality criteria to evaluate dimensionality reduction have been proposed in recent years, see [5] for an overview. As for dimensionality reduction itself, the problem to define formal evaluation criteria for dimensionality reduction suffers from the ill-posedness of the task: it is not clear a priori which structural aspects of the data should be preserved in a given task. Most quality measures which have been proposed recently measure in some way in how far ranks of data points or neighborhood relationships correspond to each other in the original space and the projection. Two recent quality measures offer a general approach and constitute frameworks that include earlier measures as special cases [5, 6]. Regarding this general framework, it becomes apparent that also the quality measure eventually depends on the needs of the user since the user can specify depending on the task which aspects of the data are particularly relevant.

Therefore, there is a need for intuitive and easily accessible quality measures which allow the user to determine the precise form of the measure based on the current application. The co-ranking matrix [5] already goes into this direction by pointing out the relevance of the neighborhood rank which the user believes is important. We will discuss that the global quality measure which has been derived based on this framework in the work [5] does not correspond to an intuitive interpretation by the user: on the one hand, it depends on absolute values of the ranks rather than the deviation of the ranks, i.e. the actual 'errors' made by a DR method. On the other hand, it relies on a single parameter only, the size of ranks taken into account, which controls both aspects, which errors are tolerated and which neighborhood relations are considered interesting for the mapping. We show in a simple example that this error measure leads to unexpected values which do not correspond to an intuitive understanding.

As an alternative, based on the co-ranking framework, we propose a different family of quality criteria which are based on the values of the rank errors rather than the absolute values of the ranks. This family is parameterized by two parameters which control the size of errors which are tolerated on the one hand and the size of the neighborhood of points which should be mapped faithfully by the dimensionality reduction on the other hand. This way, the user can intuitively control the resulting quality measure. We also propose an intuitive way to link formal quality criteria to a given visualization such that the user can immediately see which parts of the mapping are trustworthy.

2 Dimensionality Reduction and Quality Measures

Dimensionality reduction techniques are used for visualization by mapping a high-dimensional dataset $\Xi = \{\xi_1, \ldots, \xi_N\}$ to a low-dimensional dataset $X = \{x_1, \ldots, x_N\}$. By design and via parameters DR methods specify which properties should be maintained by the mapping. Some techniques are based on global mappings such as linear techniques, which determine a matrix to reduce the dimensionality of the data set by a linear transformation, or topographic mapping such as the self-organizing map [7] which parameterize a mapping by a lattice of prototypes in the data space. Many modern non-linear techniques are non-parametric: they map a given set of data points directly to their respective projections without specifying a functional form. This way, the mapping has large flexibility and highly non-linear effects can be obtained.

Non-parametric dimensionality reduction is often based on a cost function or objective, which evaluates in how far characteristics of the original data ξ_i are preserved by the projections x_i. Appropriate projections are then determined minimizing this objective with respect to the parameters x_i. For example, t-SNE maintains the neighborhood probabilities in both spaces, while LLE tries to place points in such a way that locally linear neighborhoods are maintained. See e.g. the article [8] for a general formalization of popular non-parametric dimensionality reduction techniques in this way.

Thus, for non-parametric DR methods, there is often a close relationship of an objective function which in some way or the other evaluates the quality of a mapping, and a DR algorithm which actually finds projections such that the quality is optimized. Here we are interested in a quality criterion which evaluates the quality of DR mappings in a uniform and intuitive way, and which provides a parameterization which can intuitively be controlled by the user. Thereby, it is irrelevant whether the resulting objective also leads to a simple optimization scheme. First approaches in this direction have been proposed based on the co-ranking framework in the work [5].

2.1 The Co-ranking Framework

Here we introduce the co-ranking framework as proposed by Lee and Verleysen [5]. Let δ_{ij} be the distance from ξ_i to ξ_j in the high-dimensional space. Analogously, d_{ij} is the distance from x_i to x_j in the low-dimensional space. From these distances we can compute the ranks of the neighbors for each point. The rank of ξ_j with respect to ξ_i in the high-dimensional space is given by

$$\rho_{ij} = |\{k \mid \delta_{ik} < \delta_{ij} \text{ or } (\delta_{ik} = \delta_{ij} \text{ and } 1 \leq k < j \leq N)\}|,$$

where $|A|$ is the cardinality of the set A. Analogously, the rank of x_j with respect to x_i in the low-dimensional space is given by

$$r_{ij} = |\{k \mid d_{ik} < d_{ij} \text{ or } (d_{ik} = d_{ij} \text{ and } 1 \leq k < j \leq N)\}|.$$

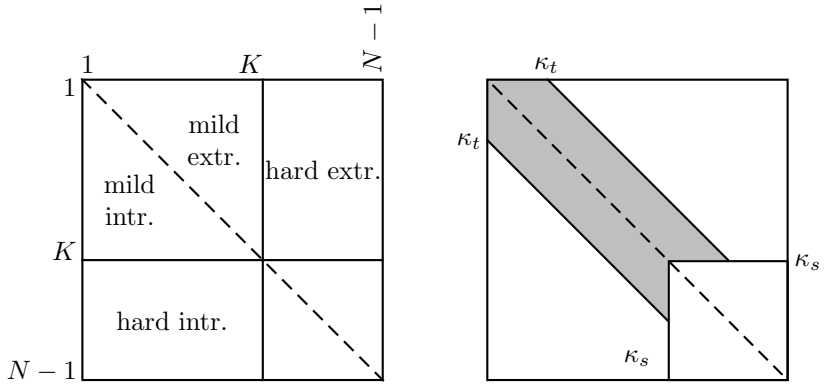


Figure 1: Weighting scheme for the co-ranking matrix: Points which are outside the relevant region (lower right rectangle) are disregarded. In the original framework as proposed by Lee and Verleysen [5] (left), points which stay within K neighborhoods are considered as correct projections. Conversely, the points the rank error of which is small enough are considered as correct in the new proposal (right).

Many existing quality criteria measure in how far ranks of points are preserved while projecting to a low dimensional space. This way, local relationships are evaluated without referring to irrelevant issues such as e.g. scaling of the data.

To generalize such measures, the co-ranking matrix Q [9] is defined by

$$Q_{kl} = |\{(i,j) \mid \rho_{ij} = k \text{ and } r_{ij} = l\}|.$$

Errors of a DR mapping correspond to off-diagonal entries of this co-ranking matrix. A point j that gets a lower rank with respect to a point i in the low-dimensional space than in the high-dimensional space, i.e. $\rho_{ij} > r_{ij}$, is called an *intrusion*. Analogously, if ξ_j has a higher rank in the low-dimensional space it is called an *extrusion*. As shown in Figure 1, intrusions and extrusions correspond to off-diagonal entries in the upper or lower triangle, respectively.

Usually, a DR mapping is not used to map all relationships of data faithfully. Rather, the preservation of local relationships is important. Hence Lee and Verleysen distinguish two types of intrusions/extrusions, those within a K-neighborhood, which are benevolent, and those moving across this boundary, which are malign with respect to quality.

Based on this setting, a simple quality measure can be defined: it counts the number of points that remain inside the K-neighborhood while projecting, i.e., all points which keep their rank, and all mild in- and extrusions:

$$Q_{\mathrm{NX}}(K) = \frac{1}{KN} \sum_{k=1}^{K} \sum_{l=1}^{K} Q_{kl}. \tag{1}$$

The normalization ensures that the quality of a perfect mapping equals one.[1] The quality criterion is very similar to the local continuity meta-criterion (LCMC) that was proposed by Chen and Buja [10]. Note that the range of this quality measure depends on K, i.e. the size of the neighborhood which should be preserved by a DR mapping. Often, a graph of the quality values over all possible K (or a sufficient selection thereof) is plotted.

This co-ranking framework offers a very elegant framework to formalize quality criteria based on rank errors. However, it has a severe drawback: *The quality (1) depends on the number of rank errors in a region of interest only disregarding the size of rank errors.*

Let us have a look at the evaluation measure (1). A region of interest, i.e. a rank K is fixed, following the idea that ranks which are very large (larger than K) are not meaningful in the data space and the projection space and thus, they can be disregarded. The second role of K is to define what is regarded an error: an error occurs if and only if the region of interest in the original space and the projection space does not coincide. Hence the actual size of the rank error is not important. Rather, it is checked whether ranks $\leq K$ keep this property while projecting and vice versa. As an extreme case, points which change their rank from 1 to K are not counted as an error, while points which change their rank from K to $K+1$ do.

This choice of $Q_{\mathrm{NX}}(K)$ can lead to curious situations, which demonstrate the unintuitive character of the parameter K. Consider the pairwise swapping of points that is shown in Figure 2. The number of points can be chosen arbitrarily. Examining the structure quickly shows that the maximum rank error between these permutations is at most 4 (for example, when the base point moves left, and the other point moves right)[2]. Intuitively, if we consider rank error sizes up to 4 as acceptable, this mapping is perfect. This is, however, not the case when looking at $Q_{\mathrm{NX}}(5)$: there are still errors. In fact, for every value of K there will be some point that moves from, for example, rank K to a slightly higher rank, and therefore be counted as an error. This is also confirmed by the graph in Figure 3(a) which displays the quality. Even for large K, this does not reach 1 as long as K is strictly smaller than the number of data points. It is hardly possible to intuitively predict $Q_{\mathrm{NX}}(K)$ even for simple mappings.

A look at the co-ranking matrix in Fig. 2 indicates the underlying structure in this case. Since the rank error is always smaller than 5, only 4 off-diagonals of the co-ranking matrix are non-vanishing. The quality measure (1), however, only sums over rectangular parts of the co-ranking matrix. This observation also suggests how the quality (1) can be altered to lead to a more intuitive parameterization: rather than rectangular parts only, it should focus on a limited number of off-diagonals corresponding to the size of the rank deviation which is considered acceptable.

Now, we will formalize this consideration by first reformulating the quality

[1] Instead of expressing the quality, one could define a measure of error analogously as $1 - Q_{\mathrm{NX}}()$.

[2] Note that in case of a tie in distances, the point with the lowest alphabetical letter gets the lowest rank.

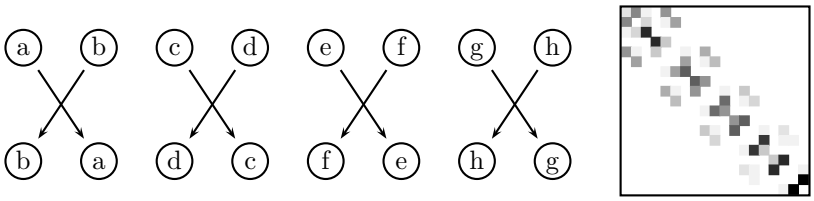


Figure 2: On the left is an example mapping from a one-dimensional set of points to a slight reordering. Since the points are only pairwise swapped, the changes in rank-distances are rather small. For the same setup with 20 points, this is confirmed by the co-ranking matrix that is shown on the right. White indicates a zero value, while black corresponds to the maximum value in the matrix.

(1) such that the two different roles of K become apparent, and then generalizing this formalization such that an explicit control of the region of interest and the tolerated rank error becomes possible.

Formally, the first role of K can be captured by a rank-significance function $w_s : R \times R \to [0, 1]$ that determines, for any pair of points i and j the extent $w_s(\rho_{ij}, r_{ij})$ to which their rank error should be taken into account.

$$w_s(\rho_{ij}, r_{ij}) = \begin{cases} 0 & \rho_{ij} > K \wedge r_{ij} > K \\ 1 & \text{otherwise.} \end{cases}$$

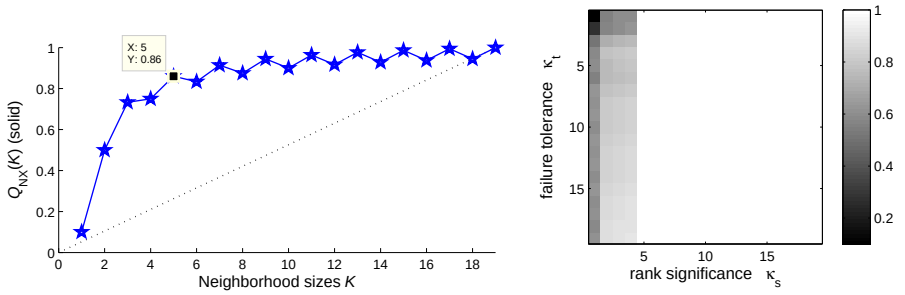
To describe the second role of K, we use a function $w_t : R \times R \to [0, 1]$ that determines the weight of the rank error E_{ij} for points i and j based on their ranks ρ_{ij} and r_{ij}.

$$w_t(\rho_{ij}, r_{ij}) = \begin{cases} 1 & \rho_{ij} \leq K \wedge r_{ij} \leq K \\ 0 & \text{otherwise.} \end{cases}$$

This counts the overlap of the K neighborhoods in the original space and the projection space, respectively. The quality is proportional to the number of points in the region of interest which are benign:

$$Q_{\mathrm{NX}}(K) = \frac{1}{2KN} \sum_{i=1}^{N} \sum_{j=1}^{N} w_s(\rho_{ij}, r_{ij}) \cdot w_t(\rho_{ij}, r_{ij}). \tag{2}$$

As discussed before, a problem is that this function depends on the actual ranks and not on the rank error. Directly examining Figure 1 confirms this. A point with high-dimensional rank 1 and low-dimensional rank K is acceptable, although it has an absolute rank error of $K-1$. On the other hand, a point that has high-dimensional rank K and low-dimensional rank $K+1$ is not acceptable, although its rank error is only 1.



(a) Quality $Q_{\mathrm{NX}}(K)$ (b) Quality $Q'_{\mathrm{NX}}(\kappa_s, \kappa_t)$

Figure 3: The figure shows quality evaluations for the mapping presented in Figure 2, on the left with the established measure, on the right with the proposed new measure. Both are evaluated for all possible parameter settings K and (κ_s, κ_t) respectively. The particular position of $K = 5$ is highlighted on the graph in the left figure. $Q_{\mathrm{NX}}(K)$ with $K \geq 5$ does not yield a value of 1, which seems rather unintuitive for the given problem. As expected, the matrix for $Q'_{\mathrm{NX}}(\kappa_s, \kappa_t)$ does have ones, for all $\kappa_s \geq 5$.

2.2 A quality measure based on rank errors

Because of this fact, we propose the following alternative failure tolerance function

$$w_t(\rho_{ij}, r_{ij}) = \begin{cases} 1 & |\rho_{ij} - r_{ij}| \leq \kappa_t \\ 0 & \text{otherwise,} \end{cases}$$

that depends on the rank error rather than the value of the ranks. The cut-off value κ_t determines which error sizes are accepted. We use the same rank-significance function w_s as we derived from [5], but substitute the parameter K by the the cut-off parameter κ_s. Following equation (2) we then get a new quality measure:

$$Q'_{\mathrm{NX}}(\kappa_s, \kappa_t) = \frac{1}{2\kappa_s N} \sum_{i=1}^{N} \sum_{j=1}^{N} w_s(\rho_{ij}, r_{ij}) \cdot w_t(\rho_{ij}, r_{ij}). \tag{3}$$

Because of the normalization, quality values are in the interval $[0, 1]$ with 1 corresponding to a perfect mapping. Figure 1 shows the region of the co-ranking matrix which is taken into account in this quality measure. One might also consider more complex or smooth functions for w_s and w_t than simple cut-offs with κ_s and κ_t respectively and corresponding normalization factors.

The new quality measure $Q'_{\mathrm{NX}}(\kappa_s, \kappa_t)$ depends on two parameters instead of only one K which allow an intuitive access to the parameters: κ_t determines which size of the rank errors which are tolerated, while κ_s singles out which ranks fall into the region of interest. This function can be displayed in a 3 D surface or colored matrix, where the position (κ_s, κ_t) is assigned the value $Q'_{\mathrm{NX}}(\kappa_s, \kappa_t)$, see Figure 3(b) for an example. The matrix in Figure 3(b) shows

all values of $Q'_{\mathrm{NX}}(\kappa_s, \kappa_t)$ for the example in Figure 2. It clearly shows that the maximum quality is reached for all $\kappa_s > 4$.

3 Local quality Assement

The quality criteria introduced in the previous section average the contributions of all points. It can be useful to visually represent the error of a single point, in order to gain insight into local qualitative changes, especially when the deviation of the quality of the mapping in the single parts is very high. This principle has been used, for example, to visualize the topographic distortion of self-organizing maps, where one can display the distance between neurons in the data space as a color in the topographic map, see [7]. Similarly, in the approach [11], the local topographic reliability of dimensionality reduction is displayed.

The quality measure as introduced above naturally gives rise to a local quality which, given a single points, displays the trustworthiness of the map in this area. We propose to use the following error function which sums the contribution of a data point to the quality measure in symmetrized form:

$$Q_i = \frac{1}{4\kappa_s N} \sum_{j=1}^{N} \left[w_s(\rho_{ij}, r_{ij}) \cdot w_t(\rho_{ij}, r_{ij}) + w_s(\rho_{ji}, r_{ji}) \cdot w_t(\rho_{ji}, r_{ji}) \right]. \quad (4)$$

As an example, in Figure 4(a), we show the popular 'swiss roll' benchmark data set. The data is mapped by t-SNE using a high perplexity parameter which produces an 'unfolded' view of the manifold, with some local tearing and distortion, see Figure 4(b). The coloring clearly reveals the tears within the manifold as well as the larger rank errors that occur at the rightmost points caused by 'unrolling' and putting the inner end of the belt far away from its original neighbors on the next spiral loop level. In a real world scenario, where the original data is high-dimensional and its detailed structure is unknown to the user, the coloring of the mapped points may help to understand local characteristics of the mapping.

References

[1] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 2000.

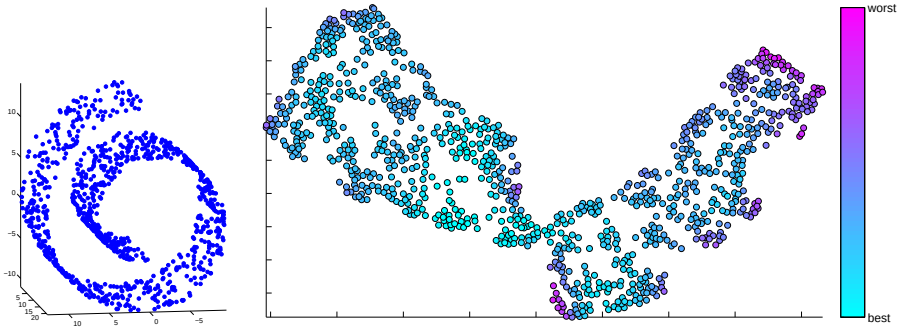
[2] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *SCIENCE*, 290:2323–2326, 2000.

[3] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[4] John A. Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

[5] John A. Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomput.*, 72(7-9):1431–1443, 2009.

[6] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res.*, 11:451–490, 2010.



(a) Original in $\mathbb{R}^3$ (b) t-SNE mapping, colored by pointwise error contributions

Figure 4: The left figure shows the swiss roll benchmark data set in its original three dimensions, the right side picture shows a two-dimensional embedding of t-SNE, using a perplexity parameter of 50. Every data point is colored by its amount of contribution to rank errors, see equation 4, with $\kappa_s = 96$ and $\kappa_t = 70$. Altogether, the visualization by t-SNE seems to be appropriate for the given data, since the global manifold structure is largely maintained. This is confirmed by the error-coloring. However, the coloring reveals local tearing of the manifold, as well as errors on the right border, caused by 'unrolling' the inner end of the spiral. The latter occurs when referring to the standard Euclidean distance in the original space. Taking geodesic distances, this effect vanishes and only the part where the manifold is teared is highlighted.

[7] A. Ultsch and H.P. Siemon. Kohonen's self organizing feature maps for exploratory data analysis. In *Proceedings of INNC'90, International Neural Network Conference, Dordrecht, Netherlands*, pages 305–308. Kluwer, 1990.

[8] Kerstin Bunte, Michael Biehl, and Barbara Hammer. Dimensionality reduction mappings. In *IEEE Symposium on Computational Intelligence and Data Mining*, pages pp. 349–356, 2011.

[9] John A. Lee and Michel Verleysen. Quality assessment of dimensionality reduction based on k-ary neighborhoods. In *JMLR Workshop and Conference Proceedings Volume 4: New challenges for feature selection in data mining and knowledge discovery*, volume 4, pages 21–35, 2008.

[10] Lisha Chen and Andreas Buja. Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association*, 104(485):209–219, 2009.

[11] Michael Aupetit. Visualizing distortions and recovering topology in continuous projection techniques. *Neurocomputing*, 70(7-9):1304 – 1330, 2007. Advances in Computational Intelligence and Learning - 14th European Symposium on Artificial Neural Networks 2006, 14th European Symposium on Artificial Neural Networks 2006.

Recognizing Human Activities Using a Layered HMM Architecture

Michael Glodek Lutz Bigalke Günther Palm
Friedhelm Schwenker
University of Ulm, Institute of Neural Information Processing, D-89069 Ulm

The development of so-called computer systems shows a tendency towards being aware of the user and the environment, offering a broad variety of interactions with the user. It is already feasible to detect faces, estimate the pose of the user, recognize emotion from speech, be aware of the environment and augment it with additional information [VJ02, KBKL09, WLBN06, SOSP08]. In this context, Oliver et al. proposed a layered cognitive system to detect human activities based on a multitude of modalities [OHG02]. The architecture detects complex activities based on a stream of crisp class assignments rendered by classifiers on the preceding layer. The current study investigates the possible increase in performance by passing the uncertainty of the class decision instead of crisp class assignments to the next layer. Oliver et al. utilized hidden Markov models (HMM) to detect the class on each layer. In order obtain a distribution over classes an alternative classifier, namely the conditioned HMM (CHMM) has been examined. The CHMM has the same structure as the latent-dynamic conditional random fields (LDCRF) [MQD07]. Unlike the LDCRF, which is based on a Markov network, the CHMMM is based on a directed graph. Compared to the HMM each latent random variable is additionally influenced by a class node. The input-output hidden Markov model (IOHMM) proposed by Bengio et al. [BF96] is, except of two aspects, also closely related to the CHMM. On the one hand, the IOHMM has additional edges connecting the class with the observation node for each time step. On the other hand, the CHMM models in analogy to the HMM and in contrast to the IOHMM the observations as emissions. The strong relation to HMM has the advantage that scientific contributions achieved for HMM can be applied without effort to the CHMM.

1 Layered Architecture

Every layer of the architecture detects sequential patterns and passes the classification results to the next layer which is then used as an input to detect complexer patterns. To obtain a sequence on every layer a sliding window is utilized such that the concatenated outputs of a layer render a new sequence for the next layer. Hence, each layer compresses the information given such that a classification on a larger time-scale is tractable.

Oliver et al. utilized crisp class decisions achieved by comparing the likelihoods of the HMM to feed the next layer and suggested log-likelihoods to be used in order to incorpo-

rate the uncertainty of the decision in the next layer. However, using log-likelihoods will lead to serious numerical problems since they are ranging potentially over $\mathbb{R}$ and tend to take very large negative values. According to our previous experience, it is difficult to train a layer based on the log-likelihoods in our numerical experiments. However, a distribution over classes is better suited to pass the uncertainty to the next layer.

The presented study focuses on the comparison of crisp class assignments and probability distribution over classes. To obtain a crisp class decision by the means of HMM, for each class $\mathbf{y}$ a HMM $\lambda_{\mathbf{y}=\mathbf{y}}$ is trained and the class having the highest likelihood is chosen by evaluating

$$\tilde{y} = \text{argmax}_{\mathbf{y}\in\mathbf{y}} \Big(p(\tilde{\mathbf{X}}|\lambda_{\mathbf{y}=\mathbf{1}})p(\mathbf{y}=\mathbf{1}), \ldots, p(\tilde{\mathbf{X}}|\lambda_{\mathbf{y}=|\mathbf{y}|})p(\mathbf{y}=|\mathbf{y}|) \Big)$$

where $\tilde{\mathbf{X}}$ denotes the windowed observations of the underlying layer and $p(\mathbf{y}=\mathbf{y})$ the class prior. The concatenated class assignments $\tilde{\mathbf{y}}$ is then used to feed the discrete HMM of the next layer. The CHMM λ on the other side renders a distribution over classes $p(\mathbf{y}=\mathbf{y}|\mathbf{X},\lambda)$ such that the distribution itself can be passed to the next layer in form of a vector

$$\tilde{\mathbf{y}} = \Big(p(\mathbf{y}=\mathbf{1}|\tilde{\mathbf{X}},\lambda), \ldots, p(\mathbf{y}=|\mathbf{y}|\,|\tilde{\mathbf{X}},\lambda) \Big).$$

2 Conditioned HMM

The CHMM extends the HMM by additional random variables $\mathbf{y}$ which are directly influencing the latent random variables $\mathbf{w}$. The Markov chain of the CHMM is illustrated in Figure 1. The nodes colored in dark gray represent the always accessible observations $\mathbf{X}$

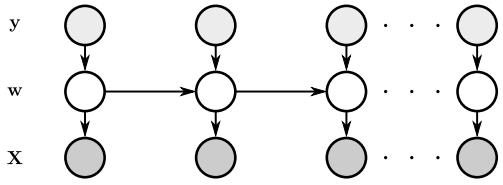


Figure 1: **Markov chain of the CHMM.**

while the light gray nodes represent the sequence of labels $\mathbf{y}$ corresponding to the observations which are only given at training. The white nodes are the hidden states $\mathbf{w}$ mediating between the labels and the observations. The joint probability is given by

$$\begin{aligned} p(\mathbf{X},\mathbf{w}|\mathbf{y},\lambda) &= p(\mathsf{w}_1=\mathbf{w}_1|\boldsymbol{\pi}) \cdot \Big(\prod_{t=2}^{T} p(\mathsf{w}_t=\mathbf{w}_t|\mathsf{w}_{t-1}=\mathbf{w}_{t-1},\mathbf{A}) \Big) \\ &\quad \cdot \Big(\prod_{t=1}^{T} p(\mathbf{x}_t=\mathbf{x}_t|\mathsf{w}_t=\mathbf{w}_t,\theta) p(\mathsf{w}_t=\mathbf{w}_t|\mathsf{y}_t=y_t,\mathbf{C}) \Big) \end{aligned}$$

where $\lambda = \{\boldsymbol{\pi}, \mathbf{A}, \mathbf{C}, \theta\}$ denotes the set of parameters of the corresponding probabilities. Exact inference can be performed using the forward-backward algorithm and can be used to determined the parameters of the model utilizing the expectation-maximization algorithm [KF09]. The conditioned distribution over the classes is obtained by

$$p(\mathbf{y}|\mathbf{X}) \quad = \quad \frac{p(\mathbf{X}|\mathbf{y})p(\mathbf{y})}{\sum_{\mathbf{y}\in\mathbf{y}} p(\mathbf{X}|\mathbf{y})p(\mathbf{y})}.$$

3 Experiment

Our experiments aim at detecting complex activities based on actions detected on a lower layer. The lower layer recognizes the actions based on the positions of the head, hand and a specific object i.e. a cup. The extracted features consists of the pairwise inter object distances and the velocity of movement of each object with direction, represented as an angle, and magnitude The considered classes for the activities in this application are: *drink from cup* (DC), *relocate cup* (CR) and *other activity* (OA_2). These activities are composed by different atomic actions, i.e. *grab cup* (GC), *move cup* (MC), *incline cup* (IC), *release cup* (RC), *scratch head* (SH) and *other action* (OA_1). For example the activity *drink from cup* is composed of the actions *grab cup*, *move cup*, *incline cup*, *move cup* and *release cup*. Two distinct data sets have been created for training and testing. The *pre-segmented data set* consists of labeled sequences for each layer and has been used for training and finding a valid model. Testing is performed using an *uncut data set* such that a real-time application is simulated and a sliding window is required.

The results of the pre-segmented and uncut data set for HMM and CHMM are shown in Table 1. The left hand side of the table shows the error rate of the training set using a ten-fold cross-validation. Although the results of the training set are promising, the test set

Table 1: **Validation and test results (pre-segmented and uncut data set respectively)**. Error rates (standard deviation) in percent and F_1 measures of discrete HMM and CHMM for each layer.

	Pre-segmented data set		Uncut data set	
First layer				
	HMM	CHMM	HMM	CHMM
Error %	4.63 (0.74)	3.82 (1.30)	48.88	50.24
F_1 GC	0.96 (0.02)	0.96 (0.02)	0.42	0.27
F_1 MC	0.94 (0.01)	0.95 (0.02)	0.59	0.57
F_1 IC	0.98 (0.03)	0.98 (0.02)	0.23	0.27
F_1 RC	0.94 (0.02)	0.95 (0.02)	0.41	0.26
F_1 SH	0.98 (0.03)	0.99 (0.03)	0.46	0.26
F_1 OA_1	0.97 (0.02)	0.98 (0.02)	0.65	0.65
Second layer				
	HMM	CHMM	HMM	CHMM
Error %	0.00 (0.00)	1.96 (0.00)	66.62	35.93
F_1 CD	1.00 (0.00)	1.00 (0.00)	0.35	0.70
F_1 CR	1.00 (0.00)	1.00 (0.00)	0.18	0.20
F_1 OA_2	1.00 (0.00)	1.00 (0.00)	0.43	0.70

(right hand side) reveals that the real-time application is by far more challenging. While the HMM based on the crisp class assignments achieves only an error rate of 66.62% the CHMM, which detects the activities based on the uncertainty of the lower layer, obtains an error rate of 35.93%.

Future work will aim at exploring the presented architecture in a complexer setting with a focus on human-computer interaction. The outputs of the layered architecture shall furthermore be integrated in a framework which incorporates uncertainty into symbolic information processing. A promising approach to be investigated here is the Markov logic network (MLN) [TD08].

Acknowledgement The presented work was developed within the Transregional Collaborative Research Centre SFB/TRR 62 "Companion-Technology for Cognitive Technical Systems" funded by the German Research Foundation (DFG).

References

[BF96] Y. Bengio and P. Frasconi. Input-Output HMMs for Sequence Processing. *IEEE Transactions on Neural Networks*, 7(5):1231–1249, 1996.

[KBKL09] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics. *Eurographics State of the Art Reports*, pages 119–134, 2009.

[KF09] D. Koller and N. Friedman. *Probabilistic graphical models: Principles and techniques*. The MIT Press, 2009.

[MQD07] L.P. Morency, A. Quattoni, and T. Darrell. Latent-Dynamic Discriminative Models for Continuous Gesture Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[OHG02] Nuria Oliver, Eric Horvitz, and Ashutosh Garg. Layered Representations for Human Activity Recognition. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, 2002.

[SOSP08] S. Scherer, M. Oubbati, F. Schwenker, and G. Palm. Real-time emotion recognition from speech using echo state networks. *Artificial Neural Networks in Pattern Recognition*, pages 205–216, 2008.

[TD08] S. Tran and L. Davis. Event Modeling and Recognition Using Markov Logic Networks. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 610–623. Springer, 2008.

[VJ02] P. Viola and M. Jones. Robust real-time object detection. volume 57, pages 137–154, 2002.

[WLBN06] U. Weidenbacher, G. Layher, P. Bayerl, and H. Neumann. Detection of head pose and gaze direction for human-computer interaction. *Perception and Interactive Technologies*, pages 9–19, 2006.

Unsupervised Segmentation of Object Manipulation Operations from Multimodal Input

Alexandra Barchunova * Robert Haschke † Ulf Großekathöfer ‡ Sven Wachsmuth §
Herbert Janssen ¶ Helge Ritter ‖

Abstract

We propose a procedure for unsupervised segmentation of bimanual high-level object manipulation operations in multimodal data. The presented procedure applies a two-stage segmentation and a selection step to observation sequences. We employ an unsupervised Bayesian method to identify homogeneous segments which correspond to primitive object manipulation operations. The data is recorded using a contact microphone, a pair of Immersion CyberGloves and ten pressure sensors positioned on the fingertips.

The assessment of the temporal correctness and structural accuracy of the segmentation procedure has showed satisfactory results. We have achieved an average error of 0.25 seconds in comparison to the actual segment borders. The examination of the structural accuracy for a given parameter combination has showed only insignificant deviation of the generated segmentation structure from the corresponding test data.

Finally, we sketch an application of our method to unsupervised learning and representation of object manipulations.

1 Introduction

An important objective of today's robotics research is to enable robots to interact with humans in everyday scenarios. Within this area, we focus on the topic of autonomous learning and identification of bimanual object manipulations from sequences. In order to participate in a simple interaction scenario or learn from a human, a robot needs the ability to autonomously single out relevant parts of the movement executed by a human. It also needs a mechanism to identify and organize these parts. In order to address this requirement, we propose a novel approach for unsupervised identification of high-level bimanual object manipulation operations within action sequences. Inspired by the fact, that humans employ different information sources – like hearing, proprioception, haptics and vision – to fulfill this task, we propose a multi-modal approach to segment and identify action sequences. To this end we consider an audio signal, tactile sensor readings from all finger tips, and hand postures acquired by CyberGloves [1].

Analysis of various sensor readings describing the human hand dynamics during manual interaction have been conducted recently by different researchers [2, 3, 4]. In general, one is interested in autonomous identification of action primitives in the context of imitation learning and human-machine interaction [5, 6]. Within this domain, Matsuo et al. focused on force feedback [7] while a combination of different sensors like CyberGlove, Vicon or magnetic markers and tactile sensors has been used by [8], [4] and [9]. In [10] a bimanual approach is described.

Despite the variety of sensors and approaches used in action segmentation and identification, one modality, namely the audio signal, has been mostly ignored in this domain. However, in the area of speech recognition it is well known, that the audio signal not only transmits the mere verbal content, but also conveys temporal structure of interactions and actions [11].

Our past work has been concerned with unsupervised segmentation and classification of raw motion data and its linear projection into a low-dimensional space [12]. The experiments within this preliminary study have showed that the absence of structural analysis of object manipulation sequences restricts the scenario to a small set of distinct and unambiguous manipulations. To tackle more complex and ambiguous action sequences, we employ a Bayesian segmentation method to analyze the sequential structure.

In our scenario, during a considerable number of simple high-level object manipulations (e.g. grasping, shifting, shaking, stirring or rolling) application of force is naturally accompanied by a sound. We exploit this fact by performing

*Bielefeld University, Cor-Lab, abarch@cor-lab.uni-bielefeld.de
†Bielefeld University, Neuroinformatics, rhaschke@techfak.uni-bielefeld.de
‡Bielefeld University, Ambient Intelligence, ugrossek@techfak.uni-bielefeld.de
§Bielefeld University, Applied Informatics
¶Honda Research Institute Europe
‖Bielefeld University, Neuroinformatics

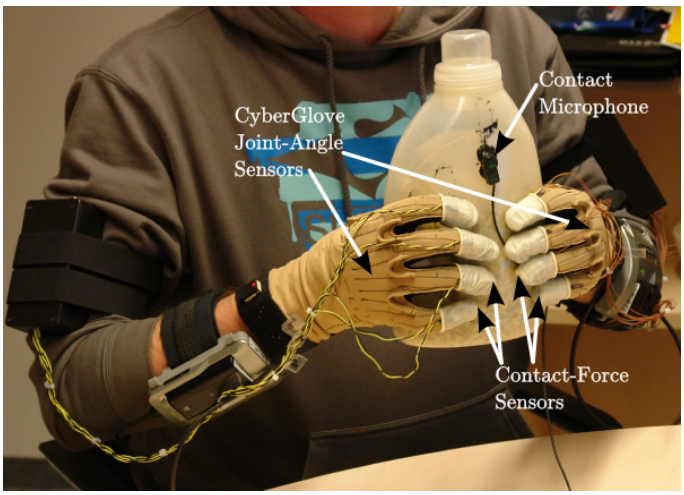


Figure 1: Experimental setup: a subject wearing contact and joint angle sensors performs manipulation operations with a (instrumented) plastic bottle provided with a contact microphone.

segmentations based on the analysis of the audio signal structure and of contact forces recorded on the fingertips. The resulting segmentation solely depends on the temporal structure of the data and is invariant to absolute data values, way of grasping or the manipulation object. Our method does not employ any specific knowledge about the parts of the action sequence. Furthermore, it does not require a large set of domain-specific heuristics describing each action primitive as is commonly the case in similar approaches [8, 4, 13].

We evaluate our method in an everyday scenario in which a human subject performs several object manipulation operations with a large non-rigid plastic bottle with a handle. In this evaluation, we assess the performance of the segmentation method w.r.t. the accuracy of the generated segment borders and the overall structure of the produced segmentation. Additionally we briefly outline the results of applying an unsupervised learning procedure, which has been used in similar tasks ([14, 15]), to cluster the identified action segments. The developed method is applicable to interactive scenarios such as imitation learning, cooperation and assistance.

The rest of this paper is organized as follows: Sec. 2 explains the acquisition of action sequences within the scenario. Sec. 3 introduces the two steps of the proposed method: preprocessing (Sec. 3.1) and segmentation (Sec. 3.2). In Sec. 4, we discuss our evaluation method and experimental results of the segmentation procedure, and report on an application of the proposed method as a preprocessing stage of an action recognition module (Sec. 5). Sec. 6 concludes the paper with a brief discussion and outlook.

2 Scenario and Experimental Setup

In our scenario, a human subject performs sequences of simple uni- and bi-manual object manipulations with a gravel-filled plastic bottle[1], as can be seen in Fig. 1.

We use the following sensors to record multimodal time series of the performed action sequences (corresponding modality names used in formulas appear in parenthesis):

- one contact microphone attached to the bottle (`a`). The contact microphone focuses on in-object generated sound, ignoring most environmental noise.
- 2×24 joint-angles calculated from the measurements of two Immersion CyberGlove devices (`j`: both hands, `jl`: left hand, `jr`: right hand). The Immersion CyberGlove II devices output sensors values describing the configurations of finger- and palm-joints.
- 2×5 FSR pressure sensors attached to the fingertips of each CyberGlove (`t`: both hands, `tl`: left hand, `tr`: right hand) record the contact forces.

This collection of sensors yields a 29-dimensional ($24 + 5$) representation for each hand in addition to a scalar audio signal. The subject was told to perform a sequence of basic manipulation actions in fixed order as listed in the following enumeration. To obtain ground truth for later evaluation, the beginning or end of an action within a sequence was signalled to the subject as explained in Sec. 4. To achieve a rich variance of timing between trials,

[1] The use of gravel instead of liquid is due the necessity of a distinct audio signal and also safety concerns.

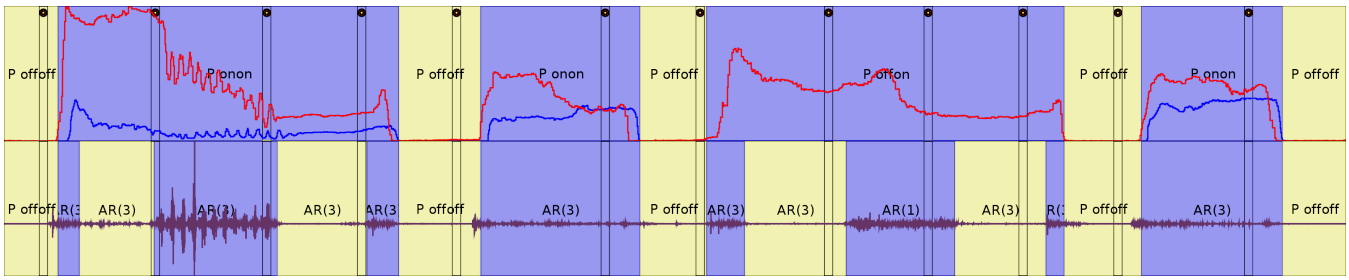


Figure 2: Initial segmentation and "subordinate" sub-segmentation for one multimodal time series. The first row shows the result of applying Fearnhead's method with joint threshold models of the tactile data of both hands (see Sec. 3.2.1 for details). The segmentation is overlaid with the tactile signals of both hands. The second row shows the refinement of the segmentation in the first row that is computed by applying Fearnhead's method to the audio signal within each "contact" segment (see Sec. 3.2.2 for details). In the second row the segmentation is overlaid with the audio signal.

the desired duration of most elements was sampled from a Gaussian distribution with standard deviation of 0.5s as specified in parentheses:

1. pick up the bottle with both hands (2 s + η_1)
2. shake the bottle with both hands (.7 s + η_2)
3. put down the bottle (1 s)
4. pause (1 s)
5. unscrew the cap with both hands (1.2 s + η_3)
6. pause (1 s)
7. pick up the bottle with right hand (2 s + η_4)
8. pour with right hand (1 s + η_5 + 1 s + η_5)
9. put down the bottle (1 s)
10. fasten the cap with both hands (1.2 s + η_6)

The random variables $\eta_i \sim \mathcal{N}(0, .5\ s)$ denote the randomized timing of subsequences. The overall length of the time series accumulates to approximately 30 seconds.

3 Method

The recorded time series of multiple sensor modalities capture complex and high-dimensional descriptions of action sequences. The focus of this paper is on segmentation and selection of relevant data. Furthermore, we briefly outline a subsequent clustering step to demonstrate that the proposed method can serve as a preprocessing stage for an unsupervised learning procedure to recognize action primitives. In the following paragraphs we describe the segmentation process based on the tactile and audio modalities.

3.1 Preprocessing

In a preprocessing step, the original audio-signal is normalized to a given variance range with respect to the amplitudes of individual samples. The signal is also subsampled and recording artifacts are removed by discarding samples whose amplitude exceeds a specified threshold. We use the resulting processed audio signal in the segmentation step described in Sec. 3.2. This preprocessing is necessary for successful segmentation due to the characteristics of Auto-Regressive models used in the segmentation process.

3.2 Segmentation

In our two-stage segmentation approach, we use tactile information to obtain a preliminary rough split of the sequence into subsequences of "object interaction" and "no object interaction". This analysis of hand-object contacts uses force data from both hands. Subsequences that have been recognized as "object interaction" are analyzed in detail w.r.t. qualitative changes of the audio signal in order to refine the rough segmentation.

In both stages, the segmentation is performed by applying Fearnhead's method [16] for unsupervised detection of multiple change-points in time series. The input to Fearnhead's algorithm is a time series $y_{1:T}$ [2] and a set of models $\mathcal{M}$ for homogeneous subsequences. The output is a set of integer change-points $1 < \tau_1 < \cdots < \tau_N < T$ at which qualitative changes in the data $y_{1:T}$ occur. A set of such change-points is dual to segmentation of the form $(y_{s_i:t_i})_{1\leq i\leq N+1}, s_1 = 1, t_i = \tau_i = s_{i+1}, t_{N+1} = T$ which partitions the data into $N + 1$ subsequences. Within the probabilistic framework of Fearnhead's algorithm, the optimal segmentation is obtained by maximizing the Bayesian posterior[3] $P(y_{1:T} \mid \tau_{1:N})P(\tau_{1:N})$ which consists of a likelihood term and a prior distribution over segmentations $P(\tau_{1:N})$. In a common choice of this prior, the probability $P(\tau_{1:N})$ is composed of probabilities of individual segment lengths which are computed according to the geometric distribution $P(l) = \lambda(1 - \lambda)^{l-1}$. Consequently, the prior is characterized by a single parameter λ that is reciprocal to the expected segment length under a geometric distribution, i. e. $\lambda \propto 1/u$ where u is the expected length of subsequences. Once λ has been chosen, neither the number of change-points N nor any information regarding their positions have to be specified in advance. Due to the difference in the input content of the time series, both segmentation steps of our procedure specify their own method for λ calculation. We use the notation λ^α in the first stage and λ_{sub} in the second, subordinate segmentation stage. These values will be discussed in the respective subsections.

In addition to the prior distribution of segment lengths, the algorithm employs a finite set of models $\mathcal{M}$ to represent different regimes in segments of the time series. Each model $m \in \mathcal{M}$ assigns marginal likelihoods $P(y_{s:t} \mid m)$ to segments $y_{s:t}$, $1 \leq s < t \leq T$, of the time series. Prior probabilities $P(m)$ are associated with all models. In this paper, we only consider sets of up to four models with uniform prior distributions.

In the following two subsections, we describe the application of Fearnhead's algorithm to two different subsets of the available modalities in combination with two suitable sets of models $\mathcal{M}$ and $\mathcal{M}_{\text{sub}}$. The two-stage application of the segmentation procedure and the modality-specific local models constitute the main contributions of this paper.

3.2.1 Segmentation based on Tactile Modalities

The first step performs a rough joint analysis of the tactile signals of both hands. For the application of Fearnhead's method in this stage, we set the value of the prior parameter $\lambda^\alpha = 1/T^\alpha$ for each trial α of length T^α. Although this choice conceptually corresponds to a single expected segment, it turned out to be suitable for small numbers of segments as well. This has been confirmed by the experimental evaluation. The analysis uses four pairs of threshold models. Each model of a pair describes the tactile state, i.e. "object contact" vs. "no object contact", for one hand. We denote the "object contact"-models with capital-letter subscripts: m_L and m_R for the left and the right hand respectively. The corresponding notation for the "no object contact"-models is m_l and m_r.

The marginal likelihood, that a model fits to a time series segment $y_{s:t}$ is of the form:

$$P(y_{s:t} \mid m_l) = {p_o}^n \quad \text{and} \quad P(y_{s:t} \mid m_L) = {p_o}^{u-n}$$

where p_o is the fixed probability that a sample does not fit the model (in this case m_l), $u = t - s$ is the segment length, and n is the number of such samples within the time series segment, e.g. $n = \left|\{y_{k|\texttt{tl}} > \gamma \mid s \leq k < t\}\right|$. The parameter γ specifies the threshold for recognizing contact.

Combining these individual models, $\mathcal{M}$ consists of the following four joint models: "no contact for both hands" (m_{lr}), "contact for left hand only" (m_{Lr}), "contact for right hand only" (m_{lR}), and "contact for both hands" (m_{LR}). The marginal likelihoods of these joint models are computed as products of the individual likelihoods, e.g.:

$$P(y_{s:t} \mid m_{lR}) = P(y_{s:t} \mid m_l) \cdot P(y_{s:t} \mid m_R)$$

Assignments of the four joint contact-state models to segments in a computed segmentation are illustrated in the first row of Fig. 2. Contact assignments identify parts of the time series that are directly associated with object interactions. Accidental movements of one or both hands between manipulations are separated from manipulation operations in this step. Such movements occur for instance during an approach phase prior to grasping. The assignment of models to segments can be exploited to exclude joint and tactile modalities (`jl`, `tl` for left hand; `jr`, `tr` for right hand) of "inactive" hands from subsequent processing steps (e.g. clustering, see Sec. 5). For example, the assignment of $m_{l,R}$ to a segment $y_{s:t}$ leads to the corresponding data fragment $y_{s:t|\texttt{jl,tl}}$ being excluded. When the model $m_{l,r}$ is assigned, the segment in question can be ignored entirely.

[2] We use the notation $x_{a:b} \equiv (x_a, \ldots, x_b)$. We use $x_{\_|\texttt{mod}}$ to indicate the restriction to modality `mod`.

[3] We suppress $P(y_{1:T})^{-1}$, which is irrelevant for the maximization.

In contrast to a pointwise application of threshold methods, Fearnhead's method – even when used with threshold models – is not sensitive to noise which could otherwise lead to severe oversegmentation with many extremely small segments. On the downside, Fearnhead's method requires the specification of a prior distribution on segment lengths, i.e. the λ parameter.

3.2.2 Sub-segmentation of Object Contact Segments Based on Audio Signal

In this subordinate segmentation step, all segments produced and not discarded in the previous step are sub-segmented using Fearnhead's method. This time, the audio signal in the constructed sub-segments is assumed to be produced by Auto-Regressive (AR) models of order 1, 2 or 3: $\mathcal{M}_{\text{sub}} = \{AR(1), AR(2), AR(3)\}$ [16]. Thus the sub-segmentation is formed by selecting segments that exhibit homogeneous oscillatory properties within the audio modality. In contrast to the procedure outlined in the previous paragraph, the value of the segment length distribution parameter λ_{sub} is fixed. In our evaluation (Sec. 4), a suitable value for λ_{sub} is estimated by means of a grid-search.

The sequential application of segmentation and selection steps yields a set of segments that are characterized by constant contact topology in respect to overall hand activity as well as homogeneous characteristics of the audio signal. The assignment of "object contact" threshold models from the first segmentation step is discarded in this final segmentation result since it is not exploited in further steps.

4 Experimental Results

4.1 Data Pool

We recorded 50 trials of the action sequence described in Sec. 2 with a single subject in one session. In principle, the structure of all these trials should be identical except the timing. However, it turned out to be rather difficult for the subject to perform such a high number of trials without structural variations. As a result, some trials exhibit structural differences like missing or additional tactile contacts or repeated actions. However, we made no attempt to correct these irregularities.

In the domain of unsupervised recognition of human actions, there is no established methodology for quantitative evaluation. To avoid time-consuming hand-labelling of our data, we generate and use randomized action time schedules for all trials in the following way: for a particular trial, audio cues are emitted according to the corresponding schedule to mark the start or end times of actions. We rely on the subject to react to these cues and align their executed actions as closely to them as possible. The audio cues (similar to dial tones) are provided via head phones to prevent their presence in the recorded audio modality.

Each cue consists of a sequence of four beep sounds[4] : the first three are preparatory and allow the subject to anticipate the fourth signal which notifies the associated event (beginning or end of action execution) to the subject. The timing of cues is derived from the structure described in Sec. 2 by randomizing the duration of individual actions. We record timestamps of generated cues, as an indication of the timing of scheduled actions. In our evaluation, we use these recorded cue timestamps as ground-truth. This enables us to assess the correctness of timing and the number of generated segments. Note that this ground-truth is an approximation due to differences between cues and the actual timing of action execution. We write $c^{\alpha}_{i,j}$, $j \in \{1,2,3,4\}$ to denote the point in time at which the j-th signal of the i-th cue is emitted in trial α [5].

4.2 Segmentation Quality

In this section, we analyze the results of applying the two-stage segmentation described in Sec. 3.2 to the data discussed above. We assess the obtained segmentations w.r.t. the following three aspects: the number of calculated segments, the number of undetected segment borders and the timing accuracy of the generated segmentation. We perform this assessment of our procedure for a large set of combinations of the adjustable parameters. These are: the contact threshold value γ, the segment length distribution parameter λ_{sub} and the range parameter for the normalization of the audio signal ρ. In our experiments, we have used all possible combinations of $\lambda_{\text{sub}} \in 10^{\{-4,-5,-6,-7,-8\}}$, $\rho \in \{6,8,10,12\}$ and $\gamma \in \{15,20,30,40,50,60,70,80\}$. The goal of these experiments is to assess the respective influences of the parameters and to find a parameter combination that yields segmentations most close to the ground-truth in all three abovementioned aspects.

To obtain quantitative results, the cue-based ground truth data is exploited as follows: First, for each main cue signal $c^{\alpha}_{i,4}$ within each trial α, the temporally closest generated change-point is searched within a temporal window around the start time of the cue signal. Depending on whether timing or oversegmentation is assessed, we use a

[4] The preparatory cue signals are .1 s long, the pause between signals is .2 s long and the main signal lasts .2 s.

[5] When the trial is clear from context or not important, we drop the superscript and write cue times as just $c_{i,j}$.

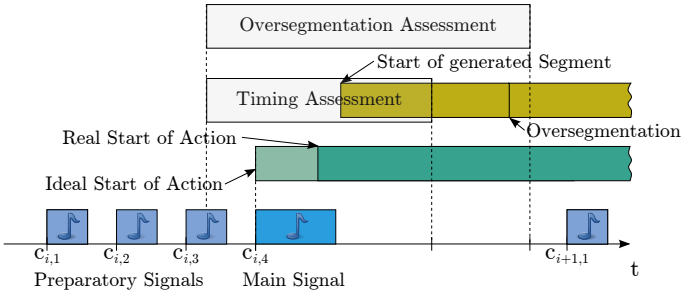


Figure 3: Temporal relations between cues, actions and generated segments. The execution of an action by the subject is expected to start (light green bar) at the beginning of the cue signal $c_{i,4}$, but the actual beginning of the execution usually deviates (dark green bar). In our evaluation, we try to find automatically generated segments (dark yellow bar) that correspond to these actions in different areas (light gray boxes) around the cues (See Sec 4.2 for details).

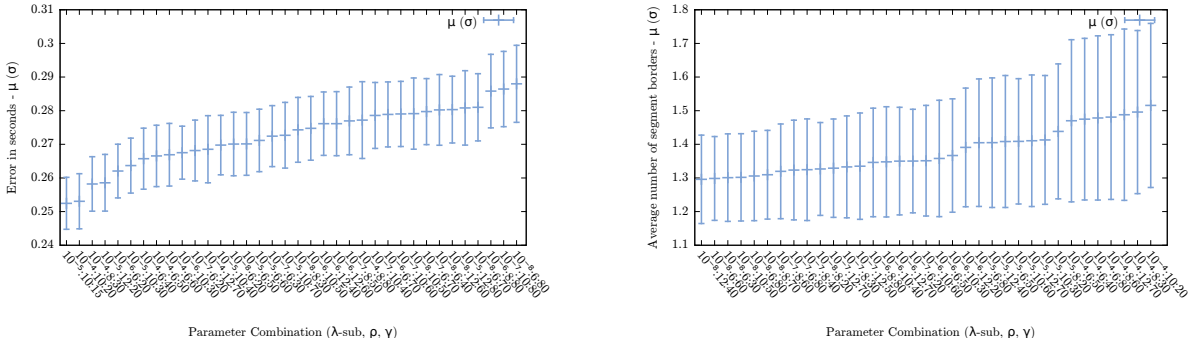


Figure 4: Left: Average distances between cues and corresponding estimated segment borders. Distances (*y-axis*) are in seconds and sorted by increasing average error for different combinations of the parameters λ_{sub}, ρ and γ (*x-axis*). Right: Sorted average number of estimated segment borders for actions (*y-axis*) w.r.t. parameter combinations (*x-axis*). In both figures, averages are over all trials and all actions for each parameter combination.

smaller or larger window (See Fig. 3 for an illustration of the procedure). If such a change-point can be identified, the temporal distance to the cue signal serves as an indicator of the accuracy of the segmentation. Otherwise the manipulation performed in response to the cue signal is considered as not having been detected.

Fig. 4 (left) shows the timing deviation of the estimated segments from the ground-truth. In order to determine the timing error for a given parameter combination, we first average over all trials resulting in twelve values, one for each cue. We calculate the displayed values of mean and variance by additionally aggregating all twelve cues. The window size around each cue used in the evaluation was set to $[c_{i,3}, (c_{i,4}+c_{i+1,1})/2]$ (see Fig. 3). The resulting average time-intervals between the cues and the closest estimated segment border are sorted in the order of ascending error. From Fig. 4 (left) it can be easily seen, that segment borders generated by the proposed method are extremely robust w.r.t. all parameters. The average error lies in the range 0.25 to 0.29 seconds. We observe lower error values in conjunction with higher values of λ_{sub} and lower values of γ. The higher error values co-occur with smaller values of λ_{sub} and larger values of the γ parameter. We note that the remaining minimal error of approx. 0.25 seconds might originate from the subject's need to adapt the hands before executing the scheduled movement. The parameters that yielded the best results in this experiment were $\lambda_{\text{sub}} = 10^{-5}$ and $\gamma = 15$.

The goal of the following experiment is to evaluate the number of segments generated for each cue. Fig. 4 (right) illustrates the dependency of the average number of estimated segment borders on the parameters. Average and variance values are calculated analogously to Fig. 4 (left). However, the environment used to estimate the number of candidates for one cue is set to be $[c_{i,3}, (c_{i,4} + c_{i+1,3})/2]$. Thus the whole sequence is covered by the calculation (see Fig. 3). This experiment shows a strong dependency between the amount of oversegmentation and the parameter λ_{sub}. Smaller values of λ_{sub}, yield fewer candidates within a cue environment. We observed the best results for $\lambda_{\text{sub}} = 10^{-8}$. This parameter has a clear and a considerable influence on the structure of the resulting segmentation. Despite the increase in deviation of timing from the ground-truth of about 0.01 seconds, we choose the parameter set $\lambda_{\text{sub}} = 10^{-8}, \rho = 12, \gamma = 40$ for further calculations.

Fig. 5 (left) shows the cue-specific average number of generated segments for the abovementioned parameter set.

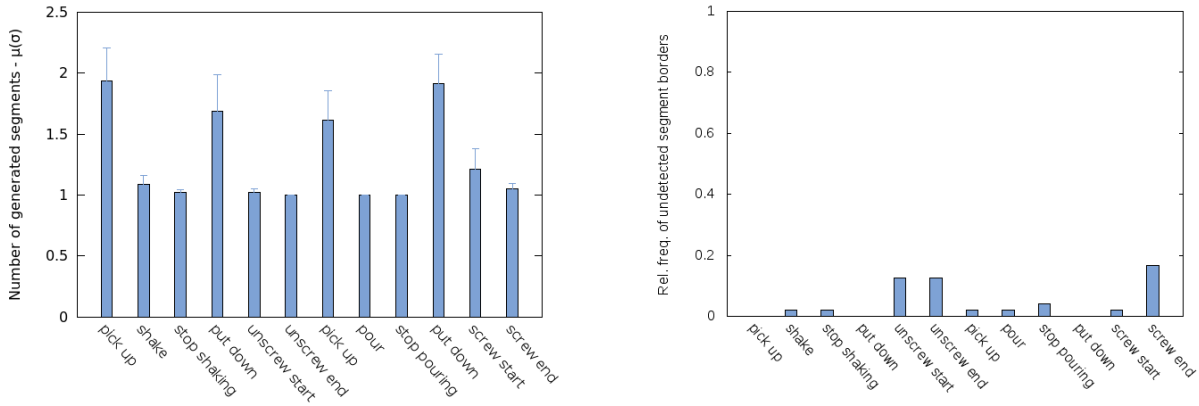


Figure 5: Left: Average number of segment borders (*y-axis*) for each action (*x-axis*) for the parameter combination $\lambda_{\text{sub}} = 10^{-8}, \rho = 12, \gamma = 40$. Right: Relative frequencies of undetected segment borders (*y-axis*) for each action (*x-axis*) for the aforementioned combination of parameters. Note that some actions consist of multiple sub-actions for which no ground-truth information is available (See Sec. 4.2 for details).

In this figure one can clearly differentiate between two groups of events: double events and single events. The first group contains for example *pick up and lift* and *put down*. The start of these actions is marked by a cue, but the duration is so short that no end-cue can be issued correctly to signal the end of the action to the subject. Thus the average number of generated change-points close to two is almost optimal. The second group contains single events like *start shaking*, *end shaking*, *start unscrewing* or *end unscrewing*. In this group, the beginning or the end of the action is marked by the cue. Thus the average number of generated change-points, approximately one, is close to optimal as well. The average and the variance values are computed over all trials. Fig. 5 (right) shows the cue-specific average relative frequency of undetected segments. The high likelihood of detection failures for the *screwing* event is possibly due to the incorrect execution timing of the subject.

5 Application Example for Unsupervised Learning with OMMs

We consider the segmentation method we presented and evaluated in the previous sections as a building block for more sophisticated unsupervised methods. To support this claim, we briefly outline an unsupervised procedure for identification and representation of action primitives based on the proposed method.

To perform identification and representation of action primitives, segments which contain semantically similar actions have to be grouped and models of these groups have to be formed. We address both tasks by embedding the concept of Hidden Markov Models (HMM), which yields good results in representation and modeling of sequential data, in a clustering approach. In the procedure sketched here, we use Ordered Means Models [17], an efficient variant of HMMs with flexible left-to-right topology and Gaussian emission densities.

From the perspective of unsupervised clustering and representation, the output of the proposed segmentation method is a set of multimodal data sequences $\{y^\beta\}_{1 \leq \beta \leq B}$ that are unlabeled w.r.t. the trials and actions from which they originate. The application of OMMs to partition such a dataset into k groups in an unsupervised manner, can be considered a special case of the well-known k-means clustering. OMMs $\lambda_1, \ldots, \lambda_k$ are used as the associated prototypes of k clusters. A suitable distance function then is the negative log-likelihood that a sequence y^β is generated by an OMM λ_j: $d(y^\beta, \lambda_j) = -\log P(y^\beta \mid \lambda_j)$. Given this, a k-OMMs clustering algorithm partitions data sequences into k groups by minimizing the objective function

$$E = -\sum_{\beta=1}^{B} \sum_{j=1}^{k} w_{\beta,j} \log P(y^\beta \mid \lambda_j).$$

subject to $w_{\beta,j} \in \{0, 1\}$ and $\forall \beta : \sum_{j=1}^{k} w_{\beta,j} = 1$.

Prior to performing k-OMMs clustering, two preprocessing steps are applied to the output of the segmentation step. Firstly, the time-domain audio signal is replaced by a coarse characterization in the frequency domain. We apply a sliding-window version of the Discrete Fourier Transform to the audio signal and extract ten coefficients of the lowest frequencies from each result. The time series of these coefficients replaces the audio-signal. This transformation is motivated by the fact that the oscillatory nature of the time-domain audio signal is not compatible with the OMM emission models, which assume piecewise constant data with fixed-variance Gaussian noise. Secondly, we assign

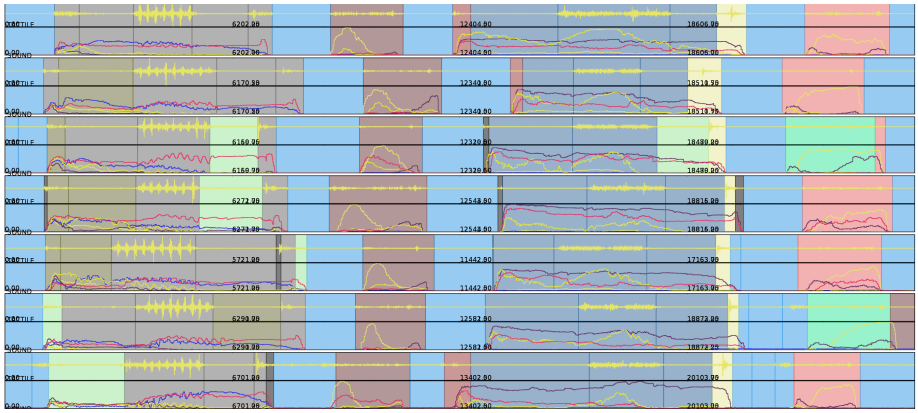


Figure 6: Assignment of labels (*designated by random colors*) to segments according to the best matching model in a small subset of trials. In each row, the segmentation, label assignments, audio signal (*top half*) and tactile information (*bottom half*) is shown. Corresponding segments in adjacent trials do not line up because of the randomized timing.

constant values to modalities associated with an "inactive" hand for the duration of the inactivity. This step is intended to prevent the representation of patterns that are not related to object manipulation in learned OMMs.

Fig. 6 qualitatively shows the result of applying the sketched clustering and learning procedure in the following way: in a training step, twelve OMMs are formed based on segmentations obtained with the presented segmentation method. Then, in a test step, segmented action sequences are classified to the best-matching OMM model. Identically colored segments are considered semantically equivalent.

6 Conclusions and Outlook

In this paper, we presented a novel method for unsupervised segmentation of object manipulation operations in the context of a bimanual interaction scenario. We carried out experiments with a human subject and applied the proposed method to the collected data. The experimental evaluation has showed satisfactory results for both: the segmentation timing and the structural accuracy. These results and an application in an OMM-clustering has showed that the method is able to select primitive object manipulation operations. Future research will be concerned with learning higher level representations of sequences of object manipulation operations. Within this context, the problem of semantically equivalent clusters will be addressed as well. It is also desirable to reduce the number of tunable parameters.

7 Acknowledgments

Alexandra Barchunova gratefully acknowledges the financial support from Honda Research Institute Europe.

References

[1] Http://www.cyberglovesystems.com/products/cyberglove-ii/overview.

[2] K. Bernardin, K. Ogawara, K. Ikeuchi, and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden markov models," *IEEE Trans. on Robotics*, 2005.

[3] R. Dillmann, O. Rogalla, M. Ehrenmann, R. Zollner, and M. Bordegoni, "Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm," in *Proc. ISRR*, 2000.

[4] H. Kawasaki, K. Nakayama, and G. Parker, "Teaching for multi-fingered robots based on motion intention in virtual reality," in *Proc. IECON*, 2000.

[5] B. Sanmohan, V. Krüger, and D. Kragic, "Unsupervised learning of action primitives," in *Proc. Humanoid Robots*, 2010.

[6] W. Takano and Y. Nakamura, "Humanoid robot's autonomous acquisition of proto-symbols through motion segmentation," in *Proc. Humanoid Robots*, 2006.

[7] K. Matsuo, K. Murakami, T. Hasegawa, K. Tahara, and K. Ryo, "Segmentation method of human manipulation task based on measurement of force imposed by a human hand on a grasped object," in *Proc. IROS*, 2009.

[8] M. Pardowitz, S. Knoop, R. Dillmann, and R. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics,*, 2007.

[9] C. Li, P. Kulkarni, and B. Prabhakaran, "Motion Stream Segmentation and Recognition by Classification," in *Proc. ICASSP*, 2006.

[10] R. Zollner, T. Asfour, and R. Dillmann, "Programming by demonstration: Dual-arm manipulation tasks for humanoid robots," in *Proc. IROS*, 2005.

[11] L. Schillingmann, B. Wrede, and K. Rohlfing, "A computational model of acoustic packaging," *Trans. on Autonomous Mental Development*, vol. 1, 2009.

[12] A. Barchunova, M. Franzius, M. Pardowitz, and H. Ritter, "Identification of high-level object manipulation operations from multimodal input," in *Conf. ACIT*, 2010.

[13] R. Zollner and R. Dillmann, "Using multiple probabilistic hypothesis for programming one and two hand manipulation by demonstration," in *Proc. IROS*, 2004.

[14] T. Grosshauser, U. Großekathöfer, and T. Hermann, "New sensors and pattern recognition techniques for string instruments," in *NIME*, 2010.

[15] N.-C. Wöhler, U. Großekathöfer, A. Dierker, M. Hanheide, S. Kopp, and T. Hermann, "A calibration-free head gesture recognition system with online capability," in *Pattern Recognition*, Istanbul, Turkey, 2010.

[16] P. Fearnhead, "Exact and efficient bayesian inference for multiple changepoint problems," *Statistics and Computing*, 2006.

[17] U. Großekathofer, T. Lingner, H. Ritter, and P. Meinicke, "What is a hidden markov model without transition probabilities?" *submitted*, 2010.

Online learning in the loop: fast explorative learning of inverse models in high dimensions

Matthias Rolf and Jochen J. Steil
Research Institute for Cognition and Robotics (CoR-Lab), Bielefeld University

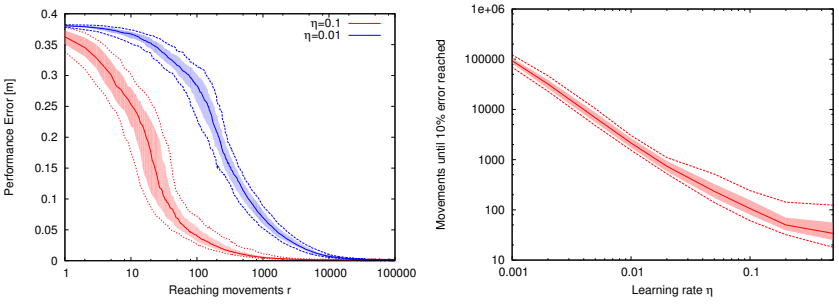
I. Introduction

Learning to generate appropriate actions to achieve some behavioral goal is one of the fundamental problems in cognitive robotics. In most scenarios actions and goals are specified in different spaces. Suppose a robot must reach some goal state $x^* \in \mathbf{X} \subset \mathbb{R}^n$, it can not "set" this state directly but has to generate an action $q \in \mathbf{Q} \subset \mathbb{R}^m$ that will lead to the observation of x^*. The causal relation between actions and their results is typically specified by a forward function $f(q) = x$. If f is not known analytically, the robot has to learn an *inverse model* $g(x^*) = \hat{q}$ that suggests appropriate actions such that the goal is achieved $f(g(x^*)) = x^*$ [1]. An illustrative example are inverse kinematics problems, in which a robot has to chose joint angles q that move the effector (e.g. the robot's hand) towards some position x^*. Learning inverse models can be done by exploring supervisory data (x, q) and obtaining g by regression. Although supervised data can be generated, this problem differs substantially from standard regression schemes including explorative ones like active learning. In general it is not possible to probe a correct solution q^* for a given target x^* directly from the environment. Exploration is only accessible in the reverse direction by applying an action q and observing the outcome x. Designing an exploration method that finds solutions for a set of targets $\mathbb{X}^*$ is far from trivial since $\mathbf{Q}$ is typically high-dimensional and the forward function non-linear. Numerous approaches have been introduced to obtain inverse models by exhaustive exploration in $\mathbf{Q}$, but which is not applicable in high-dimensional domains. Although active learning can alleviate the problem, it still assumes that the entire space can be sampled within the lifetime of an agent, at least to know which regions are irrelevant. Is it possible to explore inverse models efficiently, without attempting a full exploration of the action space?

II. Online Goal Babbling

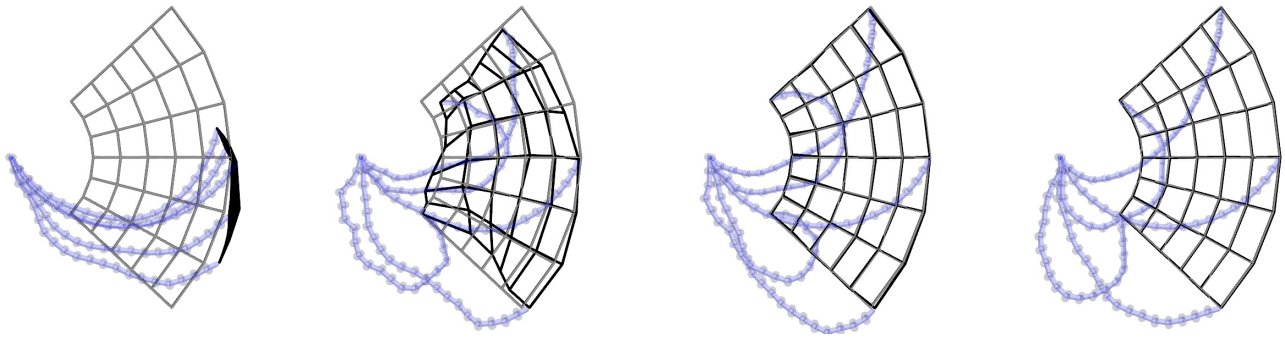
We have previously drawn inspiration from infant developmental studies to tackle this problem. While infant sensorimotor exploration is traditionally modeled as random, it was shown it [2] that infants perform goal-directed explorative movements long before they master some sensorimotor skill. We have modeled this kind of "goal babbling" in [3], [4] and showed that it allows to bootstrap inverse models in very high-dimensional and non-linear domains. The basic exploration method is to use the current inverse estimate g to suggest an action for exploration and add some exploratory noise E:

$$q_t \quad = \quad g(x_t^*, \theta_t) + E_t(x_t^*). \tag{1}$$



(a) (b)

Fig. 1. (a) The performance error decreases rapidly over the number of movements. A ten times higher learning rate results in a speed up of approx. 20. (b) shows the number of movements until the initial error is decreased by 90%.



(a) After 10 movements. (b) After 100 movements. (c) After 1000 movements. (d) After 10000 movements.

Fig. 2. Example of the bootstrapping dynamics for the inverse kinematics of a 20 degrees of freedom arm. Online goal babbling rapidly finds solutions for all target positions (gray grid). The process finds postures (overlayed in blue) that reflect a very smooth and convenient kinematic solution.

For each action, the outcome x_t is observed and the parameters θ of the inverse model are updated with a supervised learning step. This process is iteratively repeated over time until the inverse estimate yields accurate results. At each point in time the inverse model projects the target positions x^* into the potentially high-dimensional action space, which implies that only a low-dimensional manifold is explored at a time. Hence, the method completely avoids a full exploration of $\mathbf{Q}$. The method is therefore highly efficient and was shown to scale up to $m=50$ dimensions.

Recent work [5] shows that the learning speed can be further increased dramatically when *online learning* is applied after each exploration step along continuous target paths x_t^*. Online learning during goal-directed exploration unfolds a *positive feedback loop* in which the learning rate acts as a "gain". Because the exploration is informed by previous samples, the next example will be more informative if a big learning step was applied. This self-information reinforces until an accurate inverse model is found. Experiments using local-linear maps [6] as regression method are shown in Fig. 1 and 2. Fig. 1 shows the learning progress for the inverse kinematics of a five-dimensional robot arm for different learning rates. Increasing the learning rate by a factor 10 increases the overall learning speed by a factor of approx. 20. For high learning rates the error is reduced with enormous speed and reaches a low level already after a few hundred movement paths have been explored. Fig. 2 illustrates the same setup, but with a 20-dimensional problem. Even in this high-dimensional domain an approximate solution is found already after executing 100 movement paths, whereas an exhaustive exploration of 20 dimensions is by far not feasible in the lifetime of any agent. Systematic investigation of the scalability [5] reveals that the exploratory cost (in terms of samples or movements required) is almost constant when the dimensionality is scaled up to $m=50$.

III. Discussion

Our experiments show that the combination of online-learning and an informed, goal-directed exploration process allows to find inverse models in high-dimensional domains in an enormously efficient manner. The learning-rate dependent speedup of the process can not be explained with the traditional view of online-gradients as a stochastic approximation of batch-learning. Rather, it shows how learning scenarios can substantially benefit from online-learning if it is applied in the loop with exploration, where it outperforms batch-learning [3] by orders of magnitudes.

References

[1] D. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends in Cog. Sci.*, vol. 2, no. 9, 1998.
[2] C. von Hofsten, "Eye-hand coordination in the newborn," *Developmental Psychology*, vol. 18, no. 3, 1982.
[3] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *IEEE Trans. Auto. Mental Development*, vol. 2, no. 3, 2010.
[4] ——, "Mastering growth while bootstrapping sensorimotor coordination," in *EpiRob*, 2010.
[5] ——, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *ICDL-EpiRob*, 2011.
[6] H. Ritter, "Learning with the self-organizing map," in *Artificial Neural Networks*, T. Kohonen, Ed. Elsevier Science, 1991.

Learning a Neural Mutlimodal Body Schema: Linking Vision with Proprioception

Johannes Lohmann
Department of Psychologie III
University of Würzburg, Röntgenring 11, 97070 Würzburg
johannes.lohmann@uni-wuerzburg.de

Martin V. Butz
Department of Psychologie III
University of Würzburg, Röntgenring 11, 97070 Würzburg
butz@psychologie.uni-wuerzburg.de

1 Introduction

The brain represents the body in various, interactive, multimodal frames of reference. We investigate how a neural, visual-filter-based arm representation can be linked to a neural, population-encoded angular arm representation. The resulting associative representation should be able to provide bidirectional predictions between the modalities. That is, visual information of an arm should allow the prediction of corresponding joint angles and proprioceptive joint angle perceptions should allow the prediction of corresponding visual information. We present preliminary results limited to one arm limb in 2D space – extensions to multiple limbs and joints are discussed and currently under development.

2 The Architecture

The architecture consists of three parts. First there is a simple representation of joint angles using a neural population code. Second, the visual representation is implemented as a collection of visual filters. Finally, vision and proprioception are associated via Hebbian learning (see Figure 1 for an overview).

Currently the angular space of each joint is represented by an one-dimensional population code – each neuron being centered at one particular angle. If the corresponding joint is moved, a Gaussian shaped activation curvature arises in the population code with the peak at the neuron closest to the current angle.

The visual input is a grayscaled, 2D image of the virtual arm with the first joint (e.g. the "shoulder") in the center of the image. This input is propagated through the framework introduced by [4], which consists of four layers. The first layer, called S_1, applies a collection of Gabor filters with 16 scales and 4 orientations to the grayscaled input image. To enhance performance, we use a CUDA (NVIDIA) implementation of this filter bank, which was developed in

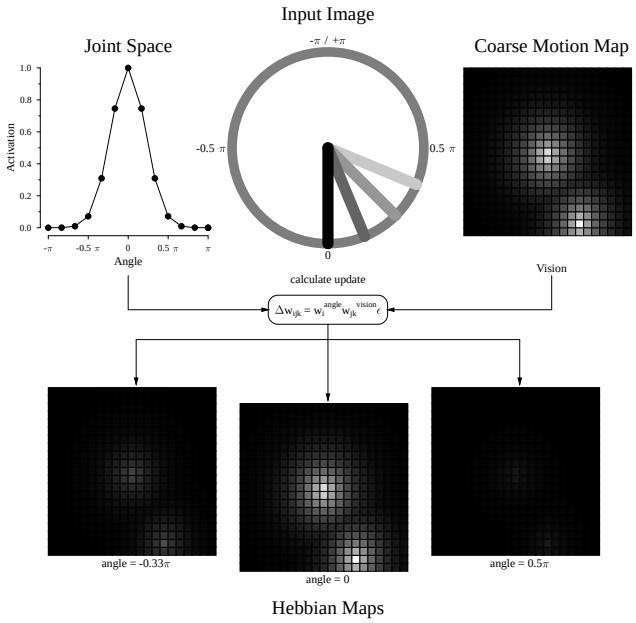


Figure 1: Overview of the proposed architecture.

our lab [3]. The second layer, called C_1, calculates local maxima over position and scale. It reduces the initial 16×4 images to 8×4 images. To reduce the amount of considered data further, we choose one of the scale bands with all orientations for further processing. We applied one of two additional processing steps to the chosen scale band. Either we calculated a flicker or a motion map given the scalebands of the current and the previous trial. A description of these maps can be found in [2]. Finally, the flicker or motion map is used as input for Hebbian learning.

The integration of the joint space and the visual representation is realized via simple Hebbian learning according to the following learning rule:

$$\Delta w_{ijk} = w_i^{angle} w_{jk}^{vision} \epsilon, \tag{1}$$

where Δw_{ijk} refers to the change of a weight in the Hebbian map, w_i^{angle} refers to the activation of the ith node in the population code representing the activation of the angular space, w_{jk}^{vision} refers to the activation in the two dimensional visual representation (j indicates the image width and k the height), and finally ϵ is a weight parameter determining the speed of the adaption.

The current weight and the changed value are added together. Finally, all weights of the map are normalized according to the following equation:

$$w_{ijk}(t+1) \leftarrow \frac{w_{ijk}(t+1)}{\sum_{j=0}^{w} \sum_{k=0}^{h} w_{ijk}(t+1)} \tag{2}$$

The resulting Hebbian maps resemble the correlation between changed joint-angles and according changes in the visual representation. Given these mappings, bidirectional predictions from one modality to the other are possible.

3 First Results

Our evaluation focuses on two measures. First, we investigate the difference between the observed and the predicted activations in joint space. Second, we compare the distance between the peaks of both distributions. This distance is a qualitative measure in neural units, which denotes the distance between the most active nodes rather than an angular value.

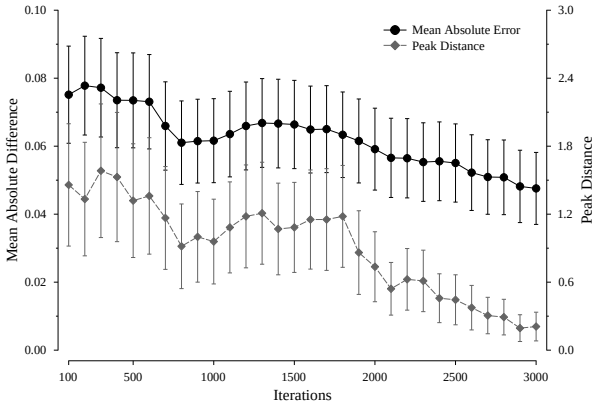


Figure 2: Exemplar learning progress: Peak distances converge to zero distance while the mean absolute differences continue to decline.

Figure 2 gives an example of typical results with the current architecture. For the displayed results the flicker map was used as input for the Hebbian learning. Please note that the results report averages and standard deviations of 5 independent runs. The concordance between the observed and the predicted activation of the joint space improves over time. Due to the applied learning algorithm and the random movements of the simulated arm, the time necessary to achieve a stable solution varies.

4 Outlook

The current architecture is limited to associating one limb in a two dimensional input space. Additionally, the current representation of joint angles via population codes is rather simple and might be replaced by a more sophisticated approach, like dynamic fields [1]. To extend the current architecture to handling multiple joints, at least three challenges have to be tackled:

1. The different joints have to be detected from the Gabor filtered input images.

2. A focus mechanism is necessary to obtain respective joint-centered representations.

3. The co-representations of the respective joints and limbs have to be correctly assigned and maintained over time.

We believe that the first task may be solved by means of the motion map proposed by [2]. Preliminary tests revealed that the highest activations correspond to joint positions. A simple clustering algorithm could be used to obtain the coordinates of the joints in the Gabor filtered image.

To deduce the joint-respective visual information selectively, a focus mechanism will be additionally necessary that does not only extract head-centered representation of the respective limbs and joints, but that also suppresses visual information belonging to other segments of the arm. Such a process could be realized via multiplicative focus models of spatial attention, such as the one introduced by [5]. We believe that this approach would be particularly interesting as it also introduces a working memory model.

The last challenge refers to the binding problem of maintaining a consistent model of individual joints, associating their respective angular and visual representations modularly over time. At the moment, the architecture lacks a sensorimotor model to filter the modular arm state information flow. Additional distance information from a three dimensional arm representation may additionally help to assign the individual arm limbs and joints to the corresponding visual information. Work is in progress to consider and integrate the respective sources of information and modularly bind them over time.

While this is cleary work in progress, we believe that the proposed architecture is able to provide a bidirectional link between vision and proprioception. The representation can be used for anticipatory processing from one modality to the other. With the described extensions, we are certain that the architecture will be able to account for effects of spatial, arm-specific attentional processes as well as for working memory effects of an arm-specific body model. Moreover, the multi-modal representation with its continuous interactions is expected (a) to handle noisy and partially missing sensory information effectively as well as (b) to enable highy flexible behavioral control, which will be able to resolve redundancies and incorporate task-specific priorities on the fly.

References

[1] W. Erlhagen and G. Schöner. Dynamic field theory of movement preparation. *Psychological Review*, 109(3):545–572, 2002.

[2] L. Itti, N. Dhavale, and F. Pighin. Realistic avatar eye and head animation using a neurobiological model of visual attention. In *Proceedings of SPIE 48th Annual International Symposium on Optical Science and Technology*, volume 5200, pages 64–78, 2003.

[3] K. L. Reif and M. V. Butz. Cuda implementation of V1 based on Gabor Filters. Technical report, University of Würzburg, Cognitive Bodyspaces: Learning and Behavior, 2010.

[4] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 29(3):411–426, 2007.

[5] J. Taylor, M. Hartley, and N. Taylor. Attention as sigma-pi controlled ach-based feedback. *IJCNN05*, pages 256–261, 2005.

Object-Class Segmentation using Deep Convolutional Neural Networks

Hannes Schulz and Sven Behnke
University of Bonn, Computer Science VI,
Autonomous Intelligent Systems Group
Friedrich-Ebert-Allee 144, 53113 Bonn, Germany
{schulz,behnke}@ais.uni-bonn.de

Abstract

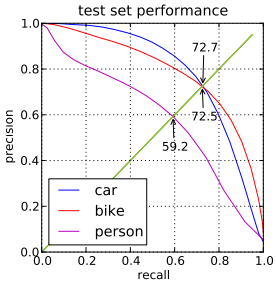
After successes at image classification, segmentation is the next step towards image understanding for neural networks. We propose a convolutional network architecture that outperforms current methods on the challenging INRIA-Graz02 dataset with regards to accuracy and speed.

1 Introduction

Neural networks have long history of usage for image classification, e. g. on MNIST [1], NORB [2], and Caltech [3]. For these datasets, neural networks rank among the top competitors [4]. Despite the success, we should note that these image classification tasks are quite artificial. Typically, it is assumed that the object of interest is centered and at a fixed scale, i. e. that the segmentation problem has been solved. Natural scenes rarely contain a single object or object class. Such images need to be analyzed on various scales and positions for objects of different categories. Object detection and object-class segmentation are thus the logical step towards general image understanding. In this work, we propose variations of the convolutional network for object-class segmentation. We show that with HOG and color input, intermediate outputs and epsilon-insensitive loss error function, we can achieve state-of-the-art accuracy on the INRIA Graz-02 (IG02, [5]) dataset. Due to the efficient reuse of information during convolution as well as a fast GPU implementation, we achieve a framerate of about 10 fps during recall.

2 Related Work

In the deep learning community, research on real images has largely focused on object detection (as opposed to segmentation). For example, using extensive dataset augmentation, pretraining of a sparse encoder, bootstrapping, Kavukcuoglu, Sermanet, Boureau, Gregor, Mathieu, and Cun [6] perform comparably well on the INRIA pedestrian dataset. Licence plates and faces are blurred in Google Street View using a convolutional neural network as part of a larger pipeline. Both techniques are applied in a sliding window, that is, the probability of a pixel



	PR-EEP (%)		
	Car	Bike	Person
Ours	**72.7**	**72.5**	59.2
CRF[8]	72.2	72.2	**66.3**
LIN[9]	62.9	71.9	58.6

Figure 1: Precision/Recall on the IG02 dataset.

being member of a class is determined independently for every pixel and scale. We propose to use a convolutional architecture with multi-scale input, resulting in efficient reuse of data structures. Jain, Bollmann, Richardson, Berger, Helmstaedter, Briggman, Denk, Bowden, Mendenhall, and Abraham [7] proposed convolutional architectures and cost functions to detect boundaries prior to segmentation. We acknowledge that this can improve segmentation results at the borders, but we believe that this should be a second step after finding object or object-class hypothesis. Most current approaches start with an oversegmentation of the image, e. g. Fulkerson et al. [8] classify superpixels based on histograms of features in their neighborhood. Superpixels are often expensive to compute and potentially introduce errors that are hard to correct later. Finally, Aldavert, De Mantaras, Ramisa, and Toledo [9] use a handtuned integral linear classifier cascade to achieve close to very good performance. However, we achieve better accuracy at a higher framerate.

3 Methods

Preprocessing We use eight square feature maps as input. Three maps are the whitened color channels, five maps represent histogram of oriented gradients (HOG180) features. The whitening kernel is derived from 5×5 random patches of the training set. HOG features are calculated at twice the map resolution and then subsampled. We perform these operations at three scales, with resolution decreasing by a factor of two. The teacher, i. e. an image where each pixel is marked with the class it belongs to, is split into one map per class where pixels are 1 when they are in the class and are 0 otherwise. Finally, the teacher is smoothed and downsampled for each scale.

Network Architecture For each scale s, we have input maps m_{si}, two convolutions resulting in maps m_{s1}, m_{s2} and one (intermediate) output layer o_s. The activities of o_s are determined by m_{s1} and fed to m_{s2} with additional convolutions. Between scales, we use maximum pooling to gain some spatial invariance. At each output layer, we measure the pixelwise class error using the epsilon-insensitive loss function $E(x, \hat{x}) = \max(0, |x - \hat{x}| - \varepsilon)^2$, where we fix $\varepsilon = 0.2$. This loss function does not punish small deviations from the target value and essentially acts as a regularizer which plays well with the final thresholding.

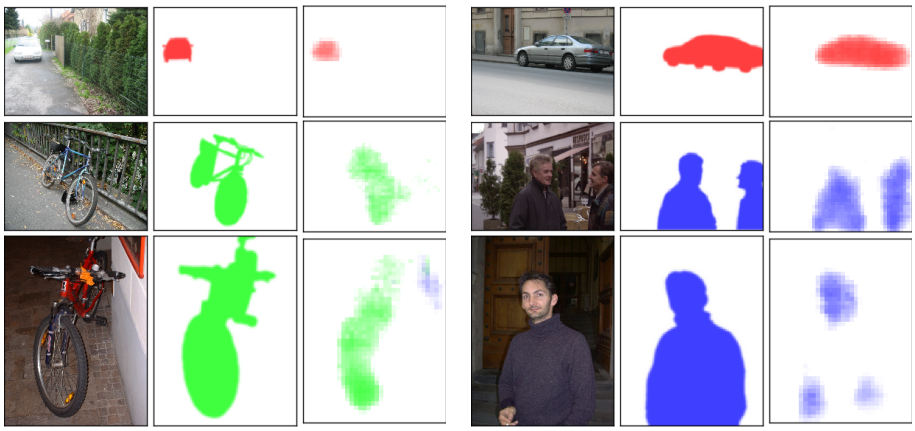


Figure 2: Sample test set object-class segmentations. Left: original image, center: ground truth segmentation, right: our segmentation. The colors red, green, blue represent cars, bikes and persons, respectively. White represents values at or below the EEP thresholds. Large objects, such as on lower right, still have potential for improvement.

The error is backpropagated through the network in the usual way, see e. g. [10]. Errors of intermediate output are scaled by a factor of 0.1. With six hidden layers, the network can be regarded as a "deep" network.

Training We update the weights with the accumulated errors after each epoch using the RPROP [11] algorithm with standard settings, which avoids the need to cross-validate a learning rate. All operations except preprocessing are performed on GPU using the CUV library [12].

4 Results

We test our architecture on the challenging INRIA Graz-02 dataset [5]. The dataset contains images of bikes, cars and persons covering an extremely wide range of pose, scale and lighting. We use the training/testing splits suggested on the dataset website, resulting in (after horizontal mirroring) 958 training and 479 testing images. The images are scaled to 172×172 and squared by horizontal or vertical centering and mirroring into non-occupied space. We use 32 maps on all layers, and filters of size 7×7. Error is measured as in [8] using precision-recall at equal-error rate (PR-EER), at input resolution. After 2000 weight updates, we find that in two categories we outperform state-of-the-art (see Fig. 1). We did not observe overtraining, which we attribute to the regularizing effect of the epsilon-insensitive loss. Some selected segmentations are depicted in Fig. 2. While our method generally performs well on small to medium scales, there is still room for improvement in the precise estimation of currently blurred boundaries. We further observe difficulties in images with e. g. large persons (lower right). Without pre-processing, we are able to process 28 fps, assuming current GPU HOG implementations for preprocessing, we estimate an estimated 10 fps for the trained network.

5 Conclusion

In this paper, we showed that convolutional networks can achieve state-of-the-art performance in object-class segmentation with regards to accuracy as well as speed. We plan to improve our results further using conditional random fields (CRFs) for post-processing.

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[2] Y. LeCun, F. Huang, and L. Bottou. "Learning methods for generic object recognition with invariance to pose and lighting". In: *CVPR*. 2004, pp. 97–104.

[3] L. Fei-Fei, R. Fergus, and P. Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories". In: *CVIU* 106.1 (2007), pp. 59–70.

[4] D. Ciresan, U. Meier, J. Masci, L. Gambardella, and J. Schmidhuber. "High-Performance Neural Networks for Visual Object Classification". In: *CoRR* abs/1102.0183 (2011).

[5] M. Marszatek and C. Schmid. "Accurate object localization with shape masks". In: *CVPR*. 2007.

[6] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun. "Learning Convolutional Feature Hierarchies for Visual Recognition". In: *NIPS 23*. Ed. by J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta. 2010, pp. 1090–1098.

[7] V. Jain, B. Bollmann, M. Richardson, D. Berger, M. Helmstaedter, K. Briggman, W. Denk, J. Bowden, J. Mendenhall, and W. Abraham. "Boundary learning by optimization with topological constraints". In: *CVPR*. 2010, pp. 2488–2495.

[8] B. Fulkerson, A. Vedaldi, and S. Soatto. "Class segmentation and object localization with superpixel neighborhoods". In: *ICCV*. 2009, pp. 670–677.

[9] D Aldavert, R. De Mantaras, A. Ramisa, and R. Toledo. "Fast and robust object segmentation with the Integral Linear Classifier". In: *CVPR*. 2010, pp. 1046–1053.

[10] D. Scherer, A. Müller, and S. Behnke. "Evaluation of pooling operations in convolutional architectures for object recognition". In: *ICANN*. 2010, pp. 92–101.

[11] M. Riedmiller and H. Braun. "A direct adaptive method for faster backpropagation learning: The RPROP algorithm". In: *Neural Networks, 1993., IEEE*. 1993, pp. 586–591.

[12] H. Schulz, A. Müller, and S. Behnke. "Exploiting local structure in Boltzmann machines". In: *Neurocomputing* 74.9 (2011), pp. 1411–1417.

A spiking neural network for situation-independent face recognition

Marco K. Müller, Michael Tremer,
Christian Bodenstein, and Rolf P. Würtz
Institut für Neuroinformatik
Ruhr-University Bochum, Germany

July 30, 2011

1 Introduction

Invariant face recognition refers to estimating the identity of a person irrespective of situation. Human perception is excellent at both finding the identity of a known person and estimating the situation of both known and unknown persons on the basis of a facial image. ("This is John in his twenties in the disco" or "This girl is sunbathing on the beach and seems to enjoy it"). Certainly, the human visual system is good at the *separation* of personal identity and situation. This is possible by using the vast visual experience acquired with many persons in many situations.

From a machine learning point of view, the requirement to recognize identity independent of situation is a case of generalization. However, invariance under even a simple visual transformation such as translation in the image plane is not a generalization performed naturally by known learning mechanisms. Therefore, methods to control the generalization on the basis of examples are required. Paradoxically, this is also a requirement for setting up autonomous learning systems, which can autonomously select learning examples in order to take already learned concepts to a higher degree of abstraction.

Visual invariance can, to a limited degree, be learned from real-world data based on the assumption that temporally continuous sequences leave the object identity unchanged [2, 6, 1, 13]. Slow feature analysis has recently been successfully applied to 3D rotation by [3].

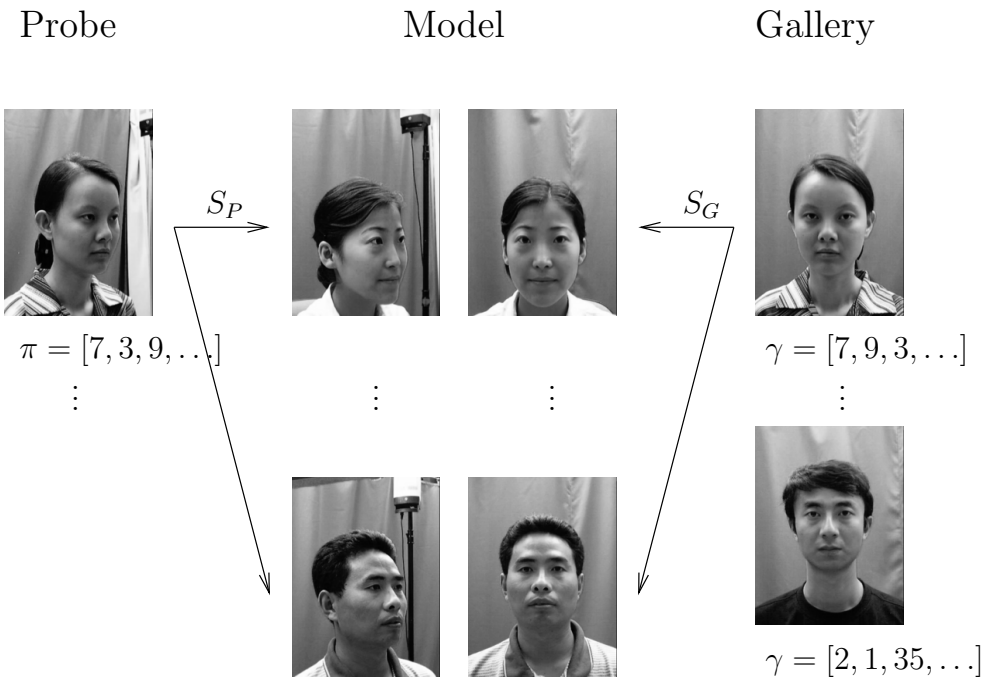


Figure 1: Situation-independent recognition is mediated by a model database of some persons in all situations. Probe and gallery images are coded into rank lists π and γ by their similarities to the models. These rank lists are comparable, while the graph similarities are not (feature indices have been dropped for clarity, and the numbers in the rank lists are just examples).

Nevertheless, all successful recognition systems have the required invariances built in by hand. This includes elastic graph matching [8] and elastic bunch graph matching (EBGM) [14], where the graph dynamics explicitly have to probe all possible variations in order to compare an input image with the stored models. Neural architectures that perform this matching include [15, 9, 16, 7], with the more recent ones being massively parallel and able to explain invariant recognition with processing times comparable to that of the human visual system. These methods work fine for the recognition of identity under changes in translation, scale, and small deformations, including small changes in three-dimensional pose.

We here briefly review a system that can learn invariances in a moderately supervised way from a set of examples of individual faces in several situations. Person identification generalizes to other individuals that are known only in *one* situation [10, 11]. We then show ongoing work on details of a neural

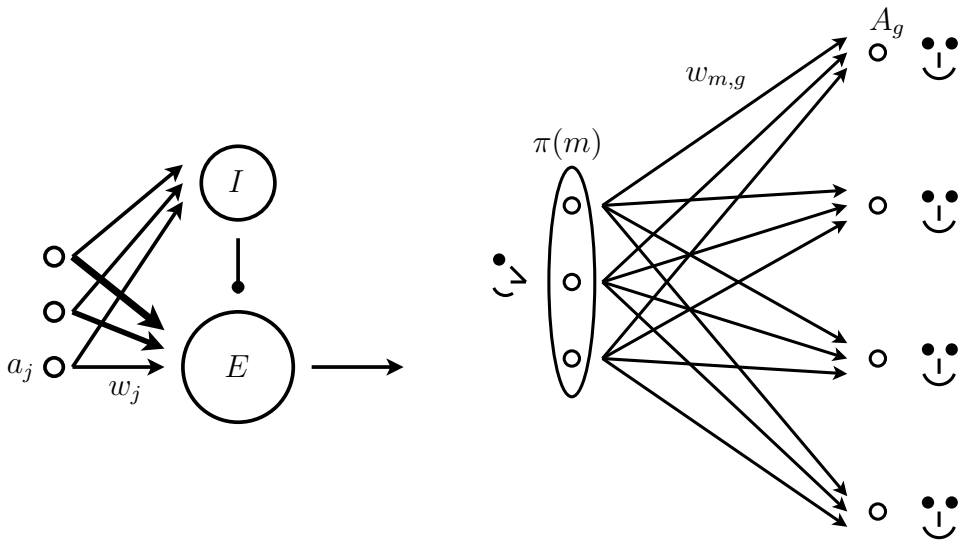


Figure 2: Left: A neural circuit sensitive to the order of firing neurons, the preferred order is stored in the weights w_j (after [12]). Right: The same circuit is repeated for each gallery image (more precisely for each feature in each gallery image). The probe image is represented as a rank list π according to similarities with model images in the same situation. The similarities of the gallery to the model images in neutral situation are coded in the weights $w_{m,g}$.

network based on spike timing, which can achieve invariant recognition in a very short time given parallel neuronal architecture.

2 Rank list recognition

The problem of learning the transformations between different situations can be solved by representing individuals in different situations by the rank list of their similarities to the model images in the same situation. Every probe image also creates a similarity rank list with the model images in its situation. See figure 1 for an illustration and [11] for full details.

3 Neural network for rank list comparison

[12] have proposed a neural network that can evaluate rank codes. A set of feature detectors responds to an input pattern such that the most similar detector fires first. The order in which the spikes arrive can then be decoded by a circuit depicted in the left half of figure 2.

In this paper we describe a spiking network implementation of the recognition procedure. The architecture is shown in figure 2 with the simplification that only one rank-list evaluating circuit per face is shown. We assume a neuronal module that calculates the similarity of stored model images to the actual probe image. Each feature of each gallery subject has one representing neuron. The similarity influences the time a neuron corresponding to this subject sends a spike. The higher the similarity the earlier the spike.

Following [12] rank lists are evaluated by a combination of an excitatory and inhibitory cell. In the excitatory cell weighted spikes are accumulated, in the inhibitory one unweighted spikes build up inhibition. In order to become active only for one desired order of arriving spikes the weights must be as follows (see [12, 11])

$$w_{m,g} = \frac{1}{N_M} \lambda^{\gamma_g(m)} \,. \tag{1}$$

The activity A_g then becomes

$$A_g = \sum_{m=1}^{N_M} \lambda^{\pi(m)} w_{m,g} \,, \tag{2}$$

$$= \frac{1}{N_M} \sum_{m=1}^{N_M} \lambda^{\pi(m)+\gamma_g(m)} \,, \tag{3}$$

which is a useful similarity function for the rank lists π and γ_g (see [11].

We have implemented the network in a continuous-time fashion, meaning that the precise spiking times are implemented as floats. This allows to study the robustness of the network under the influence of disturbances like imprecision in spike timing, synaptic delays, multiple spikes, etc...

After some global reset, each feature detector fires a spike at time:

$$t_i = 1 - S(J_i^M, J_i^G)\pi(m) + \gamma_g(m \tag{4}$$

Note that the similarities have values in $[0, 1]$ and these times as well. To map these values to biologically realistic timings requires a time unit of about 20ms.

4 Experiments and results

Like in [11], we have tested the system on the pose and illumination variations of the CAS-PEAL database [5, 4]. The landmarks are found by elastic bunch graph matching, starting from very few images, that were labeled by hand.

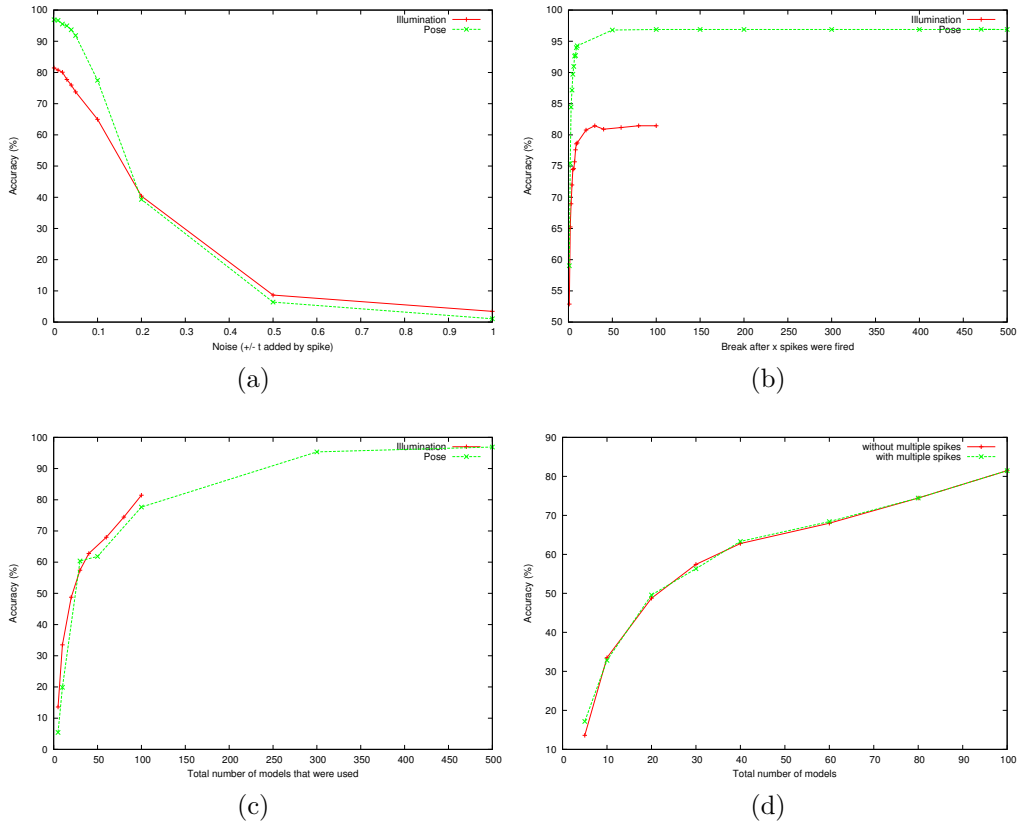


Figure 3: Results of experiments on the spike-based network.

24 subjects have been set aside for manual labeling. From these, the basic bunch graphs have been built (12 for pose, 8 for illumination). The remaining 1015 subjects have been partitioned into model sets and testing sets. In the standard partition for the pose case, the first 500 subjects were used for model and the following 515 for testing. In the illumination case the first 100 subjects were used for model and the following 91 for testing.

4.1 Random noise

First we have added random offsets equally distributed in $[-d, d]$ to the spike timings of (4) and measured the recognition rate. The results are shown in figure 3a.

$$t_i = 1 - S(J_i^M, J_i^G) + \chi(d) \tag{5}$$

Random spike timing errors are tolerated if the noise interval is around 0.05 time units.

4.2 Early stopping

Identity decisions can be made faster if the gallery neurons do not wait for all spikes to come in. As can be seen in figure 3b the first 20 spikes are enough to reach the full recognition performance, and already stopping after the first spike yields acceptable recognition rates. Note that the recognition rates for the methods tested in [4] on the same database are 71% for pose and 51% for illumination.

4.3 Dependence on size of model gallery

Model learning is only useful if the number of individuals in the model can be be much smaller than the number of people in the gallery. We have tested different model sizes with a fixed gallery size of 500 individuals for pose and 91 for illumination. The results are shown in figure 3d.

4.4 Multiple spikes

The assumption that an activated feature detector would fire only a single spike at a precise time is not in accordance with neurophysiology. The general view is that activation causes a spike train, with activity being coded in the *f*requency of spikes. In a second simulation the active neurons created a volley of spikes, which lasted for $T = 3$ time units.

$$t_i(n) = n \cdot (1 - S(J_i^M, J_i^G)) \, , n \in \left\{ 1, 2, \ldots \frac{T}{1 - S(J_i^M, J_i^G)} \right\} \tag{6}$$

Subsequent spikes might interfere with the evaluation of the rank lists, because they cannot be distinguished from first spikes.

5 Discussion

In this paper, we have presented some experiments showing the robustness of the network performance under errors in spike timing, shortness of models, and the presence of spike trains instead of single spikes. In ongoing work, we are varying the details of the spiking network in order to find promising parameter regimes for applying the network to larger problems.

Acknowledgements

We gratefully acknowledge funding from the German Research Foundation (WU 314/2-2 and WU 314/5-2). Portions of the research in this paper use the CAS-PEAL face database collected under the sponsorship of the Chinese National Hi-Tech Program and IS VISION Tech. Co. Ltd. [5, 4].

References

[1] Marian Stewart Bartlett and Terrence J. Sejnowski. Learning viewpoint-invariant face representations from visual experience in an attractor network. *Network – Computation in Neural Systems*, 9(3):399–417, 1998.

[2] P. Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.

[3] M. Franzius, N. Wilbert, and L. Wiskott. Invariant object recognition with slow feature analysis. In V. Kurkova, R. Neruda, and J. Koutnik, editors, *Artificial Neural Networks - ICANN 2008, pt. I*, volume 5163 of *LNCS*, pages 961–970, 2008.

[4] Wen Gao, Bo Cao, Shiguang Shan, Xilin Chen, Delong Zhou, Xiaohua Zhang, and Debin Zhao. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics Part A*, 38(1):149–161, 2008.

[5] Wen Gao, Bo Cao, Shiguang Shan, Delong Zhou, Xiaohua Zhang, and Debin Zhao. The CAS-PEAL large-scale Chinese face database and baseline evaluations. Technical Report JDL-TR-04-FR-001, Joint Research & Development Laboratory for Face Recognition, Chinese Academy of Sciences, 2004.

[6] G.E. Hinton. Learning translation invariant recognition in massively parallel networks. In G. Goos and J. Hartmanis, editors, *PARLE Parallel Architectures and Languages Europe*, number 258 in Lecture Notes in Computer Science, pages 1–13. Springer, 1987.

[7] Jenia Jitsev and Christoph von der Malsburg. Experience-driven formation of parts-based representations in a model of layered visual memory. *Frontiers in Computational Neuroscience*, 3(15):1–18, 2009.

[8] Martin Lades, Jan C. Vorbrüggen, Joachim Buhmann, Jörg Lange, Christoph von der Malsburg, Rolf P. Würtz, and Wolfgang Konen. Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42(3):300–311, 1993.

[9] Jörg Lücke, Christian Keck, and Christoph von der Malsburg. Rapid convergence to feature layer correspondences. *Neural Computation*, 20(10):2441–2463, 2008.

[10] Marco K. Müller, Alexander Heinrichs, Andreas H.J. Tewes, Achim Schäfer, and Rolf P. Würtz. Similarity rank correlation for face recognition under unenrolled pose. In Seong-Whan Lee and Stan Z. Li, editors, *Advances in Biometrics*, LNCS, pages 67–76. Springer, 2007.

[11] Marco K. Müller and Rolf P. Würtz. Learning from examples to generalize over pose and illumination. In Cesare Alippi, Marios Polycarpou, Christos Panayiotou, and Georgios Ellinas, editors, *Artificial Neural Networks – ICANN 2009*, volume 5769 of *LNCS*, pages 643–652. Springer, 2009.

[12] S. Thorpe, A. Delorme, and R. Van Rullen. Spike-based strategies for rapid processing. *Neural Networks*, 14(6-7):715–725, 2001.

[13] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):715 – 770, 2002.

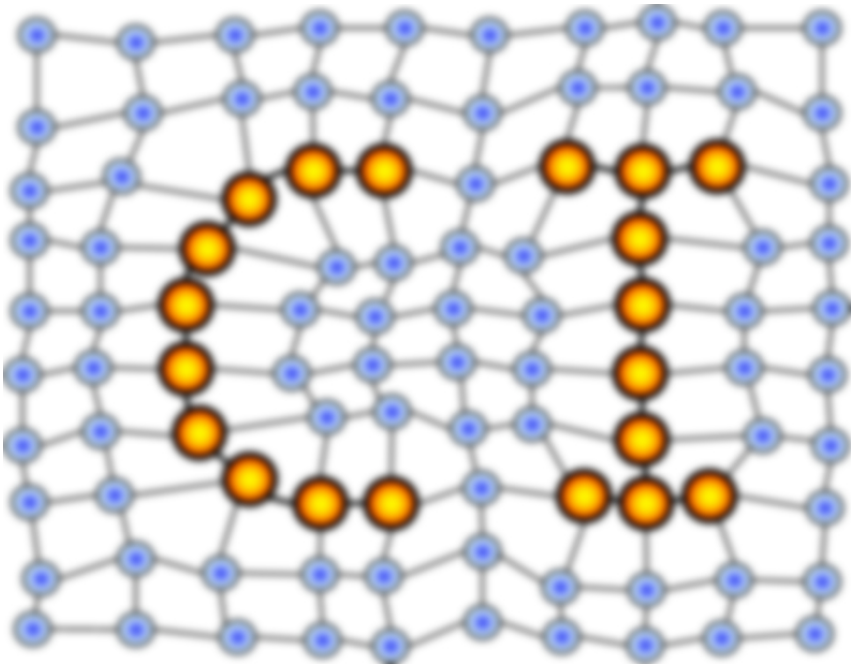
[14] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger, and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

[15] Laurenz Wiskott and Christoph von der Malsburg. Recognizing faces by dynamic link matching. *Neuroimage*, 4(3):S14–S18, 1996.

[16] Philipp Wolfrum, Christian Wolff, Jörg Lücke, and Christoph von der Malsburg. A recurrent dynamic model for correspondence-based face recognition. *Journal of Vision*, 8(7), 2008.

MACHINE LEARNING REPORTS

Report 05/2011



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.