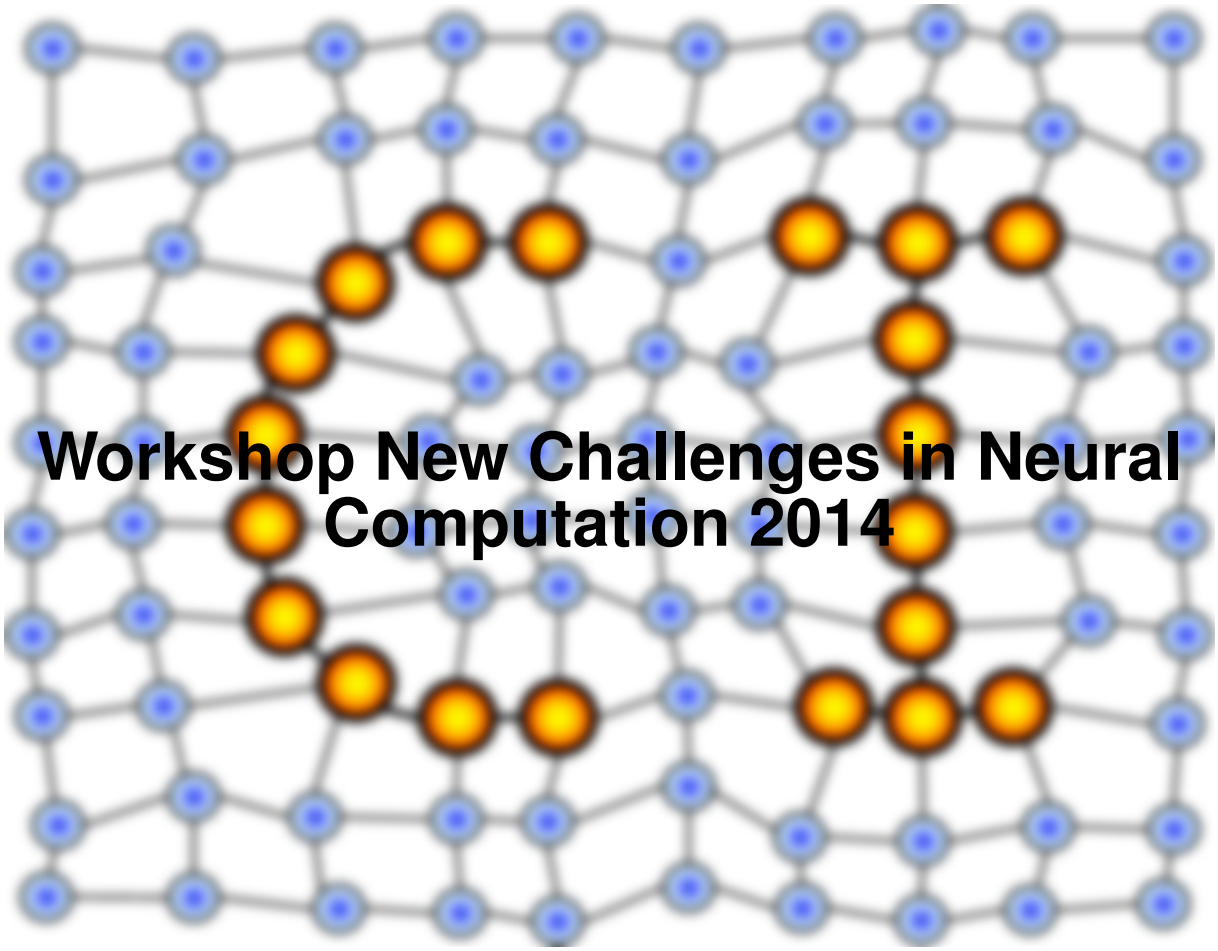
MACHINE LEARNING REPORTS



Workshop New Challenges in Neural Computation 2014

Report 02/2014
Submitted: 26.08.2014
Published: 02.09.2014

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

Table of contents

New Challenges in Neural Computation NC^2 – 2014

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

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

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

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

The workshop New Challenges in Neural Computation, NC^2, took place for the fifth time, accompanying the prestigious GCPR conference in the beautiful town of Münster, Germany, where many buildings still mirror its historic relevance: here, important events such as the Westphalian peace took place. The workshop itself centers around challenges and novel developments of neural systems and machine learning, covering recent research concerning theoretical issues as well as practical applications. This year, ten contributions from international participants have been accepted as short or long contributions, respectively, covering diverse areas connected to knowledge integration, application challenges, feature detection, or dynamics. In addition, we welcome two internationally renowned researchers as guest speakers, Prof. Dr. Peter Tiňo from Birmingham University, U.K., presenting 'Learning in the model space for temporal data' and Prof. Dr. Michael Biehl from Groningen University, The Netherlands, presenting a talk on 'Prototype-based classifiers and their application in the life sciences'. This invitation became possible due to the sponsoring of the European Neural Networks Society (ENNS) and the German Neural Network Society (GNNS). Within the workshop, a meeting of the GI Fachgruppe on Neural Networks took place.

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

Keynote talk: Learning in the Model Space for Temporal Data

Peter Tiňo, University of Birmingham, U.K.

Abstract:
I will first introduce the concept of learning in the model space. This talk will focus on time series data. After reviewing recent developments in model based time series kernels, I will introduce a framework for building new kernels based on temporal filters inspired by a class of "reservoir" models known as Echo State Networks. I will briefly outline the key theoretical concepts of their analysis and design. The methodology will be demonstrated in a series of sequence classification tasks and in an incremental fault detection setting.

Keynote talk: Prototype-based classifiers and their application in the life sciences

Michael Biehl, University of Groningen, The Netherlands

Abstract:
This talk reviews important aspects of prototype based systems in the context of supervised learning. Learning Vector Quantization (LVQ) serves as a particularly intuitive framework, in which to discuss the basic ideas of distance based classification. A key issue is that of choosing an appropriate distance or similarity measure for the task at hand. Different classes of distance measures, which can be incorporated into the LVQ framework, are introduced. The powerful framework of relevance learning will be discussed, in which parameterized distance measures are adapted together with the prototypes in the same training process. Recent developments and insights are summarized and example applications in the bio-medical domain are presented in order to illustrate the concepts.

Sparse Spectrum Hidden Markov Models of Metastable Systems

Hao Wu

Department of Mathematics and Computer Science, Free University of Berlin,
Arnimallee 6, 14195 Berlin, Germany
hwu@zedat.fu-berlin.de

Abstract. The spectral decomposition and estimation plays a very important role for analyzing and modeling metastable systems, because the dominant eigenvalues and eigenfunctions usually contain a lot of essential information of the metastable dynamics on slow timescales. The projected Markov model theory shows that a metastable system can be equivalently described as a small-sized hidden Markov model (HMM) under the assumption of strong metastability, therefore the dominant spectral components of a metastable system can be identified from simulation and experimental data by HMM learning. However, in the case that the number of dominant spectra is unknown, the choice of number of hidden states is still a challenge for traditional HMMs. In this paper, a sparse spectrum HMM is presented to address this problem

1 Background

For many physical and chemical process, the coarse-graining dynamics can be formulated by the following model:

$$x_{t+\tau}|x_t \sim p\left(x_{t+\tau} = x|x_t = x'\right) = \left(\mathcal{P}\left(\tau\right)\delta_{x'}\right)\left(x\right) \tag{1}$$

$$y_t|x_t \sim \Pr\left(y_t = k|x_t = x\right) = \chi_k\left(x\right) \tag{2}$$

where x_t and y_t represent the system state and observation at time t, the state process $\{x_t\}$ is assumed to be an ergodic and reversible Markov process driven by a Markov propagator $\mathcal{P}\left(\tau\right)$, the observation space $\mathcal{O} = \{1, \ldots, K\}$ is a finite set, and $\chi_k\left(x\right)$ is called the observation probability function for the observed value k. Generally speaking, the system observations $\{y_t\}$ are obtained from Galerkin discretization. In such cases, each k represents a finite element space in the state space and $\chi_k\left(x\right)$ is the corresponding characteristic basis function with $\chi_k\left(x\right) \in \{0, 1\}$. Obviously, the above model is in fact an HMM, but it is generally infeasible to reconstruct $\mathcal{P}(\tau)$ from $\{y_t\}$ by direct statistical inference because of the continuity of state space and the complexity of the dynamics of $\{x_t\}$. In [1], it was shown that if the state process $\{x_t\}$ has only m metastable states and satisfies some technical assumptions, the dynamics of $\{y_t\}$ can be described by an m-state HMM, and the dominant eigenvalues and projected eigenfunctions of $\mathcal{P}\left(\tau\right)$ can be extracted from transition probabilities and observation probabilities

of the m-state HMM. Thus, if a suitable m is given, we can efficiently perform the spectral identification through HMM learning. However, the choice of m is difficult in practical applications, and the numerical experiments in [1] show that the estimation results of the HMM method is very sensitive to the value of m.

2 Sparse spectrum HMMs

Here we develop an infinite-state Bayesian HMM called *sparse spectrum HMM* for spectral estimation, which is a modified version of the stick-breaking half-weighted model (SB-HWM) proposed in [2] and constructs prior distributions of the infinite-dimensional transition matrix $\mathbf{A} = [a_{ij}] = [\Pr(s_{t+\tau} = j | s_t = i)]$ and observation matrix $\mathbf{B} = [b_{ij}] = [\Pr(y_t = j | s_t = i)]$ as

$$\begin{aligned} G &= \sum_{k=1}^{\infty} w_k \delta_{\mathbf{b}_k} \sim \mathrm{DP}(\beta, D) \\ H' &= \sum_{i,j} h'_{ij} \delta_{\mathbf{b}_i \times \mathbf{b}_j} \sim \mathrm{DP}(\gamma, G \times G) \\ \mathbf{H} &= [h_{ij}] = [(h'_{ij} + h'_{ji})/2] \\ \mathbf{A} &= \mathbf{A}(\mathbf{H}) \end{aligned} \tag{3}$$

where s_t denotes the discrete state of the HMM at time t, $\mathbf{b}_i = (b_{i1}, \ldots, b_{iK})$ denotes the i-th row of $\mathbf{B}$, D represents the prior distribution of $\{\mathbf{b}_i\}$, $\mathrm{DP}(\alpha, G_0)$ denotes a Dirichlet process with concentration parameter α and base measure G_0, the symmetric matrix $\mathbf{H}$ is the half-weighted matrix of the HMM and the transition matrix $\mathbf{A}$ can be viewed as a function of $\mathbf{H}$ (see [2] for details). In contrast with the other infinite-state HMMs, including the SB-HWM, the sparse spectrum HMM has the following advantages:

1. The sparse spectrum HMM defines a sparse prior distribution of eigenvalues, i.e., in most cases, the transition matrix $\mathbf{A}$ generated by (3) has only a few eigenvalues which are significantly larger than zero. (Note that most infinite-state HMMs cannot guarantee the sparsity of the eigenvalue set, and tend to generate a lot of "pseudo-dominant eigenvalues". The detailed analysis is given in [2].)
2. This model involves only two Dirichlet processes, whereas most of the existing infinite-state HMMs contain infinite Dirichlet processes or stick-breaking processes.
3. The prior of this model is exchangeable, i.e., if we renumber the states and permute the indices of elements in $\mathbf{A}$ and $\mathbf{B}$ accordingly, the prior density value remains unchanged.
4. It is easy to construct a truncated approximation of (3) by replacing the prior of G with $G = \sum_{k=1}^{L} w_k \delta_{\mathbf{b}_k}$ and

$$(w_1, \ldots, w_L) \sim \mathrm{Dir}(\beta/L), \quad \mathbf{b}_k \overset{\text{iid}}{\sim} D \tag{4}$$

and the posterior distribution of the truncated model can be efficiently sampled by the Markov chain Monte Carlo approach. (The detailed sampling algorithm is omitted here due to the space limitation.)

3 Open problems and future work

3.1 Theoretical analysis of sparsity

In [2], it was proved that the i-th eigenvalue λ_i of an SB-HWM satisfies

$$\mathbb{E}\left[|\lambda_i|\right] = O\left(c^{\frac{i}{3}}\right) \tag{5}$$

where c is a constant belonging to $(0, 1)$. It is worth investigating if we can get the similar results on the sparse spectrum HMM in this paper.

3.2 Convergence of truncated models

Because we can only use the truncated approximation of sparse spectrum HMMs for Bayesian inference in practice, the following problem is very important for both applications and theoretical analysis: *Does the prior distribution of state sequence* $\{x_0, x_\tau, \ldots, x_{N\tau}\}$ *and spectral components of the truncated sparse spectrum HMM converge to that of infinite sparse spectrum HMM as the truncation length* $L \to \infty$*?* If the answer is yes, can we further get the convergence rate?

3.3 Combination with neural networks

In all the above, we assume that the observation variables y_t is a discrete variable, which is generally obtained by clustering observation data in practical applications. By combining multilayer perceptron (MLP) networks, we can extend the proposed method to continuous observations. Note that for the continuous observation y_t, the observation probability density of the HMM can be expressed as

$$p\left(y_t|x_t\right) = \frac{p(x_t|y_t)p(y_t)}{p(x_t)} \tag{6}$$

where $p(x_t)$ can be given by HMM, $p(y_t)$ is a constant which has nothing to do with the estimation procedure, and the conditional probability $p(x_t|y_t)$ can be estimated by an MLP network [3].

References

1. Noé, F., Wu, H., Prinz, J.H., Plattner, N.: Projected and hidden Markov models for calculating kinetics and metastable states of complex molecules. Journal of Chemical Physics **139**(18) (2013) 184114

2. Wu, H.: A Bayesian nonparametric model for spectral estimation of metastable systems. In: Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI). (2014) 878–887
3. Cohen, M., Cohen, I., Rumelhart, D., Morgan, N., Franco, H., Abrash, V., Konig, Y.: Combining neural networks and hidden markov models for continuous speech recognition. In: ICSLP. (1992) 915–918

The impact of frequency distributions in a perceptual grouping oscillator network

Martin Meier, Robert Haschke and Helge J. Ritter

Neuroinformatics Group
Bielefeld University, 33501 Bielefeld, Germany
{mmeier,rhaschke,helge}@techfak.uni-bielefeld.de

The Kuramoto Model [2] is a recurrent network composed of limit cycle oscillators, whose dynamic is designed to facilitate phase synchronization among populations of oscillators. A single oscillator O_n in a population of N oscillators is described by it's phase θ_n and frequency ω_n. The recurrent update equation of the model is

$$\dot{\theta}_m = \omega_m + \frac{K}{N} \sum_{n=1}^{N} \sin(\theta_n - \theta_m). \tag{1}$$

Hence, the oscillators are globally coupled with a coupling strength K. The frequencies ω_m are constant and drawn from a random distribution. These frequencies introduce a separating force into the network model, which drives oscillators away from each other. The $K/N \sum \sin(\theta_n - \theta_m)$ term counteracts this separation, forcing the oscillators into a phase-synchronized state. This model can be used to describe different kinds of natural phenomena, for example synchronous flashing of fireflies or the synchronization of pacemaker cells in the heart. Please see [4] for a review of synchronization effects in Kuramoto models.

In [3], a network based on a hierarchical model of coupled Kuramoto oscillators is introduced, which is able to solve a broad spectrum of perceptual grouping tasks, for example texture based image segmentation and contour integration. The key principle of this network is to represent features from an input space in a one-to-one relation by Kuramoto oscillators. The coupling strength between the oscillators is chosen based on the similarity of the features according to some distance metric.

In contrast to the original Kuramoto model (1), the hierarchical model includes individual coupling strengths f_{mn} between oscillators O_m and O_n to facilitate feature-dependent synchronization of oscillators. It is described by the recurrent update equation

$$\dot{\theta}_m = \omega_m + \frac{K}{N} \sum_{n=1}^{N} f_{mn} \sin(\theta_n - \theta_m). \tag{2}$$

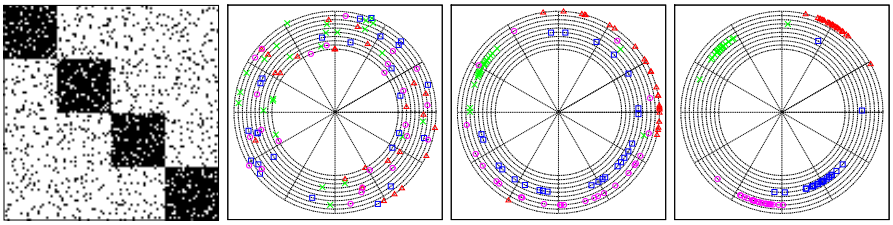
Using this equation, positively coupled oscillators, e.g. $f_{mn} = 1$, will attract each other and gather at a similar phase whilst negatively coupled oscillators ($f_{mn} = -1$) act repelling and spread in their phases.

In the original model, the frequencies ω_n are drawn from a random distribution and constant. We proposed to employ a set of discrete frequencies $\omega_0 \alpha$,

$\alpha = 1, \ldots, L$ and update the frequency of each oscillator based on the support they gain from all oscillators. The support is calculated based on cosine similarity of oscillator phases, limited to the range $[0, 1]$ and weighted by their local coupling i.e. $S_m(\alpha) = \sum_{n \in \mathcal{N}(\alpha)} f_{mn} \cdot \frac{1}{2} (\cos(\theta_n - \theta_m) + 1)$. The new frequency w_m is then chosen to maximize the support, thus boosting phase synchronization:

$$\omega_m = \omega_0 \cdot \underset{\alpha}{\operatorname{argmax}} \Big(S_m(\alpha) \Big) \tag{3}$$

This adaptation allows an easy assignment of grouping results: Oscillators sharing a common frequency index α represent the same group. Additionally, adapting oscillators to the same frequency reduces the phase spread compared to randomly drawn, constant frequencies.



(a) f_{mn} (b) Initialization. (c) 10 updates. (d) 20 updates.

Fig. 1: This figure displays an example of the dynamics for a network with 100 features, 10 discrete frequencies and 4 target groups. The leftmost panel shows the coupling matrix f_{mn} between features, where a black pixel indicates an attracting coupling while white represents a repelling coupling. Fig. 1b to 1d visualize the state of the oscillators in phase space (polar coordinates) at initialization, after 10 and 20 updates. The color and symbol of the oscillators represents the desired target state.

The behavior of the network in an artificial grouping task is shown in Fig. 1. The coupling matrix f_{mn} for 100 features, divided into four groups with 10% noise is shown in Fig. 1a, whilst the state of the oscillators in phase space is shown in Fig. 1b–1d. After 10 updates the network achieves a perfect grouping result in terms of frequency assignment. After 20 updates, oscillators representing the same target group have a high phase synchrony as well.

Although the recurrent update (2) appears intriguingly simple, the model and its' variations opened a wide spectrum of research, e.g. see [1] for an overview. By introducing a frequency adaption (3) in conjunction with discrete frequencies and recalling the original update equation (1), where the frequencies ω_n induce phase spread while the sin() term drives the oscillators towards a mean phase, the question arises whether the distribution of these frequencies has an impact on the network dynamics. To get a first insight into this question, we employed genetic algorithms (GA) to generate different sets of discrete frequencies and analyzed if the GA was able to improve the target state of the network by varying the discrete frequencies.

The initial grouping problem was similar to the settings in [3]. The compatibility f_{mn} was expressed as a matrix which encodes the couplings among 100 features, split into four target groups of 25 features. The matrices contained 25% noise. The oscillator networks contained 10 discrete frequencies. The crossover probability of the GA was set to 50% with a mutation probability of 3%. The population consisted of 100 chromosomes, where each chromosome encoded the discrete frequencies of an oscillator network. The fitness function was designed in terms of the grouping quality $q = [0, 1]$ and the oscillator order $r = [0, 1]$. A grouping quality q of one represents a perfect result compared to a given target labeling, whilst the order r represents the phase coherence of the oscillators. A value of one means that all oscillators share the same phase θ. For a more comprehensive description of both evaluation measures please refer to [3]. For the initial trials, two fitness functions are evaluated. On the one hand the product of quality and order $q * r$ and on the other hand the average of quality and order $\frac{q+r}{2}$. Additionally, two different ranges of frequencies are used, $[\frac{\pi}{2}, 2\pi]$ and $[\frac{\pi}{2}, 20\pi]$. The results are shown in table 1 for the different conditions. The last column shows the product of quality and order without a GA optimization, averaged over 1000 trials.

cond.	avg, 2π	avg, 20π	mul, 2π	mul, 20π	no GA
μ and σ	0.986 ± 0.001	0.989 ± 0.01	0.971 ± 0.001	0.979 ± 0.001	0.963 ± 0.03

Table 1: Mean and standard deviation of the fitness function over 1000 evolution steps for each condition. The leftmost column shows the result over 1000 trials for a non-optimized oscillator network.

These results suggest, that the frequency distribution does not have a significant impact on the grouping behavior of the oscillator network, at least for the considered artificial grouping problem. In contrast to the hierarchical model (2), the frequency adaption (3) reduces the phase spread induced by randomly drawn, fixed ω values in oscillator groups, which could counteract possible impacts of different ω distributions on the dynamics. Extending this investigation towards a mean phase analysis of oscillator groups will be of future interest.

References

1. Acebrón, J.A., Bonilla, L.L., Vicente, C.J.P., Ritort, F., Spigler, R.: The Kuramoto model: A simple paradigm for synchronization phenomena. Reviews of modern physics 77(1), 137 (2005)
2. Kuramoto, Y.: Chemical oscillations, waves, and turbulence. Courier Dover Publications (2003)
3. Meier, M., Haschke, R., Ritter, H.J.: Perceptual grouping through competition in coupled oscillator networks. Neurocomputing 141, 76–83 (2014)
4. Strogatz, S.H.: From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators. Phys D 143(1), 1–20 (2000)

Spiking network simulations

Tim U. Krause, Phil Y. Schrör, and Rolf P. Würtz

Institut für Neuroinformatik, Ruhr-Universität, Bochum, Germany
e-mail: utz.krause@web.de, phil.schroer@rub.de, rolf.wuertz@ini.rub.de

Action potentials, also called spikes, are a very widespread, though not universal, communication mechanism between neurons. Their biophysics are well understood and extensively modeled [1, 2].

It is less well known what the precise code of these spike trains is. Classically, it is assumed that the *frequency* of spikes codes for the activation of the neuron. The evidence for that is strong at the sensory and motor end of the nervous system. It takes, however, relatively long integration times (at least 3 spikes) to measure such a frequency for further processing. Therefore, a system using this code throughout would be rather slow. Also each spike costs energy, and it would be rather inefficient to require many spikes for some bits of information.

On the other hand, the timing of spikes carries much more information than the frequency, to a theoretical limit of a real number for each spike. That precision is, of course, also limited by the noise on the timing. There is a continuum of possible codes from pure frequency coding to relevant information carried by a single spike time.

To speed up processing, it has been suggested by Thorpe [3] that the order of arrival times of spikes at a neuron can distinguish between $w!$ cases, with w the number of incoming synapses. We have exploited that idea for learning of (arbitrary) invariances by rank-order coding [4, 5]. We could also show that timing noise as well as interfering spikes from bursting can be tolerated to a certain extent without breaking the performance of the network.

Furthermore, it has been observed in real neurons that the precise timing of spike arrival can be an important variable for plasticity or weight learning. This phenomenon is called spike-time dependent plasticity (STDP) [6, 7].

Beyond biological relevance the question arises what technical problems can be solved by computation based on spike-times. We have started to explore that question experimentally by computer modelling. In the current work we will completely abstract from spike shape and propagation dynamics and describe each spike by a single floating point number, which is its creation time.

For that view, a neural network consists of a directed graph with neurons as nodes and connections as edges. Edges have two scalar properties. The first is the classical *weight*, which specifies how much the potential of a postsynaptic neuron is changed by one incoming spike. The second is the time one spike needs to travel from the creating neuron to the postsynaptic one, which we will call *delay*.

Neurons accumulate incoming weighted spikes in a local *potential* and create a new spike once a threshold has been passed. The second relevant parameter for

neurons is the *decay rate* of the potential, the third a possible *refractory period*, during which incoming spikes have no effect.

The evolution of the network is calculated by adding delays to creation times, updating neuron potentials, and recording new spikes. We have built a simulator [8] to model such a network efficiently by simple bookkeeping of the times new spikes are created in the network. The implementation is able to simulate large numbers of spikes, such that comparisons between rate coding and temporal coding can be made.

We present first results on small networks like a frequency bandpass filter, a coincidence detector, and a fully connected network. We also compared STDP to rate-based Hebbian learning in a feedforward network in a supervised training mode.

We have made some experiments on image segmentation [9], which must be fast to account for the speed of perception.

Future applications will include more learning experiments. A particularly interesting question is how delays can be learned (by, e.g., by myelinization), what the time constants and implications for information processing are.

References

1. A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology-London*, 117(4):500–544, 1952.
2. Romain Brette, Michelle Rudolph, Ted Carnevale, Michael Hines, David Beeman, James M Bower, Markus Diesmann, Abigail Morrison, Philip H Goodman, Frederick C Harris Jr, et al. Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of Computational Neuroscience*, 23(3):349–398, 2007.
3. S. Thorpe, A. Delorme, and R. Van Rullen. Spike-based strategies for rapid processing. *Neural Networks*, 14(6-7):715–725, 2001.
4. Marco K. Müller, Michael Tremer, Christian Bodenstein, and Rolf P. Würtz. Learning invariant face recognition from examples. *Neural Networks*, 41:137–146, 2013.
5. Marco K. Müller, Michael Tremer, Christian Bodenstein, and Rolf P. Würtz. A spiking neural network for situation-independent face recognition. In Barbara Hammer and Thomas Villmann, editors, *Proceedings of New Challenges in Neural Computation, Frankfurt, August 2011*, number 5/2011 in Machine Learning Reports, pages 62–69, 2011.
6. Sen Song, Kenneth D Miller, and Larry F Abbott. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9):919–926, 2000.
7. Guo-qiang Bi and Mu-ming Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 24(1):139–166, 2001.
8. Tim Utz Krause. Rate coding and temporal coding in a neural network. M.Sc. thesis, Electrical Engineering, Univ. of Bochum, Germany, January 2014.
9. Phil Yannik Schrör. Analyzing spike-time synchrony for image processing. B.Sc. thesis, Applied Informatics, Univ. of Bochum, Germany, July 2014.

Role of competition in robustness under loss of information in feature detectors

Arash Kermani Kolankeh[1] Michael Teichmann and Fred Hamker

Chemnitz University of Technology, Department of Computer Science, Chemnitz, Germany,
arash.kermani-kolankeh@informatik.tu-chemnitz.de

Abstract. In this work the robustness under loss of information is considered as a new criterion for effectiveness of feature detectors. Four feature detectors with different levels of competition among their units are evaluated and it is observed that robustness under loss of information is directly related to the quality of competition in the feature detector.

Keywords: competition, neural networks, Hebbian learning, lateral inhibition, independent component analysis, non-negative matrix factorization, predictive coding/biased competition, occlusion, information loss

1 Introduction

There exist some criteria like sparseness, independence and visual appearance of components which have been widely used to measure the effectiveness of feature detectors. In this work we have introduced a new criterion based on the robustness of systems under loss of information. This is important as it measures the effectiveness of solution in a real task and is not based just on theoretical concepts.

2 Methods

In order to investigate the role of competition in robustness against loss of information we compared Fast Independent Component Analysis (FastICA) [1], Non-Negative Matrix Factorization with Sparseness Constrains (NMFSC) [2], Predictive Coding/Biased Competition (PC/BC) [3] and a Hebbian neural network with anti-Hebbian lateral connections [4] on occluded, whitened and resized MNIST handwritten digits. The occlusion was made by randomly putting 5 to 60 percent of the pixels to zero (Figure 1) which simulates loss of information. At first, each of the feature detectors was trained on the train set of the MNIST dataset and (after convergence) the activities were captured. Then the occluded data was fed to each of the feature extractors and the same as for the train set, the activities were saved. Then the train and test phase activities were given to Linear Discriminant Analysis (LDA) classifier to observe how much the accuracy of the classification drops by increasing the occlusion (loss of information). A

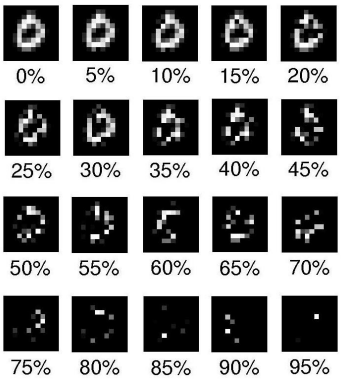


Figure 1. Digit 0 as an example of 0% to 95% occlusion of the input digit patches. Zero to 60% occlusion was used for the current study.

simple LDA classifier was chosen, as a sophisticated classifier would compensate the loss of information itself. A higher accuracy of classification under occlusion would mean that the feature detector has been able to correctly "guess" what the occluded input has been and compensate the loss of information.

3 Results

We observed (Figure 2) that FastICA shows the lowest robustness, NMFSC takes the second place, Hebbian neural network comes after and the highest robustness is shown by PC/BC. This is consistent with the fact that there is no competition in FastICA. In NMFSC while giving the opportunity of learning novel features to units by implicitly inhibiting the input to well-tuned components [5], it does not let the tuned components compete with each-other. In other words, although competition exists implicitly, it doesn't strongly occur among well-tuned components and they may share some redundant information. In the Hebbian network although, competition occurs among the components by suppressing activities which frequently happen simultaneously. This kind of local competition among cells prevents them from learning redundant information. This helps the units to be well-tuned to very important characteristics of the data and minimally confuse different structures if some parts of the input are missing. Finally, PC/BC as a generative model benefits from feedback error detecting weights which divisively inhibit the input. Like Non-Negative Matrix Factorization, this method tries to minimize the error between the input and its reconstruction, although some additional normalizations in feed-forward weights and in feed-back weights (which are proportional to the feed-forward weights) make competition possible also among well-tuned components. This generative method in fact benefits from a globally guided competition which is superior to the Hebbian neural network which uses locally available information to each cell for competition.

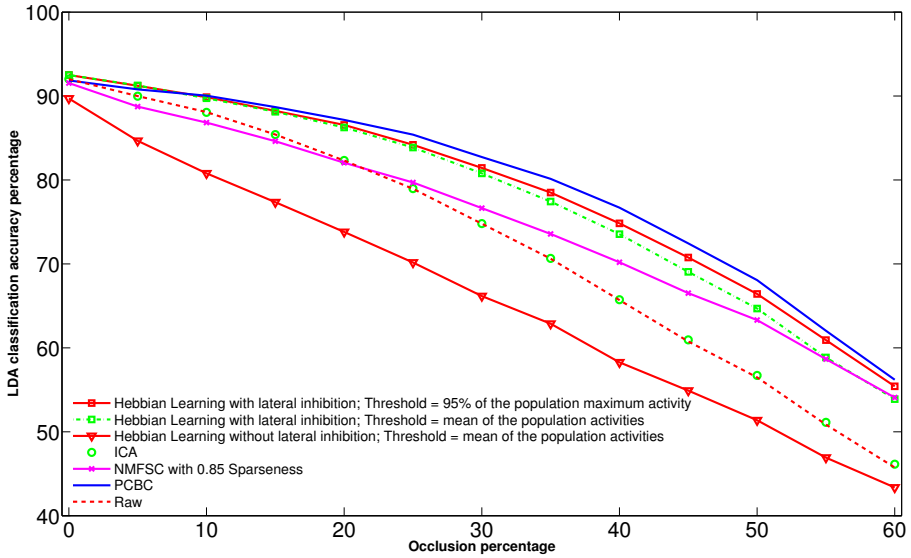


Figure 2. Classification performance on the output of FastICA, NMFSC, Hebbian Neural Network, and PC/BC as well as on the raw data. The robustness of classification under occlusion is in direct relation to the effectiveness of competition among units of feature detectors.

We have concluded that the more effectively components compete with each other during and after learning, the more robust is the system against loss of information. Although the Hebbian neural network is an attempt to simulate the function of primary visual cortex in a biologically plausible way and it could rely only on local information, we used the new idea to improve its efficiency. In the original network [4] the border between increasing and decreasing phases of feed-forward weights was the population mean. Increasing this threshold to 95% of the maximum activity which implies a kind of winner take all mechanism increased the robustness of the neural network. This also supports the idea that even within a single system, increasing the competition among units improves robustness against loss of information. On the other hand turning the competition off (after learning) resulted to the worst robustness (Figure 2). This tells us that even if the components are correctly learned, lack of competition will result in confusion if a small amount of data is lost.

References

1. Hyvärinen, A. and Oja, E. (1997), A Fast Fixed-Point Algorithm for Independent Component Analysis, *Neural Comput. Neural Computation* , 9:1483–1492,
2. Hoyer P.O., E. (2004). Non-negative Matrix Factorization with Sparseness Constraints. *Journal of Machine Learning Research*, 5:1457–1469.
3. Spratling M. W. (2012). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. *Neural Computation*, 24(1), 60–103.
4. Teichmann, M., Wiltschut, J., Hamker, F.H. (2012). Learning invariance from natural images inspired by observations in the primary visual cortex. *Neural Computation.*, 24(5):1271–1296.
5. Spratling M. W., De Meyer K. and Kompass R.(2009). Unsupervised Learning of Overlapping Image Components Using Divisive Input Modulation. *Computational Intelligence and Neuroscience*, Volume 2009, Article ID 381457.

Learning Transformation Invariance for Object Recognition

Jens Hocke, Thomas Martinetz

Institute for Neuro- and Bioinformatics, University of Lübeck

Abstract. Based on Tomaso Poggio's M-theory, we propose a method to learn transformation invariant representations. Using an artificial dataset, we demonstrate that our supervised method learns invariance to shifts, and on the MNIST data we show first results for learning the unknown transformations underlying handwritten digits.

1 Introduction

Visual object recognition is a challenging task in computer vision. Even small changes to an object's pose can yield dramatic changes to the 2D image in its pixel representation. Therefore, a representation invariant to such changes is mandatory for achieving good recognition rates. Modern approaches to that problem are scale-invariant feature transform (SIFT) [1] for coping with scale invariance and convolutional neural networks [2, 3] for coping with shift invariance.

Recently, the M-theory [4] was proposed explaining how invariance could be implemented in the ventral stream. Besides the theoretical insights on invariance, also a simple algorithm based on this theory is presented to find transformation invariant representations. However, there are two limitations. First, the theory explains only in-plane transformations, and, second, in the algorithm presented, the transformations are assumed to be known in advance. Addressing the later drawback we present a method based on the M-theory to learn invariance to unknown transformations. This enables us to gain (approximate) invariance to complex and unknown transformations.

After introducing the core ideas of the M-theory and describing our approach we demonstrate its potential in an artificial setting and on handwritten digits, assuming these digits undergo complex transformations when written by different people.

2 M-theory

According to the M-theory [4], invariance to a group G of transformations can be achieved in a representation using orbits O. This is the core idea of the M-theory, which we used for our method. In the following we will describe this concept, and refer the reader to [4] for a more exhaustive description of the theory. Here,

we use $g \in G$ to denote the group elements, and by $g(\boldsymbol{x})$ we denote the group's action applied to the image $\boldsymbol{x} \in \mathbb{R}^D$. By applying all transformations $g_i \in G$ to some image $\boldsymbol{x}$ an orbit $O_{\boldsymbol{x}} = \{g_i(\boldsymbol{x}) | g_i \in G\}$ is induced. This orbit is unique for the object in $\boldsymbol{x}$, and it is invariant to the transformations in G. For example the group of in-plane rotations would induce an orbit containing all possible rotated versions of the original image $\boldsymbol{x}_1$. The orbit for some other image $\boldsymbol{x}_2 = g_i(\boldsymbol{x}_1)$ that can be obtained from $\boldsymbol{x}_1$ by rotation would be the same, because for both $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ all possible rotated versions are contained in the orbit. Of course for some different image $\boldsymbol{x}_3$ that can not be obtained from $\boldsymbol{x}_1$ by rotation the orbit would be different.

For object recognition we would need to generate and compare the orbit of an unknown object to the stored orbit of a known object. It is not clear how to measure the similarity of two obits. One possibility is to use the probability distribution $P_{\boldsymbol{x}}$ induced by the transformations g_i on the image. For these distribution the following holds:

$$\boldsymbol{x}_1 \sim \boldsymbol{x}_2 \iff O_{\boldsymbol{x}_1} = O_{\boldsymbol{x}_2} \iff P_{\boldsymbol{x}_1} = P_{\boldsymbol{x}_2}. \tag{1}$$

However, these probability distributions are extremely high dimensional making it impractical to obtain them. Therefore, we would like to embed the invariance and discrimination properties of the distributions to a space of lower dimension. The Cramér-Wold theorem [5, 4] ensures that these high dimensional probability distributions can be described by D distributions $P_{\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle}$ over one dimensional projections $\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle$, where $\boldsymbol{p}_n, n = 1, \dots D$ are the projection vectors. To discriminate a finite number of distributions, empirically a small number of projections $N < D$ is sufficient [4].

Instead of transforming the input image $\boldsymbol{x}$, we can also apply the inverse transformation to the projection vectors $\boldsymbol{p}_n$:

$$\langle g_i(\boldsymbol{x}), \boldsymbol{p}_n \rangle = \langle \boldsymbol{x}, g_i^{-1}(\boldsymbol{p}_n) \rangle. \tag{2}$$

By applying the transformations to the templates, we avoid transforming every new image. This allows an invariant and discriminative representation in a simple two layer neural network with the transformations stored in the synapses. The first layer generates all the outputs using scalar products of all weight vectors $\boldsymbol{w}_{in} = g_i^{-1}(\boldsymbol{p}_n)$ with the input $\boldsymbol{x}$, and the second layer quantifies the distributions over the outputs of the first layer.

The restriction to groups of transformations allows only few transformations like periodic boundary shifts and in-plane rotations. Other common transformations such as shifts and scaling may not be fully observed by projection vectors of finite length. However, invariance to these partially observable groups can be achieved for a range of parameters and for non-group transformations approximate invariance can be achieved.

3 Invariance Learning

In the original M-theory, the weight vectors $\boldsymbol{w}_{in}$ are derived from the given transformation, e.g., translation or rotation. In our approach we want to learn

these weights to be able to adapt to unknown transformations. We quantify the distributions $P_{\langle g_i(\boldsymbol{x}),\boldsymbol{p}_n\rangle}$ by moments m. So every input image $\boldsymbol{x}$ is characterized by

$$y_{nm}(\boldsymbol{x}) = \sum_{i}^{I} \left(\boldsymbol{w}_{in}^{\top}\boldsymbol{x}\right)^{m}, \tag{3}$$

which is invariant to the transformations $g_i \in G$. In order to obtain a unique and discriminate set of outputs y_{nm}, the number N of projections, the number I of weight vectors per projection and the set of moments need to be set appropriately.

For our supervised approach a set of labeled training images is needed, and there should be multiple images per class available. For every class $c \in C$, moment $m \in M$, and projection $\boldsymbol{p}_n, n = 1, \ldots N$ a target value t_{cmn} is introduced. These target values are used to learn the unknown outputs y_{nm} for every class, with equal outputs for intraclass tuples and different outputs for interclass tuples. The following energy term enforces the moments of the projections to match their target

$$E_{\mathcal{S}} = \sum_{k}\sum_{m}\left(t_{cmn} - \sum_{i}\left(\boldsymbol{w}_{in}^{\top}\boldsymbol{x}_k\right)^{m}\right)^{2}. \tag{4}$$

By minimizing this term invariance to transformations in the training set is obtained, because the distributions for intraclass tuples are matched. However, this term will not guarantee a discriminative result. Therefore, a second energy term is introduced to enforce a minimum distance between the target vectors of every possible tuple of different classes c and c'

$$E_{\mathcal{D}} = \sum_{c,c'} \max\left(1 - ||\boldsymbol{t}_c - \boldsymbol{t}_{c'}||, 0\right)^{2}, \tag{5}$$

with the target vectors $\boldsymbol{t}_c = \left(t_{c,1,1}, t_{c,1,2}, \ldots, t_{|C|,|M|,N}\right)^{\top}$. The energies $E_{\mathcal{S}}$ and $E_{\mathcal{D}}$ are combined using the weighting factor α

$$E = \alpha E_{\mathcal{S}} + (1-\alpha)E_{\mathcal{D}}. \tag{6}$$

Using this energy term (6) targets t_{cmn} and the weight vectors $\boldsymbol{w}_{in}$ can be learned by gradient optimization, after the targets t_{cmn} and the weight vectors $\boldsymbol{w}_{in}$ have been initialized randomly. In our experience stochastic gradient descent was too slow, and, therefore, we used the Sum of Functions optimizer [6], which in addition to the speed also needs no learning rates to be set.

4 Distance to Center Classification

In case we were able to learn full invariance to a transformation, all images of one class $c*$ will lie exactly on the corresponding target vector $\boldsymbol{t}_{c*}$. If only approximate invariance was achieved, all these images are clustered around $\boldsymbol{t}_{c*}$.

Therefore, the closest target vector determines the class label $c*$ for some image $\boldsymbol{x}$:

$$c* = \arg\min_c ||\boldsymbol{y}(\boldsymbol{x}) - \boldsymbol{t}_c||, \tag{7}$$

with $\boldsymbol{y} = \left(y_{,1,1}, y_{1,2}, \ldots, y_{|M|,N}\right)^\top$.

5 Experiments

We show first experimental results. Many of the parameters are not optimized, yet. For all the presented results only the second moment was used to quantify the distributions. By setting the weighting parameter α to 0.01, the interclass term was emphasized, which according to our experience leads to faster convergence. The number of projections and the number of weight vectors per projection vary for the experiments, and are described for each experiment separately.

As a proof of concept we used shifted binary patches of size 4×4. 100 patches were generated randomly by setting each pixel to either one or to zero with probability 0.5. Then every patch was shifted using periodic boundary conditions (see Figure 1). On the resulting 1600 training samples, we trained two projections

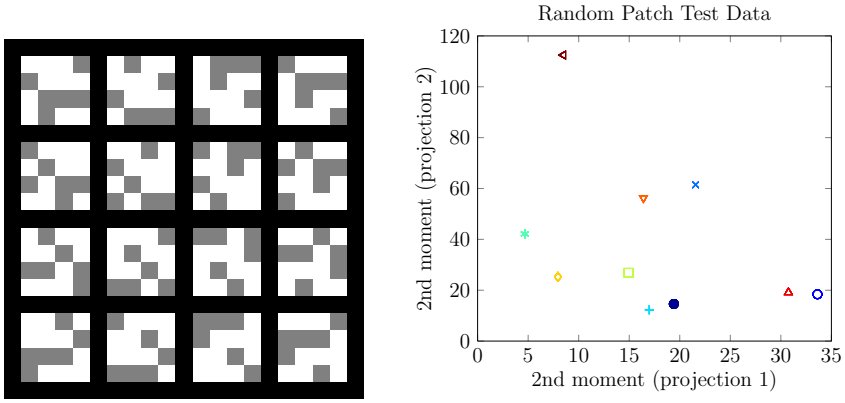


Fig. 1. The left image shows a random example patch in all its 16 possible shifts. The plot on the right shows the two second moments we obtain from projecting the orbits of the test data, i.e., $y_{1,2}$ and $y_{2,2}$ from Equation (1). Each patch is denoted by a different shape and color. All the shifted versions of a patch indeed fall on the same point, demonstrating perfect shift invariance of this representation. The 10 different test patches now can easily be discriminated.

with 16 weight vectors each. We used 16 weight vectors per projection, because we know there are 16 possible shifts. Like the training samples, 160 test samples were obtained from ten random patches by shifting. The orbits of the test samples were then projected with the learned weights using Equation (1). Since we only

use the second moment, the two projections provide two values for each input image $\boldsymbol{x}$. In Figure 1 we see that the representation is perfectly invariant to the learned transform, because all the shifted versions of a patch fall on a single point.

Going one step further, we tested our method on handwritten digits from the MNIST [3] dataset. It contains 60.000 training and 10.000 test samples. Here, we assume that every sample of a certain digit is a transformed version of a prototype digit. From the training data we learn invariance to the unknown transforms underlying MNIST, which is a much larger challenge than learning the known shifting transform in the experiment above. Since the transforms are unknown, we do not know how to select the number of weight vectors per projection, and in addition the images are of size 28×28, therefore many more parameters need to be learned.

For the visualization shown in Figure 2, we trained 2 projections with 20 weight vectors each and again take the second moments. The test data are nicely clustered into the ten digits. Since not all equally labeled digits are perfectly aligned, only an approximately invariant representation was found. However,

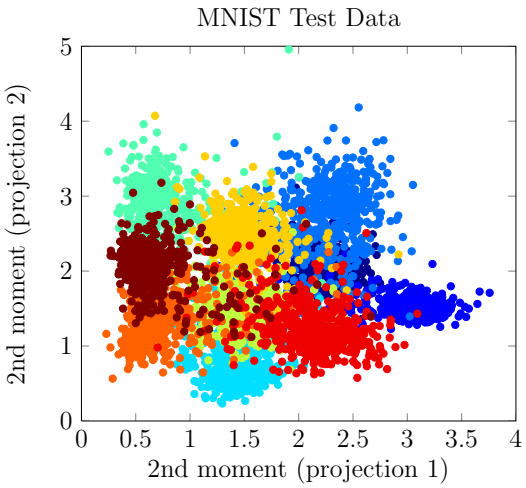


Fig. 2. This plot shows the two second moments we obtain from projecting the orbits of the MNIST test data using our method. Clearly, for every digit the samples form a cluster.

if these two projections we chose for visualization are not enough for perfect separation, we can increase the number of projections. If we use 10 projections, the distance to center classification described in Section 4 achieves 2.86% error rate on the test data significantly improving the 16.63% error rate obtained for the two projection setting. If the distance to center classification is applied in

the input pixel space of 768 dimensions, 17.97% of the samples are not classified correctly. This shows how well our method organizes the space.

6 Conclusion

Based on the M-theory, we introduced a supervised learning method to find an invariant representation. In the experiments we showed that our method can learn perfect invariance to periodic boundary shifts. For the much more complex, unknown transformations in MNIST full invariance was not achieved. However, the data was clustered good enough for a decent classification performance. We hope to improve these promising results by a better understanding of the different parameters.

References

1. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2. ICCV '99, Washington, DC, USA, IEEE Computer Society (1999) 1150–1157
2. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics **36** (1980) 193–202
3. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11) (1998) 2278–2324
4. Anselmi, F., Leibo, J.Z., Rosasco, L., Mutch, J., Tacchetti, A., Poggio, T.: Unsupervised learning of invariant representations in hierarchical architectures. CoRR **abs/1311.4158** (2013)
5. Cramér, H., Wold, H.: Some theorems on distribution functions. Journal of the London Mathematical Society **s1-11**(4) (1936) 290–294
6. Sohl-Dickstein, J., Poole, B., Ganguli, S.: An adaptive low dimensional quasi-newton sum of functions optimizer. CoRR **abs/1311.2115** (2013)

Case Study: Behavioral Prediction of Future Revenues in Freemium Games

João Alves[1], Sascha Lange[1], Michael Lenz[2], and Martin Riedmiller[3]

[1] 5d lab GmbH,
Freiburg, Germany
{joao,sascha}@5dlab.com
[2] InnoGames GmbH,
Hamburg, Germany
michael.lenz@innogames.com
[3] University of Freiburg, Department of Computer Science,
Freiburg, Germany
riedmiller@informatik.uni-freiburg.de

Abstract. In this paper we highlight Freemium games as an attractive domain of study for machine learning research, not only because of the huge amount of data that is readily available but also because of the benefits that even small improvements can bring. We discuss several options to apply machine learning methods to this domain's data, making use of the inherent information and learning business relevant predictions and classifications. Using behavioral information about the actions of 192 000 players inside the game, we demonstrate that it's possible to beat the industry gold standard solution for predicting future revenues. We also succeed in narrowing down large user groups to a small subgroup that is very likely to generate large parts of the future revenue. We conclude our case study with an outlook on how such a machine learning approach could become a module in a larger predictive analytics solution automatically evaluating and optimizing a company's marketing spendings.

Key words: Data Mining, Multilayer Perceptron, Random Forest, Freemium.

1 Introduction

The Freemium model is a pricing strategy in which a game product or other service is provided for free, however the usage of some proprietary features, functionalities or 'virtual goods' are provided at a cost. Because revenues in the Freemium model aren't realized *before* the usage of the product, as with a boxed title, but are usually *delayed* until after an initial free usage period of days or even weeks, developers are very interested in optimizing the user experience and satisfying their customers, thus making them stay longer in the game. Furthermore, it's in the very best interest of a developer to remove all obstacles that hinder the progress of users in a so-called 'conversion funnel' [21] to finally become (returning) customers. As a result, improvement and optimization

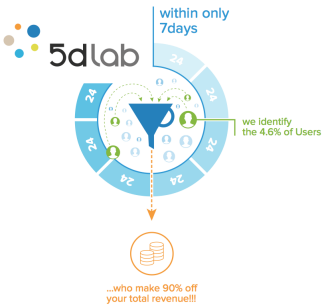


Fig. 1. Infographic with one main result of this case-study.

of every aspect of the product is a continuous and ongoing effort over the whole lifetime of the product or service.

Because of this strong need to understand how users use and interact with their system, developers usually collect huge amounts of data (truly 'Big Data'). Data typically includes session events, payment events, feature usage, game progression, and social interactions, often going as far as recording every single click of every single user on every single button within the game or service. This data is used to analyze and optimize the game in every aspect, feature by feature. A huge treasure for applying machine learning methods.

Within this article, we report on an idea to mine this data, namely the prediction of future long-term revenues from users' early behavior within a game. For this case study, we received more than 38 million player events recorded over the first seven days of play for about 192 000 new users in an international block buster game. From these events we generated high-dimensional user profiles and then tested several models on predicting the total spendings of each particular user within the following half year of playing the game. While we trained and compared different deep and shallow models, with and without feature selection, we achieved the very best results with a composite neural model.

This prediction could help developers improve a central aspect of their business model: evaluating and optimizing marketing campaigns. Profit is generated, if average lifetime spendings are higher than the average acquisition costs. Developers can't wait long to measure the value of each campaign and as such they want to have early indicators for the quality of an individual campaign to support their decision whether to stop or enlarge it. But since the primary metric of concern, the long-term revenue, can't be measured directly after only a few days, most developers use ad hoc secondary metrics (e.g. 2nd-day-return) to evaluate campaigns that do or do not correlate with the revenues. The few developers that try to rely on the primary metric through prediction, use a simple extrapolation of early spendings that was published by Faber et al [8]. In section 2, we demonstrate that including behavioral data and applying machine learning methods can actually beat this de-facto industry gold standard.

After elaborating on this case study and our findings in the following section, we'll report on how the learning and neural prediction is going to be automatized and deployed into a large-scale distributed predictive analytics software (sec. 3). We conclude with a treatment of related work (sec. 4) and a discussion (sec. 5).

2 Behavioural Prediction of Future Revenues

For our study we had available the actions and spendings performed by each user during its first 7 days in the game and then the information about the money spent after 180 days. Our working hypothesis was that it is possible to predict a user's future spendings (spendings after 180 days) from the actions he takes very early in the game (recorded events during first 7 days of play).

As a baseline method for comparing our learned behavioral model, we use a simple extrapolation based on just the money users spent during the observation period of the first 7 days, ignoring the behavior. This algorithm uses the ratio between the historical revenue during the prediction target period length and the historical revenue during the observation period length averaged over all users in the historical (training) data. This ratio is then multiplied by the revenue of the particular user whose prediction is being made [8]. Despite its simplicity this algorithm produces reasonable results and is regarded as the gold standard for revenue prediction. We also benchmark against a slightly improved, bias-corrected version of the baseline. In this improved version, users without any payments during the first 7 days of play are not assigned 0 but the somewhat larger expectation, calculated from the historic data about users who didn't pay during the first 7 days of play but started paying afterwards.

In our experiments, we started with supervised learning in a traditional setup:

a) Generation of high dimensional user profiles X.
b) Automatic feature selection (optional) $\phi : X \mapsto Z$.
c) Supervised learning of a predictor $f : \phi(X) \mapsto r$ with r the revenue.

Step a: Using the data available from the observation period we created several hundred features divided in 6 categories: spending, gameplay, game progression, social interactions, success metrics and game settings preferences. Some of these features were constructed manually (e.g.: passed an activity threshold) but most are 'automatic' (e.g.: id, min, max, sum, average, median, trend of time series and singular events).

Step b: For some methods, to reduce the dimensionality of the input space we performed automatic feature selecting using a a technique based on random forests [12].

Step c: We tested a number of linear (ordinary least squares [14], ridge [13]) and non-linear predictors (SVM [6], Random Forest [3], deep and shallow MLPs [18] trained with RPROP [17]).

Results. In these initial experiments, we found that many out-of-the box methods didn't produce good results when trained on predicting r, some–including all deep models we've tried–produced even worse results then the baseline methods. What brought an improvement and finally did the trick, was to not train on the revenue directly, but to only learn to predict an individual user's difference to an 'average user'. This was done by a) adding the output of the improved baseline to the input of the model and b) setting the learning target to the absolute difference between the baseline prediction and the user's actual revenue r.

Another improvement was achieved by using a composite model of two independent models; one model trained and used for those users, that did spend something in the observation period, and another model for those users who did not. An ensemble of two MLPs using the absolute-differences technique achieved the cross-validated mean squared error (MSE) of 966, substantially improving over the 1192 of the industry standard. This can be considered the best result of our study, concerning purely 'generic' and thus fully-automatizable methods.

We collected only the best results for different types of models in table 1 and added also a model with a complex, human-constructed architecture for a 'prediction correction' (MLP Composite + PC) module considering the data of similar users for each individual prediction. The construction of this model was the result of human investigation and usage of domain knowledge, thus cannot be easily done automatically. Nevertheless, it demonstrates, what can be accomplished within this domain.

Table 1. 10-fold Cross-Validated Regression Results. DK stands for domain knowledge, ltv7 refers to the spendings during the observation period. Models that have the improved baseline as input as well as all MLPs, were trained on absolute differences.

Model	**Input (X)**	**MSE**	**Non-Linear**	**DK**
Baseline	ltv7	1192		
Improved Baseline	ltv7	1104		
Ordinary Least Squares	ltv7	1087		
Ridge	Improved Baseline,ltv7	1053		
SVM	Improved Baseline, Behavioral	1380	X	
Random Forest	ltv7	1159	X	
MLP	Improved Baseline, ltv7	1061	X	
MLP Composite	Behavioral	966	X	
MLP Composite + PC	Multiple	886	X	X

Further experiments. To better understand the complexity of this data set and the observed high variances in the predictions, we did further experiments in the direction of a simple binary classification (paying / non-paying). In the t-SNE-generated [16] scatter plot (fig. 2) of the high-dimensional user profiles, there is no clear decision boundary between target classes of non-paying (red) and paying users (blue). There is some structure, for example in a clearly separated

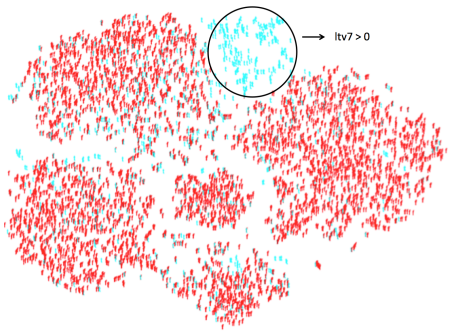


Fig. 2. t-SNE visualization of user profiles. The classification task is non paying (red 'f') and paying users (blue 't').

group of early spenders ($ltv7 > 0$), but also a huge amount of overlap. Reasons are the high randomness in users' decision to start spending and all sorts of external factors that are outside the scope of recorded ingame events.

Nevertheless, even in this situation, it's absolutely possible to come-up with a classification that is useful to developers, if one relaxes the decision criterion. The results presented in figure 1 using the RIPPER rule induction algorithm [5] for classifying if a user belonged to the class of heavy-paying user (in this particular case belonging in the top 15% highest paying users) or not. The results of the RIPPER algorithm were improved by reweighting the training instances in order to increase the cost of misclassification in the minority class [20]. After training, it was then possible to mark only about 4.6% percent of the user profiles in the testing set, who then actually generated more then 90% of the total revenue within the next 180 days (see fig. 1).

3 Integration Into a Predictive Analytics System

After proving behavioral prediction to be feasible, we decided to build a system for collecting and storing huge amounts of user events, for running learning methods against the data and for predicting revenues as well as any other target value of concern. The goal is to provide fully-automated feature selection, learning, and prediction to a number of games and services and also to do this in a highly-parallelized, distributed–thus scalable–way. The schema in figure 3 gives a general overview of the different modules that compose the system.

Collecting Events. The first concept in our system is that all the relevant game data a user generates is categorized into events. An event can be, for example, gaming session start, unit constructed, message sent or battle won. All the events are transferred directly from the game client to our system using REST protocols. To allow for very high loads of millions of events per hour, the event cache

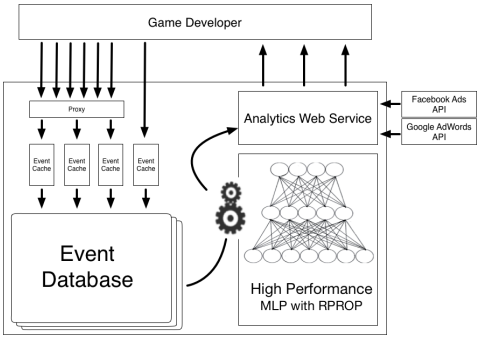


Fig. 3. Overview of the different modules of our predictive system and the way it interacts with game developers.

computers form a thin layer of additional servers in front of the database, which is tasked with caching the events and later insert them in the event database.

Big Data Storage. The event database is supposed to be a permanent storage place of all the raw events that occur in the game under analysis. Due to the characteristics and the huge amount of data that will be generated through events – billions to trillions over the years – and also due to the continuous event processing tasks that will be needed to run for generating and updating the user profiles, we use a non-relational distributed database implementation of Google's BigTable [4] named HBase. Our instances of HBase are scalable from clusters of several dozens to hundreds of node computers running distributed map-reduce tasks [7].

Aligning the data. When the users pass our initial target prediction time (180 days) we add them to our training data. The data of individual users is 'aligned' according to their 'age'; the time since account creation. This aligned data supports a sequence of models for predicting revenues on the basis of only the first 3 days of play, only the first 4 days of play, only the first 5 days of play, and so on. This allows us to give a very early prediction for players that come into the system, the first time, for example, on day 3 (with the 3-days model). We generate an improved prediction with every day the user grows older and more data becomes available (using the 4-days, ..., n-days model).

Making Predictions. Revenue predictions run daily on the users whose origin is from an active marketing campaign. The predictions are made by the N++2 neural network simulator which implements a highly parallelized multilayer perceptron with resilient propagation and is about 50-times faster than the sequential implementation in the original N++ [15]. (Re-)training can be run at a lower frequency, for example on a weekly basis, automatic feature selection would run with an even lower frequency.

Presenting results and insights. The predictions are available for the game developer under a web interface which also uses Google AdWords API and Facebook Ads API to integrate information about the ongoing marketing campaigns. Developers can see present acquisition cost per user, generated revenue per user so far, and predicted total revenue per user (the so-called average lifetime value of a user) next to each other, for every individual channel and marketing campaign. The integration of these API's also allows the developer to modify and tune ongoing marketing campaigns directly on our interface, next to the data he needs for his decision making.

4 Related Work

A previous study on knowledge extraction from player behavior was made by us on a freemium game called *Wack-a-Doo* where we focus on identifying and understanding which features help distinguish different types of users [1].

One of the techniques used for analysing player data is calculating metrics that serve as key performance indicators (KPIs); examples of KPIs in MMOGs could be the session times, the churn rate or if applicable even tutorial completion information [10]. Because of the more mature state of the field of web analytics there are some techniques that were adapted from this field and used in the context of MMOGs. Examples of this adaptation could be conversion rates analysis, user acquisition cost analysis or cohort analysis [11].

There are also data mining techniques used on more traditional computer games that have objectives such as behavior prediction [19], classification of user behavior [2], that can potentially provide very useful information to the MMOGs developers [9].

5 Discussion

Using data about the players behaviour inside the game we were able to beat the results of the de-facto industry gold standard for revenue prediction. This prediction helps developers improve their evaluation and optimization of marketing campaigns. We were also able to discern some other interesting ways for applying modern machine learning to a problem in this area.

Taking into account the needs of the game developers and the general architecture of this kind of games we proposed an automatic system that integrates collection, processing and storage of players' behavioural events, a prediction module and an interface where all the generated player information is displayed along with the marketing campaigns data. The objective of this system is to allow the game developers to evaluate their running marketing campaigns as quick as possible and to compare them.

We believe this domain to be of interest for further research, not only because of the huge amount of data but also because within this industry, there are many cases where only small improvements can already generate a huge, monetary benefit for the developers. In our experience, compared to other more traditional

industries game developers are already aware of this potential and very open to all kind of experimental machine learning methods.

References

1. Alves, J.; Neves, J.; Lange, S.; Riedmiller, M., "Knowledge Extraction from the Behaviour of Players in a Web Browser Game", Proc. of KICSS2013, 2013.
2. Anagnostou, K.; Maragoudakis, M., "Data Mining for Player Modeling in Videogames", 13th Panhellenic Conference on Informatics, 2009.
3. Breiman, L., "Random forest", Machine Learning, Vol. 45, 1999.
4. Chang, F.; Dean, J.; Ghemawat, S.; Hsieh, W. C.; Wallach, D. A.; Burrows, M.; Chandra, T.; Fikes, A.; Gruber, R. E, "Bigtable: A Distributed Storage System for Structured Data", ACM Transactions on Computer Systems 26, pp.1-26, 2008.
5. Cohen, W., "Fast Effective Rule Induction", Proc. of the Twelfth Int. Conf. on Machine Learning, vol.3, pp.115-123, 1995.
6. Cortes, C.; Vapnik, V, "Support-vector networks", Machine Learning, 20(3), pp.273-297, 1995.
7. Dean, J.; Ghemawat, S., "MapReduce : Simplified Data Processing on Large Clusters", Communications of the ACM, 51(1), pp.1-13, 2008.
8. Demler, C.; Faber, F., "Browsergames-Marketing mit eigenen Tools", Making Games, 4, 2012.
9. Drachen, A.; Thurau, C.; Togelius, J.; Yannakakis, G..; Bauckhage, C., "Game Analytics", in "Game Data Mining", pp.205-253, 2013.
10. Fields, T., "Game Analytics", "Game Industry Metrics Terminology and Analytics Case Study", Springer London, pp.53-71, 2013.
11. Fields, T.; Cotton, B., "Social game design: Monetization methods and mechanics", Waltham: Morgan Kauffman Publishers, 2011.
12. Genuer, R.; Poggi, J.; Tuleau-Malot, C., "Variable selection using random forests", Pattern Recognition, Letters 31, pp.2225-2236, 2010.
13. Hoerl, A., Kennard, R.; "Ridge regression: Biased estimation for nonorthogonal problems", Technometrics, 12(1), pp.55-67, 1970.
14. Hutcheson, G.; "Ordinary Least-Squares Regression", In The SAGE Dictionary of Quantitative Management Research, pp.224-228, 2011.
15. Labge, S.; "Tiefes Reinforcement Lernen auf Basis visueller Wahrnehmungen", Doctorate thesis, University of Osnabrck, 2010.
16. Maaten, L.; Hinton, G., "Visualizing Data using t-SNE", Journal of Machine Learning Research, 9, pp.2579-2605, 2008.
17. Riedmiller, M.; Braun, H.; A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, in Proc. of the ICNN, 1993.
18. Rumelhart, D.; McClelland, J., "Parallel distributed processing: explorations in the microstructure of cognition", Foundations, vol.1, MIT Press, 1986.
19. Shim, K., Srivastava, J., "Sequence Alignment Based Analysis of Player Behavior in Massively Multiplayer Online Role-Playing Games (MMORPGs)", Data Mining Workshops ICDMW 2010 IEEE Int. Conf., pp.997-1004, 2010.
20. Sun, Y.; Mohamed, S.; Andrew, K.; Yang W., "Cost-sensitive boosting for classification of imbalanced data", Pattern Recognition, 40, no.12: pp.3358-3378, 2007.
21. Wilson, R.D., "Using Web Traffic Analysis for Customer Acquisition and Retention Programs in Marketing", Services Marketing Quarterly, 26(2), pp.1-22, 2004.

Learning is hard work: Detecting dynamic obstacles in occupancy grid maps

Sven Hellbach[1], Frank Bahrmann[1], Sabrina Keil[1], Hans-Joachim Böhme[1⋆]

University of Applied Sciences Dresden, Artificial Intelligence and Cognitive Robotics Labs, POB 12 07 01, 01008 Dresden, Germany
{hellbach, bahrmann, keil, boehme}@informatik.htw-dresden.de

Abstract. This paper describes some ongoing research to model dynamic obstacles in occupancy grid maps. Two approaches from the field of machine learning have been evaluated. Even though, some results are already promising, the proposed problems, which appear to be simple in nature, seem to be challenging for machine learning approaches.

Keywords: HMM, MLP, occupancy grid maps, dynamic obstacles

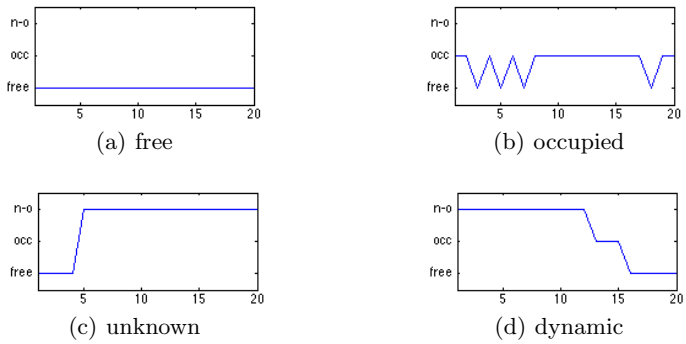
1 Introduction

In the field of mobile assistance robotics a number of scenarios like tour guide robots, shopping assistants, or elderly care, have been discussed in the recent years. For all of these scenarios some of the problems, that need to be solved, stay the same. One main aspect for being mobile, obviously is the navigation in the current environment. To achieve this several approaches have been published to represent the environment. One of the major representatives are occupancy grid maps [9].

Since assistance robots aim at supporting the human, they have to work in environments where a number of people are present and the environment is subject to changes. For some scenarios the environment can even be very crowded, e.g. tour guide robots in a museum. Unfortunately, the large number of dynamic obstacles can lead to malfunctions of the navigation algorithms. In particular, the localization will be error prone or even fail. Without going to much into detail, this is caused by the fact that the representation of the environment – the occupancy grid map – does not fit the current sensor impression any more, due to occlusion by the dynamic obstacles.

To cope with this problem, a few approaches have been published, that try to model dynamic obstacles into the occupancy grid maps. The experiments discussed in this paper are based on an approach by Meyer-Delius [8, 7], who uses Hidden Markov Models (HMM) with two states at each grid cell to describe the observed dynamics.

⋆ This work was supported by ESF grant number 100076162



(a) free (b) occupied

(c) unknown (d) dynamic

Fig. 1. Examples for the four different classes in our data. The observation over time has been windowed with 20 time steps lasting 100ms each. The data can be as clean as in (a) or noisy as in (b) for all four classes.

This paper is not meant to propose a new method that helps solving the described problem. Our intention is to point out a problem that seems to be very simple and for which a heuristic solution can probably be found quite easily, however machine learning approaches seem to have some difficulties to solve the problem adequately. However, facing a practical application mostly means we have to deal with changing environments. This results in the demand for an adaptive approach, which can not be achieved with a inflexible expert knowledge based solution.

To emphasize the contribution of this paper, we do not aim at a solution for coding cells that a frequently occupied with dynamic obstacles. Instead our focus is to decide whether a dynamic obstacle is present within in certain time frame. This time frame is meant to be as short as possible to classify a current measurement, and together with the classification result to decide whether this measurement can be used to adapt the map.

The remainder of this paper is divided as follows. We start with a description of the problem in Sec. 2. Subsequent two approaches from the field of machine learning based on Hidden Markov Models (HMM) in Sec. 3 are evaluated with respect to the described problem. For both the experimental results are directly discussed in the respective section. Finally, the paper concludes in Sec. 4.

2 Description of the problem

To compute occupancy grid maps, the environment around the robot is divided into discrete grid cells. Each grid cell then codes the probability of the corresponding position being occupied. This probability is derived from the sensor readings of the robot. The probabilities are assumed to be independent for the different grid cells. To gain a stable map and to cope with sensor noise, the observations of the sensor are superimposed over time. However, this has the

drawback that standard occupancy grid maps are only able to model static (or slow changing) environments. Commonly, the probabilities are furthermore discretized to 0.0 (free), 0.5 (unknown), and 1.0 (occupied) to improve the robustness of algorithms that rely on occupancy grid maps. The probability of 0.5 occurs if a certain cell can not be observed. This also happens if an obstacle is in the sensor's line of sight and hence occludes the cell.

Since assistance robots are used in crowded environments, the goal is now to somehow cope with the dynamic obstacle. A dynamic obstacle is an element of the environment, e.g. a person or a door, that changed its position over time. The already existing attempts to solve the problem can roughly be divided into those that try to track the object for filtering it out [1, 3, 12], and those that try to figure out whether a single grid cell currently is occupied by a dynamic obstacle [11, 8, 7]. For the first method some kind of tracking or at least detection system has to exist. This is hard to come up with for all possible dynamic obstacles. Hence, we want to focus on the second method here.

Summarizing, the problem we are facing can be regarded as a classification problem for time series (see Fig. 1). The data we need to classify originates from sensors that can usually be modeled as a ray, e.g. laser scanner, sonar scanner, or depth cameras. Thus, the state of the occupancy grid map cells can either be observed as *free* (where the ray passes through), *occupied* (the position of the obstacle), or *not observed* (all cells behind the obstacle, in the direction of the ray). From the depiction in Fig. 1 it becomes clear that the sensor readings need to be observed over a certain amount of time to be able to make a decision.

The classes we need to distinguish are *free*, *occupied*, *dynamic* (an obstacle traverses the cell), and *unknown* (the cell cannot be observed for a certain amount of time). Note that the data usually is noisy as it is exemplarily shown in Fig. 1(b). The examples shown for the classes *unknown* and *dynamic* are meant to illustrate that these two classes can be quite similar. Basically, the discriminative difference is the short observation of the cells being observed as occupied for one time step.

To generate the data we used a trained static map from a real life scenario as description for a simulated environment. Moving obstacles are modeled in the environment around the robot. The simulation is used to easily generate a large amount of data and not to simplify certain aspects of the problem.

After all, the problem occurs to be quite simple. We even benefit from a number of simplifications coming from the pre-processing methods, like the discretized observation probabilities. The first guess would certainly be that a hand crafted solution might come up easily. However, the (not yet sufficient) experiments with several machine learning methods suggest that this easy-looking problem still offers a challenge.

3 Hidden Markov Models

In particular in language processing, but also in other time series applications Hidden Markov Models (HMM) [10] belong to the most established approaches.

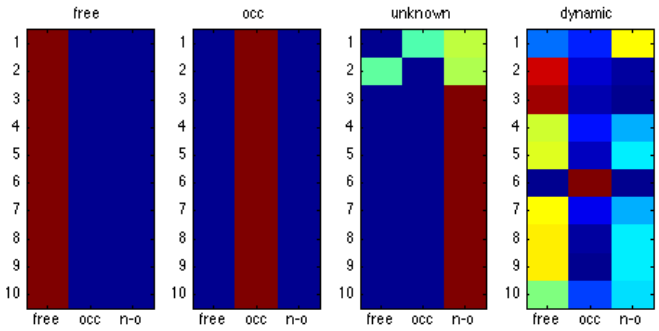


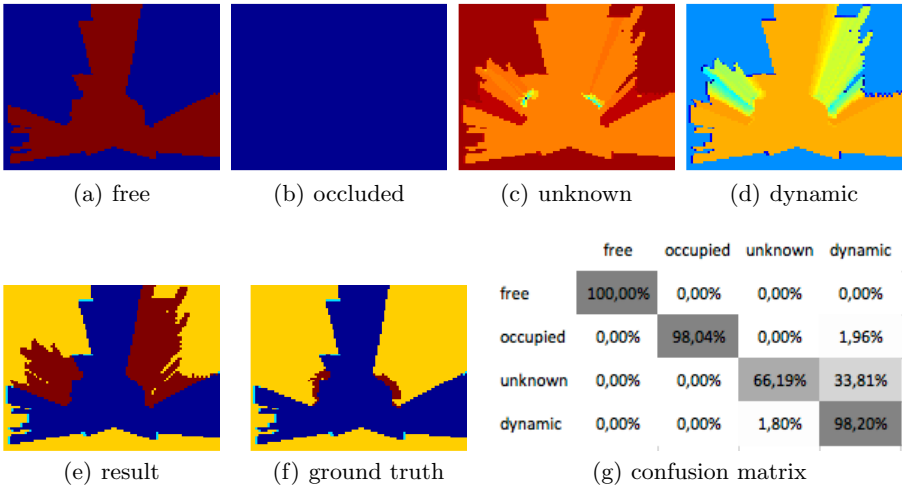
Fig. 2. HMM observation probabilities for the states 1 to 10. Each state model the discrete probabilities for observing free, occupied (occ), and not observed (n-o). The colors range from blue for a probability of 0.0 to red for a probability of 1.0 following the hue order.

Lately, they were also used for modeling the dynamics of the environment. For this, Meyer-Delius [8, 7] uses a two-state HMM in his approach. This basically describes whether there are changes occurring at all, which can also be due to noise or other side effects. In our opinion, which is based on previous experiments with the method proposed in [8, 7], it would be beneficial to focus on a longer time period.

Hence, we do not use a two state cyclic HMM, but a directed non-cyclic chain. For the experiments shown in this paper, four different HMMs – one for each class – were applied, each with ten states in the chain. Contrary to [8, 7] all parameters of the Hidden Markov Model were trained applying the expectation-maximization (EM) algorithm [10]. In Fig. 2 the learned observation probabilities are depicted for each class. As it can be seen, the different HMM can nicely be interpreted with respect to their respective class (compare Fig. 1).

Figure 3 summarizes the results achieved with the four HMMs. Each grid cell in the depicted local map is observed for a certain time. The first four plots (Fig. 3(a)-(d)) show the likelihood for the observed sequence being modeled by the respective HMM. For the computation of the classification decision a maximum likelihood approach is used. The results of this (Fig. 3(e)) are than compared with the ground truth (Fig. 3(f)) to compute the confusion matrix (Fig. 3(g)). To derive the local maps the robot is positioned at the lower center position. A laser scanner with a field of view of 270° is modeled. From the laser scans a local occupancy grid map is derived, which serves as input for the classification.

Even though for some classes the results are already satisfactory, the major problem arises in the cells that are occluded by the dynamic obstacle. The change from dynamic (red) to unknown (yellow) takes place at a position where a wall can be found in the underlying global map. Behind this wall the cells can never be observed, which seems to be simple enough to be classified correctly. In front of the wall the case depicted in Fig. 1(c) is observed.



(a) free (b) occluded (c) unknown (d) dynamic

(e) result (f) ground truth (g) confusion matrix

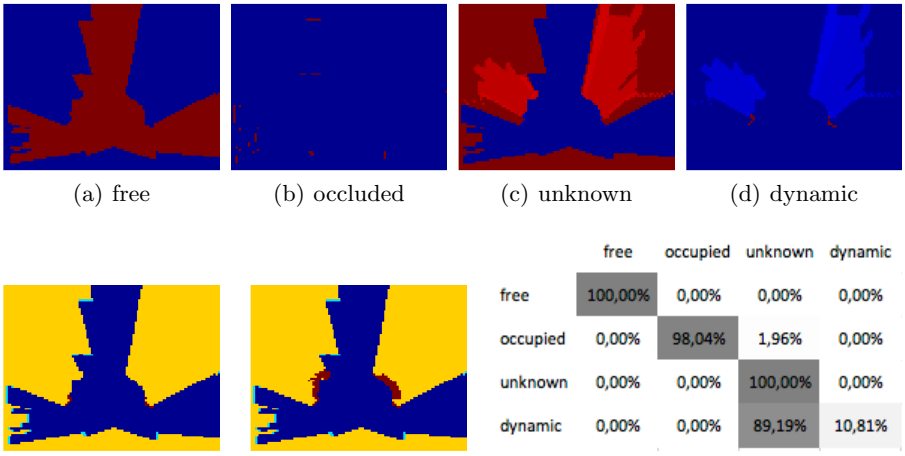
Fig. 3. Classification results using four HMMs. (a)-(d) The likelihood of one grid cell belonging to the respective class (color scale is logarithmic and follows the hue order from blue for a low probability to red for high probability). The result in (e) is gained by maximum likelihood selection. The true class is depicted in (f). For both, the colors stand for free (blue), occupied (cyan), dynamic (red), and unknown (yellow). Finally (g) shows the misclassifications in a confusion matrix the rows contain the true class, while the columns the predicted class.

A closer look at the likelihoods of each single HMM reveals that at the positions, where the misclassification occurs, the HMMs are uncertain. A maximum selection does not seem to be the adequate decision strategy at this point.

3.1 Post-Classification using Multi-Layer-Perceptron

From the results of the HMM classification, we concluded, that the maximum likelihood selection seems to be the cause of the misclassifications. The decision obviously has to deal with the fact, that all single class HMMs seem to be uncertain about the classification results. Our idea is to use a multilayer perceptron (MLP) with a single hidden layer to serve as decision maker. A similar approach has already been proposed in [5] using SVM instead of MLP.

The likelihoods from each HMM serve as input for the MLP, which is trained according to the labeled class information of our data set. Figure 4 shows the result applying the same trained HMMs as in 3. Again the first plots (Fig. 4(a)-(d)) show the results for one single class. This time we use the activation of the output neurons to gather the depicted information. Here as well, the results (Fig. 4(e)) are compared with the ground truth (Fig. 4(f)) to compute the confusion matrix (Fig. 4(g)).



(a) free (b) occluded (c) unknown (d) dynamic

(e) result (f) ground truth

	free	occupied	unknown	dynamic
free	100,00%	0,00%	0,00%	0,00%
occupied	0,00%	98,04%	1,96%	0,00%
unknown	0,00%	0,00%	100,00%	0,00%
dynamic	0,00%	0,00%	89,19%	10,81%

(g) confusion matrix

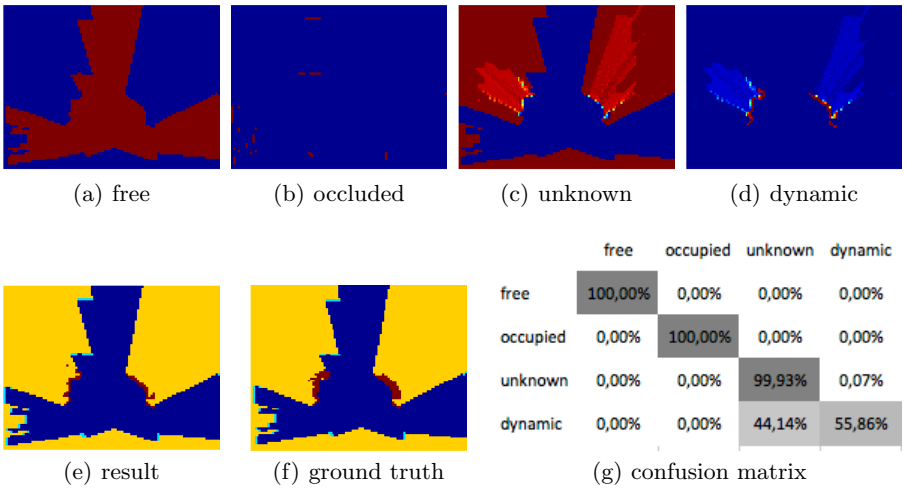
Fig. 4. Classification results using four HMMs together with a MLP. (a)-(d) The activation of the output neurons of one grid cell responsible for the respective class (the colormap follows the hue order from blue for a low activation to red for high activation). The result in (e) is gained by maximum selection. The true class is depicted in (f). For both, the colors stand for free (blue), occupied (cyan), dynamic (red), and unknown (yellow). Finally (g) shows the misclassifications in a confusion matrix the rows contain the true class, while the columns the predicted class.

Obviously, the MLP helped to eliminate the misclassification in the shadow of the dynamic objects. However, this came with the price that also most for the grid cells containing a dynamic obstacle cannot be classified correctly any more.

As stated earlier, to compute occupancy grid maps the assumption is made that all grid cells are statistically independent. With this assumption it is possible to process occupancy grid maps in real time. Even though this assumption is worth to be discussed it works for applications in static environments. However, for a dynamic environment it should be possible to improve our results, since dynamic obstacle moves from one cell to an other without appearing out of thin air and even tend to occupy multiple cells.

From a mathematical stand-point, we could start modeling the dependence on the neighboring cells, e.g. with the help of Markov random fields to connect the HMM in the spatial dimension. However, we still have to obey the requirement for real time processing. A simple idea is to model the neighborhood after the pre-classification of the grid specific HMMs and use the responses of the eight neighboring cells as additional input to the MLP, which results in an input dimension of 36.

The results for this idea are shown in Fig. 5. The activation of the output neurons are plotted in Fig. 5(a)-(d). The confusion matrix (Fig. 5(f)) is derived



	free	occupied	unknown	dynamic
free	100,00%	0,00%	0,00%	0,00%
occupied	0,00%	100,00%	0,00%	0,00%
unknown	0,00%	0,00%	99,93%	0,07%
dynamic	0,00%	0,00%	44,14%	55,86%

Fig. 5. Classification results using four HMMs together with a MLP that considers the eight neighboring cells. (a)-(d) The activation of the output neurons of one grid cell responsible for the respective class (the colormap follows the hue order from blue for a low activation to red for high activation). The result in (e) is gained by maximum selection. The true class is depicted in (f). For both, the colors stand for free (blue), occupied (cyan), dynamic (red), and unknown (yellow). Finally (g) shows the misclassifications in a confusion matrix the rows contain the true class, while the columns the predicted.

from a comparison between the the classification results (Fig. 5(e)) and the ground truth(Fig. 5(f)).

Even though there are still some misclassifications the results are very promising. A closer look reveals that the side of the dynamic obstacles facing to the robot is classified correctly. The grid cells behind this front row should show the same behavior as for the *unknown* class if no noise would be present. Due to the accumulation of the ground truth data over the time window some artifacts might occur resulting from the noise. Hence, it is important for further experiments to verify whether the ground truth labels are correct. Furthermore, it might be useful to introduce yet another class that codes the noise.

4 Conclusion

This paper presented a simple yet challenging problem for machine learning. Even though it seems that an adequate solution has been found, the question arises why this combination of both neural and probabilistic methods is necessary. Would it not be possible to use only one paradigm to solve the problem? Several aspects of the problems were discussed by evaluating solutions for the problem applying Hidden Markov Models, multi-layer perceptrons and time de-

lay neural networks. The lessons learned from these experiments is that (a) one solution for the problem is a combination of HMM and MLP, (b) the neighborhood context has to be taken into account, (c) the wrong representation can make the problem even more difficult.

Since this paper is meant to describe ongoing research there are a number of approaches that are not yet addressed here. For example, recurrent neural networks (RNN) don't have to face the problem with a constant size time window. In particular echo state networks (ESN) will be taken into account [4]. Another interesting approach is the idea behind convolutional neural networks (CNN) [2, 6] that basically learn the right convolution for the presented problem. This seems to be a promising idea for learning the right form of representation. Finally, relevance learning seems to be an adequate tool for selection the right representation from a set of given representations.

References

1. Fox, D., Burgard, W., Thrun, S.: Markov Localization for Mobile Robots in Dynamic Environments. Journal of Artificial Intelligence Research 11, 391–427 (1999)
2. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36(4), 193–202 (1980), http://dx.doi.org/10.1007/BF00344251
3. Hähnel, D., Triebel, R., Burgard, W., Thrun, S.: Map Building with Mobile Robots in Dynamic Environments. In: ICRA. pp. 1557–1563 (2003)
4. Jaeger, H.: The ''echo state'' approach to analysing and training recurrent neural networks - with an Erratum note. Tech. rep., GNRC for IT (2001)
5. van der Maaten, L.: Learning Discriminative Fisher Kernels. In: Getoor, L., Scheffer, T. (eds.) ICML. pp. 217–224. Omnipress (2011)
6. Matsugu, M., Mori, K., Mitari, Y., Kaneda, Y.: Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks 16(5–6), 555 – 559 (2003)
7. Meyer-Delius, D., Beinhofer, M., Burgard, W.: Occupancy grid models for robot mapping in changing environments. In: AAAI. pp. 2024–2030 (2012)
8. Meyer-Delius, D., Hess, J., Grisetti, G., Burgard, W.: Temporary maps for robust localization in semi-static environments. In: IROS. pp. 5750–5755 (2010)
9. Moravec, H., Elfes, A.: High Resolution Maps from Wide Angle Sonar. In: ICRA. pp. 116–121 (1985)
10. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (Feb 1989)
11. Saarinen, J., Andreasson, H., Lilienthal, A.J.: Independent Markov Chain Occupancy Grid Maps for Representation of Dynamic Environment. In: IROS. pp. 3489–3495 (2012)
12. Wang, C.C., Thorpe, C.: Simultaneous Localization and Mapping with Detection and Tracking of Moving Objects. In: ICRA. pp. 2918–2924 (2002)

Transfer Learning without given Correspondences

Patrick Blöbaum and Alexander Schulz

University of Bielefeld - CITEC centre of excellence, Germany
{pbloebau|aschulz}@techfak.uni-bielefeld.de

Abstract. The most transfer learning approaches are based on the assumption of having correspondence information between two sets of observations. This knowledge allows finding correlations among these different observation spaces for a knowledge transfer. Therefore, this correspondence information is crucial. We present an approach for finding unknown correspondences between two data sets under the assumption of having the same shared latent structures. For this, we optimize a generative model such that it generates one observed set, while being based on the other. We perform this optimization in a low dimensional space, such already allowing a transfer of information. Additionally, the found correspondences can be used in combination with any other transfer learning method.

Keywords: Transfer Learning, Generative Models, Finding Correspondence, Manifold Alignment

1 Introduction

For predicting unseen data, generalization is an important aspect in machine learning. The most used assumption to grant some kind of generalization is that feature space and marginal data distribution remain the same. In fact, there are many cases where this is not valid. For example, training the model with data that has been recorded under lab conditions and then going into practical application. This usually leads to additional changes in the data and thus the learned model would be inaccurate. Another scenario could be having only few or even no data for a specific problem, while in another similar problem are plenty available. Instead of learning a completely new model, we could be interested in using the knowledge from previous or related models. Transfer Learning (TL) addresses this problem how the knowledge from a task can be transferred to another similar task. TL can be divided in many different settings with different conditions. For further information, Pan and Yang [9] briefly summarize these in a survey.

In the following we distinguish between *source* and *target* domain, where we are interested in transferring knowledge from a source problem to a related target problem. For this, some knowledge about the relationship between source and target domain has to be available. Therefore, the most TL approaches require

correspondence information between two related sets of observations $\mathbf{X} \subset \mathcal{X}$ and $\mathbf{Y} \subset \mathcal{Y}$. Correspondence is the knowledge about which observation sample $\mathbf{x}_i$ from some source data $\mathbf{X}$ relates to which sample $\mathbf{y}_i$ from some target data $\mathbf{Y}$. Having this knowledge allows finding correlations among these different observation spaces and even allows to transform samples from one feature space into the other. Therefore, this correspondence information is crucial.

Since there exist many TL methods assuming to have correspondence information, only a few approaches try to deal with problems where this information is not available. We want to present our approach for finding correspondences between two data sets under the assumption of having same shared latent structures. The found correspondences can be used in any other TL method.

2 Related Work

Many TL approaches have the mentioned assumption that source and target space have some latent structures in common and try, therefore, to find a shared latent space of two sets of observations. One generative approach for this is proposed by Ek et al. [4]. They extend the Gaussian Process Latent Variant Model [7] by a shared and private feature space for each observation set. The shared and the private latent space together model the generation of one feature space. The shared latent space is found by using Canonical Correlation Analysis (CCA) [5] which finds the directions in two observation spaces that are maximally correlated. On the other hand, the private spaces are found by using Non-Consolidating-Component-Analysis [4], an extension of CCA, which finds the directions of maximum variance in each observation space that are orthogonal to the shared bases. However, this approach still requires the corresponding information between the observations sets.

An approach for TL without correspondences is introduced by the work of Pan et al. [8]. Their main idea is to find a shared latent space where the marginal distribution of both observation spaces are close to each other. This is done by minimizing:

$$\text{dist}(\mathbf{X}, \mathbf{Y}) = \left\| \frac{1}{n_1} \sum_{i=1}^{n_1} \phi(\mathbf{x}_i) - \frac{1}{n_2} \sum_{i=1}^{n_2} \phi(\mathbf{y}_i) \right\|_{\mathcal{H}} \tag{1}$$

where n_1 is the number of elements in observation set $\mathbf{X}$, n_2 in $\mathbf{Y}$ and $\phi(\mathbf{x}) : \mathcal{X} \to \mathcal{H}$. $\mathcal{H}$ denotes the reproducing kernel Hilbert space (RKHS). Therefore, based on the Maximum Mean Discrepancy theory [2], (1) can be seen as the distance between two distributions by calculating the distance between the means of the samples mapped into a RKHS (see [8] for further details). However, this approach tries only to match the first moments of both spaces and, hence, its capabilities are restricted.

Wang and Mahadevan proposed a framework to find a shared latent space without correspondences [12]. Their idea is to first find a joint structure to describe source and target data by local geometries and then join the two manifolds.

Their approach is based on the minimization of the following cost function:

$$\begin{aligned} C(\boldsymbol{\alpha}, \boldsymbol{\beta}) =& \mu \sum_{i,j} (\boldsymbol{\alpha}^T \mathbf{x}_i - \boldsymbol{\beta}^T \mathbf{y}_j)^2 W^{i,j} \\ &+ 0.5 \sum_{i,j} (\boldsymbol{\alpha}^T \mathbf{x}_i - \boldsymbol{\alpha}^T \mathbf{x}_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (\boldsymbol{\beta}^T \mathbf{y}_i - \boldsymbol{\beta}^T \mathbf{y}_j)^2 W_y^{i,j} \end{aligned} \tag{2}$$

where μ is a weighting factor, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ a projection to the shared latent space, $W^{i,j}$ the similarity between $\mathbf{x}_i$ and $\mathbf{y}_i$ and $W_x^{i,j}$ and $W_y^{i,j}$ the similarities between points within each observation space. The first term penalizes differences in the shared latent space while the other two terms guarantee the preservation of the neighborhood relationship within each data set. The goal is to find $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ such that (2) is minimized (see [12] for more details). The similarities $W^{i,j}$ model the correspondences between both observation sets. Although this approach is more powerful than [8], it relies on a heuristic cost function with parameters which need to be tuned.

3 Our Approach

In the following we present our algorithm for transfer learning without prior knowledge about the correspondences. It is based on maximizing the probability that a generative model learned from one set generates the other. We assume that $\mathbf{X}$ and $\mathbf{Y}$ were generated by the same latent manifold and we try to find a linear projection matrix $\mathbf{W}$ which aligns the two manifolds of $\mathbf{X}$ and $\mathbf{Y}$ in a low dimensional space. Based on this projection, we create correspondences by linking the most similar points between both observations (see Figure 1).

Given the mean centered matrices $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_N]$ and $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_K]$ where the points live on arbitrary dimensional manifolds $\mathcal{X}$ and $\mathcal{Y}$. We first find a latent space for the points $\mathbf{X}$. For this purpose, we utilize a linear mapping onto the first p principal components of $\mathbf{X}$. The resulting points in the p-dimensional space are denoted by $\mathbf{Z} = [\mathbf{z}_1 \mathbf{z}_2 \dots \mathbf{z}_N]$ where $\mathbf{z}_i \subset \mathcal{Z}$ represent points in the latent space. With respect to the assumption that $\mathbf{X}$ and $\mathbf{Y}$ share the same latent space, we now want to find a projection matrix $\mathbf{W} : \mathcal{Y} \to \mathcal{Z}$ such that a density estimation model based on $\mathbf{WY}$ generates the points $\mathbf{Z}$ with high probability.

For this purpose, we employ a probabilistic approach. We define a Gaussian Mixture Model with centroids $Q = \{\mathbf{W}\mathbf{y}_1, ..., \mathbf{W}\mathbf{y}_K\}$ in the standard way as a linear supposition of Gaussians. Then, the probability that this model generates a point $\mathbf{z}$ is

$$p(\mathbf{z}|\mathbf{W}, \mathbf{Y}) = N_{\mathrm{G}} \sum_{k=1}^{K} \pi_k \exp\left(-\frac{\|\mathbf{z} - \mathbf{W}\mathbf{y}_k\|^2}{2\sigma^2} \right), \tag{3}$$

where N_{G} is the normalization constant for Gaussians.

Similarly as in [3, 1], we introduce binary latent variables $\mathbf{h}_n$ which have a 1-of-K representation, i.e. $h_{nk} \in \{0, 1\}$ and $\sum_k h_{nk} = 1$. Following the literature of GMMs, we assume that each Gaussian generates points $\mathbf{z}_n$ and this is modeled

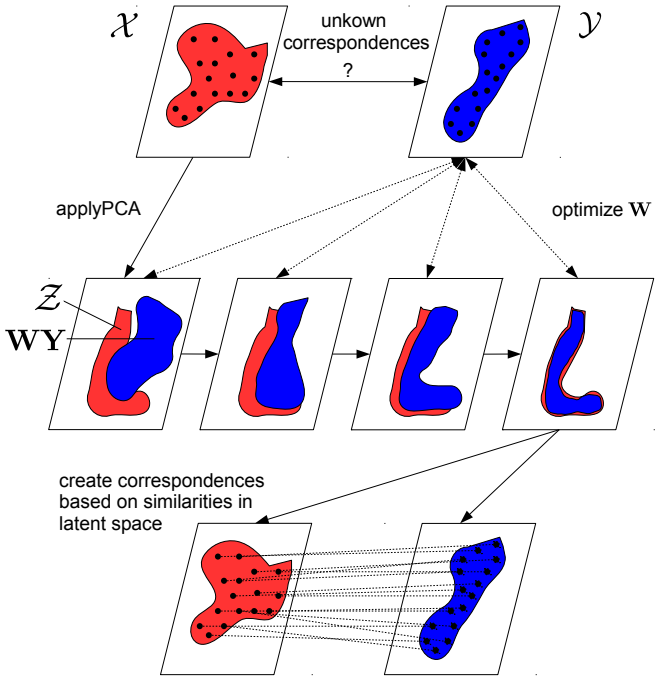


Fig. 1. Overview of the EM approach for finding correspondences between source (red) and target (blue) space. First, we linearly project the source data onto a latent space. Afterwards, we find an optimal projection matrix $\mathbf{W}$ to fit the target data in the latent source space. The correspondences can then be made by comparing the similarities in the shared latent space.

by the random variable $\mathbf{h}_n$. The distribution of this random variable is $p(\mathbf{h}) = \prod_{k=1}^{K} \pi_k^{h_k}$ with $p(h_k = 1) = \pi_k$, where we set $\pi_k = 1/K, \forall k$, for simplicity.

Following the literature, we are interested in maximizing the log-likelihood $\ln p(\mathbf{Z}, \mathbf{H}|\mathbf{W}, Y)$ with

$$p(\mathbf{Z}, \mathbf{H}|\mathbf{W}, \mathbf{Y}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{h_{nk}} \left(N_{\mathrm{G}}\right)^{h_{nk}} \exp\left(-\frac{\|\mathbf{z}_n - \mathbf{W}\mathbf{y}_k\|^2}{2\sigma^2}\right)^{h_{nk}}. \quad (4)$$

But since the hidden variables $\mathbf{h}_n$ are unknown, we consider the expectation

$$\mathbb{E}_{\mathbf{H}}\left[\ln p(\mathbf{Z}, \mathbf{H}|\mathbf{W}, \mathbf{Y})\right] = \sum_{n=1}^{N} \sum_{k=1}^{K} \mathbb{E}_{\mathbf{H}}\left[h_{nk}\right] \left(\ln(\pi_{nk} N_{\mathrm{G}}) - \frac{\|\mathbf{z}_n - \mathbf{W}\mathbf{y}_k\|^2}{2\sigma^2}\right) \quad (5)$$

where

$$\gamma_{n,k} := \mathbb{E}_{\mathbf{H}}\left[h_{nk}\right] = \frac{\exp\left(-\frac{\|\mathbf{z}_n - \mathbf{W}\mathbf{y}_k\|^2}{2\sigma^2}\right)}{\sum_{j=1}^{K} \exp\left(-\frac{\|\mathbf{z}_n - \mathbf{W}\mathbf{y}_j\|^2}{2\sigma^2}\right)} \quad (6)$$

can be seen as the responsibility of the k'th component of the mixture for the n'th point.

Now, we can maximize (5) with the popular iterative scheme known as the EM algorithm which converges to a local optimum [3, 1]: For initial parameters $\mathbf{W}^t$ we estimate $\gamma_{n,k}$ (E-step). Based on these fixed responsibilities, we maximize (5) with respect to $\mathbf{W}$ yielding new parameters $\mathbf{W}^{t+1}$ (M-step). Note that, in addition to the E-step, also the M-step can be computed in closed form since it requires to find a solution for a weighted least squares problem. Further, an additional regularization in this step can often be advantageous. This can be modeled straight forward by a Gaussian prior with zero mean.

For the bandwidth σ we employ the deterministic annealing scheme [11], i.e. we decrease the value by a small amount every EM-step. The whole algorithm is summarized in the following.

---

function FINDCORRESPONDENCES($\mathbf{X}$, $\mathbf{Y}$, p, maxIterations)
  $\mathbf{Z} \leftarrow$ ProjectWithPCA($\mathbf{X}, p$) ▷ p is the target dimension
  $\mathbf{W} \leftarrow$ ComputeProjMatrixPCA($\mathbf{Y}, p$) ▷ Initial projection matrix
  $\sigma \leftarrow$ initialize
  for i = 1 to maxIterations **do**
    for all n, k **do**
      $\gamma(n,k) \leftarrow$ calculateResponsibility($\mathbf{W}\mathbf{y}_k, \mathbf{z}_n, \sigma$) ▷ E-Step
    $\mathbf{W} \leftarrow$ solveWeightedLeastSquares($\mathbf{Y}, \mathbf{Z}, \boldsymbol{\gamma}$) ▷ M-Step
    $\sigma \leftarrow$ decrease(σ)
  correspondences $\leftarrow$ findOptimalAssignment($\boldsymbol{\gamma}$)
  return correspondences
end function

In order to find assignments, the responsibilities from the last iteration step can be used. It is possible to either have binary assignments by taking the most probable neighbor or having soft assignments represented by the responsibilities. In addition to providing correspondences, also transfer learning is possible by mapping source points from the latent space $\mathcal{Z}$ to the target space with the pseudo inverse of $\mathbf{W}$. If label information are available, this can be done class-wise for reducing the problem to only find correspondences within class structures, which is a lot easier.

An alternative stopping criteria for the algorithm is also possible. For example, comparing the alteration of the Frobenius norm of $\mathbf{W}$ between two iteration steps or checking if σ is smaller than a specific value. Using the log-likelihood (5) as a criteria is difficult due to the annealing of the bandwidth which changes the likelihood function in each step.

Regarding the computational complexity of this algorithm, the M-step is the most crucial part. We use the closed form solution $\mathbf{W} = (\mathbf{Y}^{*T}\mathbf{Q}\mathbf{Y}^*)^{-1}\mathbf{Y}^{*T}\mathbf{Q}\mathbf{Z}^*$. Assuming $\mathbf{Y}$ is a $K \times d$ and $\mathbf{Z}$ a $N \times p$ matrix, then let $\mathbf{Y}^*$ be a $NK \times d$ and $\mathbf{Z}^*$ a $NK \times p$ matrix representing all combinations of $\mathbf{Y}$ and $\mathbf{Z}$, weighted by a $NK \times NK$ diagonal matrix $\mathbf{Q}$. The complexity of $\mathbf{Y}^{*T}\mathbf{Q}\mathbf{Y}^*$ is $\mathrm{O}(d^2NK)$ and

Table 1. Percentage of points which have the true corresponding point among the k nearest neighbors.

data sets	k	Average correspondence quality
data set1	1	98.8%
	2	100%
data set2	1	98.2%
	2	100%
data set3	1	98.3%
	2	100%
data set4	1	97.1%
	2	100%
MNIST	1	97%
	2	100%

for the inversion $O(d^3)$. The asymptotically dominating upper bound for each iteration step is therefore $O(d^2NK + d^3)$.

4 Experiments

In the following experiments, we evaluate our approach only in terms of the computed correspondences. A good performance here implies that the manifolds of the two domains have been matched well and this provides a good starting point for further transfer learning.

We evaluate the correspondence quality of our approach in different toy scenarios and two possible scenarios involving the MNIST data set. In each test we have a source data set $\mathbf{X}$ and a target data set $\mathbf{Y}$. The target data consists of the modified source data in order to have a reference for the optimal assignment of an observation sample $\mathbf{x}_i$ to $\mathbf{y}_i$ for evaluation. We apply our approach to the data sets and compare the found correspondences with the optimal one. This is done by checking if the optimal assignment $\mathbf{x}_i$ to $\mathbf{y}_j$ is one of the k-nearest neighbor in a found two dimensional shared latent space. The results are averaged over 20 runs with randomly generated or sampled data points.

For *data set1* we generate 100 two dimensional data points on an "L"-shape and add a third noise dimension. This serves as source data $\mathbf{X}$. Here, we generate noise with small variance. The target data $\mathbf{Y}$ consists of the 45 degree rotated source data (see Figure 2). Our approach should be able to find a projection for $\mathbf{Y}$ which covers the latent structure of $\mathbf{X}$. To evaluate this, we check if the optimal correspondence $\mathbf{y}_i$ is one of the k-nearest neighbor of $\mathbf{x}_i$ in the latent space. The results for $k = 1, 2$ are summarized in Table 1. Already the direct neighbors agree almost perfectly. This shows that the latent space of $\mathbf{X}$ is accurately covered by $\mathbf{Y}$.

To increase the complexity of data set1, we modify the data set in two ways. First, we add 4 additional noisy features to the target data $\mathbf{Y}$ with much higher variance than the two intrinsic ones. We refer to this set by *data set2*. The mapping on the first two principal components of $\mathbf{Y}$ is shown in Figure 3. Second,

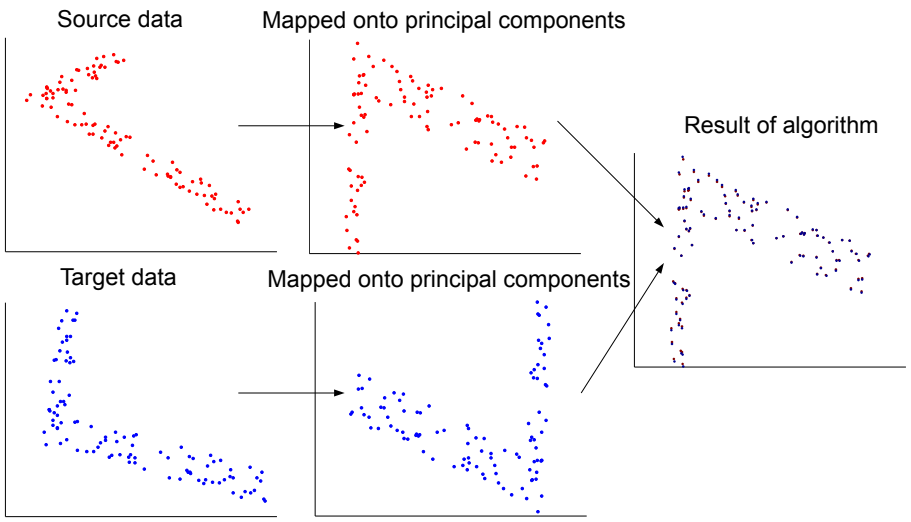


Fig. 2. Example of a toy test case with L-shaped data. The target data consists of the 45 degree rotated source data. In this case, the first two principal components of the target data are the reflection of the source data. Our algorithm tries to maximize the probability that the source data were generated by the GMM defined on the target data and, hence, aligns the latent spaces.

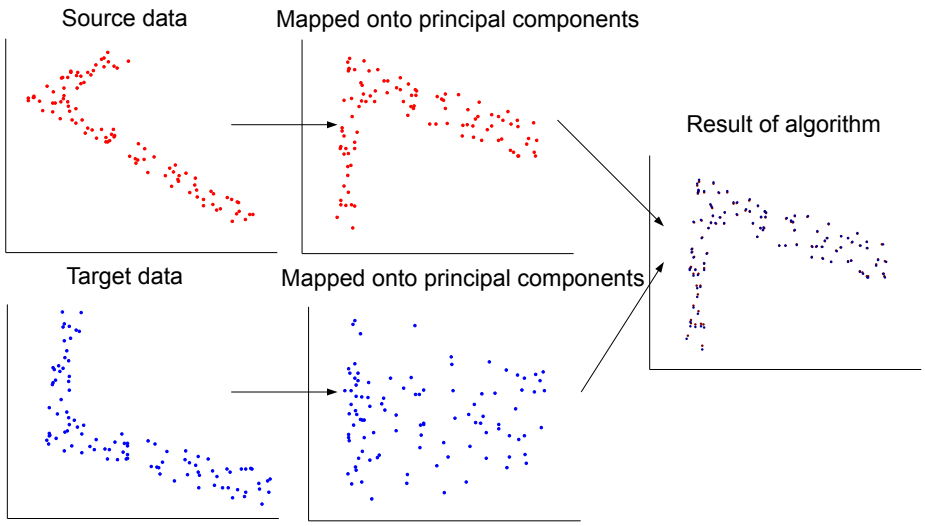


Fig. 3. Here, the rotated target data has additional noisy dimensions with much higher variance than the two intrinsic ones. The algorithm is still able to align the latent spaces.

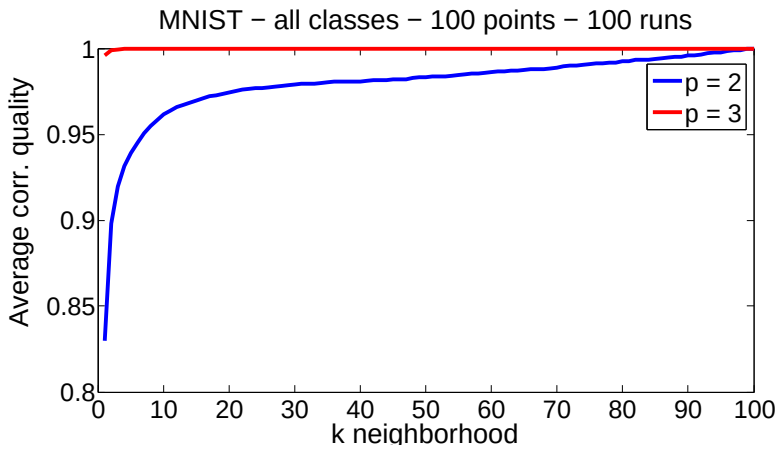


Fig. 4. The overall quality of the found correspondences with respect to the k-nearest neighbors. Blue represents the results for $p = 2$ and red for $p = 3$.

for a more realistic scenario we add an additional leg of the L-shape to the target data in 2 further dimensions. This is referred as *data set3*. The average correspondence quality of both data sets is shown in Table 1. Also in these two cases, the found correspondences are nearly optimal.

For *data set4*, we generated two random projection matrices with entries sampled from $\mathcal{N}(0, 1)$ to project the source and the rotated target data from data set2 into a 20-dimensional space. The agreement of neighbors is shown in Table 1. Since the latent structure remains the same and the random projections are linear, there were no problem to find the correspondences. Two additional tests were done with the *MNIST* data set which consists of images of handwritten digits from 0 to 9. The source data are the original images and the target data are the same images scaled down to 0.5 of the original size. For each run, we randomly select 100 images from the class of the digit "4". We obtain similar results as before, as shown in Table 1. Scaling the images can be seen as a linear operation and the results should, therefore, be similar to those with data set4.

In a last test we utilize all classes from the MNIST data set. For the full data set, linear mappings are probably not enough to correctly map the data set in a two dimensional space and, hence, the quality drops. Further, the quality heavily depends on the selected points because in some cases the sampled points have a symmetric structures which is difficult or impossible to correctly cover by linear operations. The result for all values of k and averaged over 100 runs are shown in Figure 4. For higher values of p the average quality with $k = 1$ was nearly 100% except of some cases with symmetric structures.

In our experiments we used PCA for the mapping from $\mathbf{X}$ to $\mathbf{Z}$, but any linear dimensionality reduction approach can be utilized. Some approaches, such as probabilistic PCA [10], try to cope with noise in the data and this could generally lead to better results. However, in our experiments were no significant differences.

5 Discussion

We presented an algorithm for finding unknown correspondences between two data sets from different feature spaces. Since many methods exist to find a shared latent space, the most approaches assume correspondence information. Although our approach already allows linear TL, the resulting correspondences from our algorithm can also be used for any kind of TL method. We evaluated the quality of the found correspondences by comparing them with the optimal ones. The results are very promising, but since we use a linear approach, it struggles in non-linear cases.

Therefore, the next natural step would be to extend the currently linear mapping to a non linear one, such allowing much more powerful TL. Current tests show promising results by kernelizing our approach, but this needs much more regularizations with respect to $\mathbf{W}$ and the additional kernel bandwidth. We are also still checking other non linear techniques, such as Extreme Learning Machines (ELM) [6]. The output weights of an ELM can be represented by $\mathbf{W}$ which allows an easy integration into our algorithm.

Another interesting idea could be to develop a similar EM algorithm for the approach of Wang and Mahadevan. Since they try to find and align a shared latent space of $\mathbf{X}$ and $\mathbf{Y}$ at the same time, optimizing (2) by using responsibilities instead of local features could lead to better results in terms of the manifold alignment. In this context, CCA could be used for finding $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

ACKNOWLEDGEMENTS

Funding from DFG under grant number HA2719/7-1 and by the CITEC center of excellence is gratefully acknowledged.

References

1. C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
2. K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schoelkopf, and A. J. Smola. Integrating structured biological data by kernel maximum mean discrepancy. In *IN ISMB*, page 2006, 2006.
3. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, 39(1):1–38, 1977.
4. C. H. Ek and N. Lawrence. *Shared Gaussian Process Latent Variables Models*. PhD thesis, Oxford Brookes University, 2009.
5. W. Härdle and L. Simar. Canonical correlation analysis. In *Applied Multivariate Statistical Analysis*, pages 321–330. Springer Berlin Heidelberg, 2007.
6. G. B. Huang, Q. Y. Zhu, and C. K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.
7. N. Lawrence. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *J. Mach. Learn. Res.*, 6:1783–1816, Dec. 2005.

8. S. J. Pan, J. T. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, AAAI'08, pages 677–682. AAAI Press, 2008.
9. S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
10. M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.
11. N. Ueda and R. Nakano. Deterministic annealing em algorithm. *Neural Netw.*, 11(2):271–282, Mar. 1998.
12. C. Wang and S. Mahadevan. Manifold alignment without correspondence. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1273–1278, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

Enforcing interpretability in classification by modelling constrained optimization problems using LVQ decision rules

Barbara Hammer[1], David Nebel[2], Martin Riedel[2], Thomas Villmann[2]

1 – Bielefeld University, Germany
bhammer@techfak.uni-bielefeld.de
2 – University of Applied Sciences Mittweida, Germany
{nebel,riedel,villmann}@hs-mittweida.de

Abstract. Interpretable classification models become more and more interesting for practitioners from various domains like for instance health care or spectral analysis. The strength of prototype methods allow them to inspect the resulting prototypes in the same way as the given data. They greatly benefit of having representative models in order to get additional insights. In this contribution we introduce formulations for two prototype classification methods: Generalized learning vector quantization (GLVQ) and a probabilistic variant of it, robust soft LVQ (RSLVQ). With the use of their specific classification rules, we enforce discrimination by hard constraints where representivity of the models is maximized. Since both objectives, discrimination power and representivity, may be contradictory, we also soften the optimization problem (OP) by introducing slack variables in order to allow misclassifications. These models show impressive classification performance on a two benchmark datasets.

1 Introduction

Prototype based classification models such as LVQ [4] and modern variants of it (Generalized LVQ [6] or Robust Soft LVQ [8]) represent data in terms of representatives (prototypes). This has one major benefit: The resulting models are directly interpretable by humans since prototypes can be inspected in the same way as data points. It may help to answer questions like what is typical for a certain class with respect to discriminate them from another. In [3] it is shown that the objective of finding the most representative prototypes is not included in the cost functions of GLVQ as well as for RSLVQ. For some instances both algorithms tend to learn non-typical prototypes, since they are not aiming representivity explicitly. Also in [3] a first approach is given in order to include representivity. This is done by adding a penalty term to the original cost function weighted with a parameter to handle the influence of the penalties. In case of GLVQ finding a suitable parameter is challenging since both objectives differ in their magnitudes. Another drawback for both is that classification performance is not effectively controllable. In this contribution we provide constrained optimization problems (cOP) for classification, where the constraints are described in terms of decision rules coming from LVQ classifiers. Further, the main objective of these cOP's is to maximize class-wise representivity according to the given training samples. However, this is formalized as a class-wise quantization error for GLVQ (GLVQ OP) and a class-wise Gaussian mixture in

case of RSLVQ (RSLVQ OP). Since both main objectives, quantization error as well as the Gaussian mixture model, can be used for clustering, this approach can be interpreted as a constrained clustering method. Typically, as mentioned for instance in [1], constraints describe sets of must-link and cannot-link connections between the data. A must-link constraint specifies that a pair of points connected by the constraint belong to the same cluster (or class in our case). On the other hand, a cannot-link constraint specifies that a pair of points connected by the constraint do not belong to the same cluster. In our approach we only take cannot-link constraints into account. But therefore, we do not compare memberships between pairs of data points. We rather enforce assignments to prototypes in terms of distances or probabilities, respectively. Therefore, we briefly introduce both classification methods to derive the constraints. In section three we explicitly formulate the cOP's for each method and discuss some theoretical properties of them. To demonstrate their functionality we provide classification results for an artificial dataset as well as for a real world dataset available on UCI database. Results are compared to the standard approaches as they are introduced in section two. Practical properties of our cOP's are addressed in the last section, where we also mention research question for future work.

2 The LVQ framework

In this section we briefly describe GLVQ as well as RSLVQ to derive constraints for the resulting optimization problem. Therefore we use the decision rules of the respective classifier.

A LVQ classifier is given by a set of prototypes $\mathbb{W} = \{\boldsymbol{w}_i \mid \boldsymbol{w}_i \in \mathbb{R}^n; i = 1, \ldots, k\}$, where each of them is equipped with a label $c(\boldsymbol{w}_i) \in \{1, \ldots, C\}$, assuming a multi-class problem with C classes. For standard LVQ, classification of a point $\boldsymbol{x} \in \mathbb{R}^n$ takes place by a winner takes all scheme: $\boldsymbol{x}$ is mapped to the label $c(\boldsymbol{x}) = c(\boldsymbol{w}_i)$ of the best matching prototype $\boldsymbol{w}_i$ as measured in some distance measure. In case of the probabilistic RSLVQ classifier a probability connected to the given classes. For simplicity, we restrict ourselves to the Euclidean metric, even though general metrics might be suitable for a certain dataset.

Given a training data set $\mathbb{X} = \{\boldsymbol{x}_j \mid \boldsymbol{x}_j \in \mathbb{R}^n; j = 1, \ldots, m\}$, together with labels $\mathbb{Y} = \{y_j \mid y_j \in \{1, \ldots, C\}; j = 1, \ldots, m\}$, the purpose of LVQ training is finding prototypes such that the resulting classifier achieves a good classification accuracy, i.e. $y_j = c(\boldsymbol{x}_j)$ for as many data $\boldsymbol{x}_j$ out of the training data set. Classical LVQ schemes such as LVQ 1 or LVQ 2.1 take use of a Hebbian learning heuristic. Despite all of its benefits, like classification power, the easiness of updates as well as their interpretation ability, both are not connected to a valid underlying cost function [2]. A few alternative models have been proposed which are derived from explicit cost functions. These costs aim at describing the classification performance of the respective classifier. As it is shown in [6,8] they lead to learning paradigms comparable to the update rules of classical LVQ schemes.

2.1 Generalized learning vector quantization

Generalized LVQ (GLVQ) [6] takes use of the following cost function - a summation over local errors $\mu(\boldsymbol{x}_j)$

$$E = \sum_j f\left(\mu\left(\boldsymbol{x}_j\right)\right) \qquad E = \sum_j f\left(\frac{d^+(\boldsymbol{x}_j) - d^-(\boldsymbol{x}_j)}{d^+(\boldsymbol{x}_j) + d^-(\boldsymbol{x}_j)}\right), \tag{1}$$

where $d^+(\boldsymbol{x}_j)$ belongs to the squared Euclidean distance of $\boldsymbol{x}_j$ to the closest prototype with a matching label. As opposed to that, $d^-(\boldsymbol{x}_j)$ refers to the closest prototype with a non-matching label. Further, the function f relates to a monotonic increasing function. Frequently the identity or the sigmoidal function is used. If f approximates the Heavyside function, e.g. when sigmoidal is used, E is close to the number of misclassifications. Because the numerator becomes negative iff a datum $\boldsymbol{x}_j$ is classified correctly. The denominator prevents divergence and numerical instabilities by normalizing the costs.

$$d^+(\boldsymbol{x}_j) - d^-(\boldsymbol{x}_j) \begin{cases} \leq 0, & \text{if } \boldsymbol{x}_j \text{ classified correct} \\ > 0, & \text{if } \boldsymbol{x}_j \text{ classified wrong} \end{cases} \tag{2}$$

As mentioned in [7], the numerator of the summands can be interpreted as the hypothesis margin of the classifier, such that a large margin and hence good generalization ability is aimed for while training. Hence, the objective is to minimize the GLVQ cost function E. Optimization typically takes place using a gradient technique in terms of online learning.

2.2 Robust soft learning vector quantization

In contrast to GLVQ, robust soft LVQ (RSLVQ) is based on a probabilistic model [8] aiming an optimization of the Bayesian error in terms of a likelihood ratio, but results in similar update rules compared to GLVQ

$$E = \sum_j \log \frac{p(\boldsymbol{x}_j, y_j|W)}{p(\boldsymbol{x}_j|\mathbb{W})} = \sum_j \log p(y_j|\boldsymbol{x}_j, \mathbb{W}). \tag{3}$$

Where $p(y_j|\boldsymbol{x}_j, \mathbb{W})$ describes the probability that a given datum $\boldsymbol{x}_j$ is assigned to its label. Further, $p(\boldsymbol{x}_j|\mathbb{W}) = \sum_i p(\boldsymbol{w}_i)p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ is given as a mixture of Gaussians with prior probability $p(\boldsymbol{w}_i)$ (often taken uniformly over all prototypes). While $p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ describes the probability that $\boldsymbol{x}_j$ is being generated from prototype $\boldsymbol{w}_i$, commonly chosen as an multivariate isotropic Gaussian with mean in $\boldsymbol{w}_i$. Non-isotropic Gaussians slightly extend this approach using a diagonal covariance matrix Σ_i

$$p(\boldsymbol{x}_j|\boldsymbol{w}_i) = \frac{1}{\sqrt{(2\pi)^n|\Sigma_i|}} \exp\left(-\frac{1}{2}\left(\boldsymbol{x}_j - \boldsymbol{w}_i\right)^T \Sigma_i^{-1}\left(\boldsymbol{x}_j - \boldsymbol{w}_i\right)\right), \tag{4}$$

with entries $(\sigma_{i1}^2, \ldots, \sigma_{in}^2)$.

The probability $p(\boldsymbol{x}_j, y_j|\mathbb{W}) = \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p(\boldsymbol{w}_i)p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ (δ - Kronecker delta) corresponds to the mixture components with the correct labelling. The likelihood ratio is optimized using a gradient technique.

Considering a datum $\boldsymbol{x}_j$ with its label y_j. The datum $\boldsymbol{x}_j$ is classified correctly iff

$$p\left(y_j|\boldsymbol{x}_j, \mathbb{W}\right) \geq p\left(y_k|\boldsymbol{x}_j, \mathbb{W}\right) \quad \forall y_k \neq y_j \tag{5}$$

3 LVQ as a constrained optimization problem

In this section we show possibilities how to model LVQ as a constrained optimization problem. Therefore we use the decision rules provided by the GLVQ as well as the RSLVQ classifier.

Enforcing interpretability in GLVQ, we chose the class-wise quantization error as a soft constraint as it is also used in our previous work [3]. Then, the basic constrained optimization problem can be expressed by

$$\begin{aligned} \text{GLVQ OP1:} \qquad & \min \sum_j d^+(\boldsymbol{x}_j) \\ & \text{such that } \; d^+(\boldsymbol{x}_j) \leq d^-(\boldsymbol{x}_j) \quad \forall j. \end{aligned}$$

Following Eq. 2, for each datum $\boldsymbol{x}_j \in \mathbb{X}$ a constraint is build by comparing distances of the closest prototypes with a matching ($d^+(\boldsymbol{x}_j)$) and a non-matching ($d^-(\boldsymbol{x}_j)$) label. If the given classes are well separated, all constraints are easy satisfiable by performing a class-wise clustering according to the objective $\sum_j d^+(\boldsymbol{x}_j)$. Further, in case of providing only one prototype per class, the challenge of finding closest prototypes vanishes. Because, first, for each constraint (datum) the closest correct prototype can be addressed directly. Secondly, to avoid the necessity of finding the closest wrong prototype, we can introduce class-specific constraints. Doing so, it results in $m \times (C-1)$ constraints, forcing that the distance to the correct prototype is smaller then the distance to all of the others.

Of course, this is not given in classification scenarios in general. To take some misclassifications into account, we introduce a specific slack variable $\epsilon_j \geq 0$ to all constraints in case of merged classes, e.g. no feasible solution exists.

In order to keep them small, we incorporate the sum over all slacks to the objective weighted with a parameter $\alpha \geq 0$.

$$\begin{aligned} \text{GLVQ OP2:} \qquad & \min \sum_j d^+(\boldsymbol{x}_j) + \alpha \sum_j \epsilon_j \\ & \text{such that } \; d^+(\boldsymbol{x}_j) \leq d^-(\boldsymbol{x}_j) + \epsilon_j \quad \forall j \end{aligned}$$

For all data points which are safely classified by the prototypes ϵ becomes zero. For all other data points ϵ describes how far the prototype with a matching label must been moved to classify correctly. However, for a given α the solution is a set of prototypes which satisfy the constraints as best as possible under being class-typical as best as possible, together with a set of slack variables ϵ_i indicating misclassifications.

Now, we formulate in an analogous manner a constrained optimization problem based on the decision rules coming from the RSLVQ classifier. But first we provide the main objective invoking class representivity. Therefore, the idea is to use a term which maximizes the likelihood of the observed data being generated by the underlying model. In accordance to RSLVQ, we can consider a class-wise Gaussian mixture model $p(\boldsymbol{x}_j, y_j|\mathbb{W}) = \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p(\boldsymbol{w}_i) p(\boldsymbol{x}_j|\boldsymbol{w}_i)$ with prior probability $p(\boldsymbol{w}_i)$ and Gaussian $p(\boldsymbol{x}_j|\boldsymbol{w}_i)$. This term aims at a generative

model, i.e. we address the class-wise data log likelihood

$$\log \prod_j \delta_{y_j}^c p(\boldsymbol{x}_j|c, W) = \sum_j \delta_{y_j}^c \log \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p_c(\boldsymbol{w}_i) p(\boldsymbol{x}_j|\boldsymbol{w}_i), \tag{6}$$

with prior $p_c(\boldsymbol{w}_i) = p(\boldsymbol{w}_i)/p(c)$ summing to one for every class c.

By having the main objective, we can design an optimization problem including slack variables $\epsilon_{j,c}$ derived from the decision rule Eq. 5

$$\begin{aligned} \text{RSLVQ OP:} \qquad & \max \sum_c \sum_j \delta_{y_j}^c \log \sum_i \delta_{y_j}^{c(\boldsymbol{w}_i)} p_c(\boldsymbol{w}_i) p(\boldsymbol{x}_j|\boldsymbol{w}_i) - \alpha \sum_{j,c} \epsilon_{j,c} \\ & \text{such that } \; p(y_j|\boldsymbol{x}_j, W) \geq p(c|\boldsymbol{x}_j, W) - \epsilon_{j,c} \quad \forall j; \quad \forall c \neq y_j \end{aligned}$$

4 Experiments

For experimental data we take use of an artificial dataset besides a well known real world dataset. For demonstrating representivity and how the parameter α affects the final position of the prototypes, we do the analysis for a three-class problem which consists of two two-dimensional Gaussian clusters with different covariance matrices and some degree of overlap and one cluster being separated from both. Data are randomly generated leading to 1000 points for each class. In order to show the functionality for high-dimensional data also, we use the Tecator dataset [5]. It consists of 215 spectra with 100 spectral bands ranging from 850 nm to 1050 nm. The task is to predict the fat content of the probes. We also compare the found solutions against GLVQ and RSLVQ according to classification performance and representivity.

We only focus on optimization problems GLVQ OP2 and RSLVQ OP. These are solved in Matlab with *fmincon* which is part of the optimization toolbox. For the optimization method itself we use by default an interior-point method without providing gradient informations. Initial values for the prototypes were set to class-means in each case, whereas all ϵ_j were set to zero.

Gauss dataset:
For each class we only spend one prototype. The optimization procedure was executed with four different values of $\alpha = \{0; 2.5; 5; 7.5\}$. What we can observe is that α affects the final position, at least for class 1 and class 2 (see Fig. 1). As expected α has no influence for class 3, since there is no need to move the respective prototype in order to improve classification. According to the main objective its best location is the mean of its class. In contrast to that, prototypes of class 1 and 2 move towards the decision boundary, which improves the classification performance. For $\alpha = 0$, e.g. learning class means, we achieve an accuracy value of 87.27%. However, increasing α to 7.5 leads to a gain of around 3% in accuracy. Obviously, for $\alpha = 7.5$ both prototypes become more or less equal, indicating that an appropriate value ranges between 0 and 7.5.

Tecator dataset:
The dataset itself is indicated in Fig. 2 (left). There the means together with

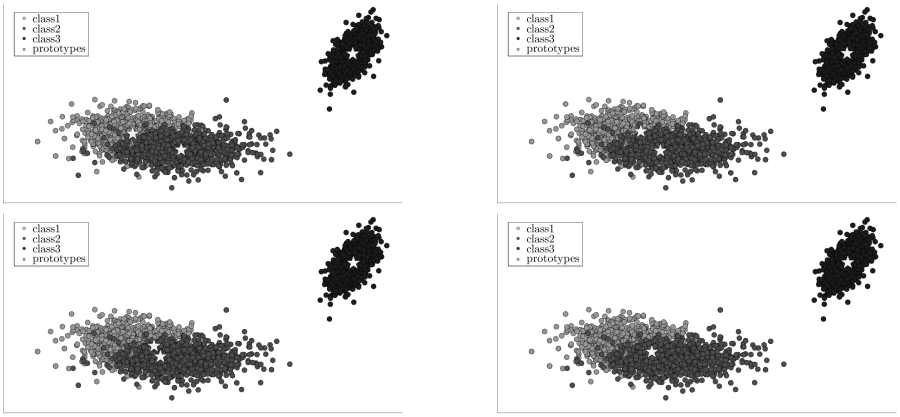


Fig. 1: From the upper left to the bottom rigth: Locations of the resulting prototypes for the Gauss dataset for varying $\alpha \in \{0; 2.5; 5; 7.5\}$. We achieve accuracies of 87.27%, 89.00%, and 90.30% for the last two values of α.

standard deviations are depicted. We can observe that there is some overlap between the classes. For this dataset we performed a simulation over α in the range of $[0, 25]$ ($[0, 50]$ in case of RSLVQ OP) with step size 0.25. We also use only one prototype for each class.

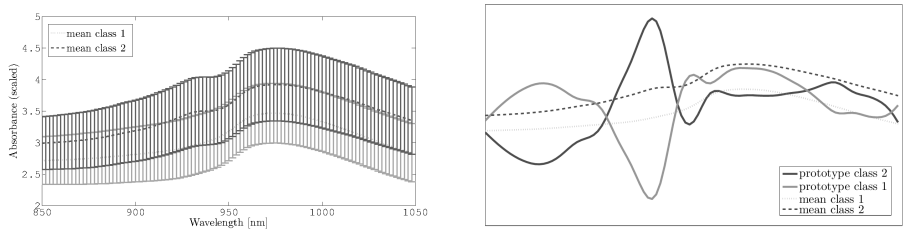


Fig. 2: Left: class means ± standard deviation for Tecator dataset
Right: Prototypes learned with GLVQ with a steep sigmoidal transferfunction. The shape of the resulting prototypes do not look representative compared to the class means. Similar results could be achieved with RSLVQ [3].

Starting with $\alpha = 0$ we end up with prototypes which coincide with the respective class means (see Fig. 4 upper left). After slightly increasing α we can observe an enormous increase in classification performance (see Fig. 3 top row) for training as well as for the test set. In comparison to standard GLVQ we are able to classify the test set error-free with $\alpha \geq 14.5$ for GLVQ OP2. With standard GLVQ only 69% of the training set could be classified correctly. Varying the parameter of the transfer-function in GLVQ improves performance but only with loss in representativity (see Fig. 2 right). In Fig. 4 together with Fig. 3 we can observe that we are able to increase classification performance but in contrast

to GLVQ with class-typical prototypes. It is also noticeable that both prototypes tend to become similar, but they differ slightly in discriminative regions of wavelengths, where both GLVQ and RSLVQ seem to distend in such regions. Comparing both GLVQ OP2 and RSLVQ OP, we can observe that they perform extremely good with comparable results on this dataset. Obviously, GLVQ OP2 is more sensitive to parameter α, because slacks ϵ_j are much larger then in RSLVQ OP, since they are bounded there. Instabilities in training performance are caused by numerics of the solver.

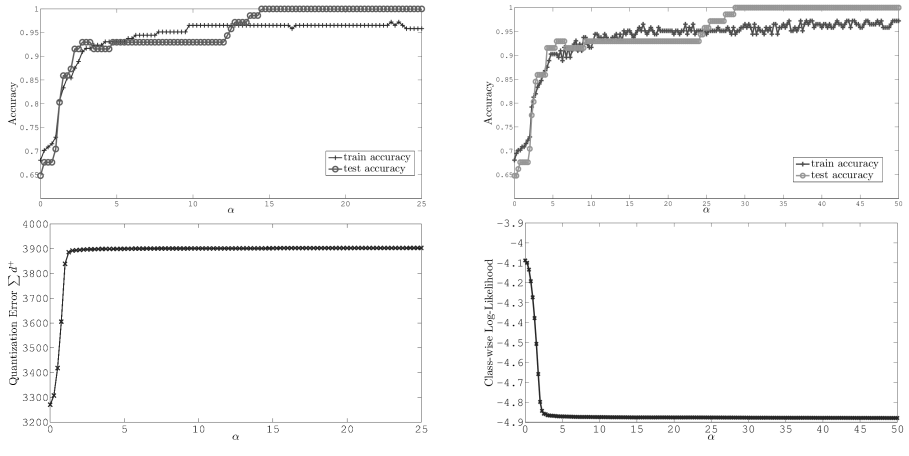


Fig. 3: Top row: Classification performance for Tecator on training and test set: Left: GLVQ OP2; Rigth: RSLVQ OP.
Bottom row: Left: class-wise quantization error; Right: class-wise data log-likelihood. Parameter: $\alpha \in [0, 25]$ for GLVQ OP2; $\alpha \in [0, 50]$ for RSLVQ OP.

5 Discussion

Standard LVQ approaches are very often excellent methods for classification tasks. But, class-representivity is not modelled explicitly. However, discrimination power and interpretability might be contradictory in most of the practical cases. Thus, we actually should treat this issue as a multi-objective optimization problem leading to pareto optimal solutions. In this contribution we designed constrained optimization problems in order to solve classification tasks. Since classification decision is modelled by the constraints while interpretability is maximized, we were able to combine both objectives. Therefore, classification constraints were introduced in terms of decision rules coming from modern LVQ variants. Two formulations (GLVQ OP2 and RSLVQ OP) were given. Both showed excellent performance on two benchmark datasets with comparable results in accuracy and representativity. For the practical usage, resulting prototypes for a given parameter α can be inspected and validated by the user, whether they find favour for a discriminative or a representative solution.

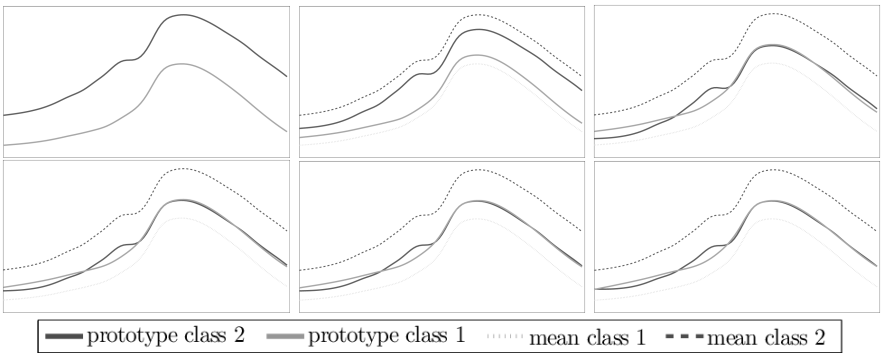


Fig. 4: From the upper left to the bottom rigth: Shape of the resulting prototypes for $\alpha \in \{0; 1.25; 2.25; 12.75; 14.25; 25\}$.

Highlighting RSLVQ OP, we did not take use of a diagonal covariance matrix at the moment. For further research it might be interesting how taking a covariance matrix into account affects the solution. This question is also comparable to use other distance measures for the GLVQ OP formulation. For instance a generalized quadratic form as it is used in [7]. Thus, the question arises if metric learning is doable in this framework. Also kernel distances might be interesting. But it is not clear in how far representivity should be modelled in that case. From the theoretical point of view, getting deeper insights into the nature of the optimization problem is another task for future work. We suggest that the solution is unique in case of using only one prototype for each class since the objective itself is convex.

Acknowledgement

BH has been supported by the CITEC center of excellence. DN and MR acknowledge funding by ESF.

References

1. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666, 2010.
2. M. Biehl, B. Hammer, P. Schneider, and T. Villmann. Metric learning for prototype based classification. In *Innovations in Neural Information – Paradigms and Applications*. Springer, 2009.
3. B. Hammer, D. Nebel, M. Riedel, and T. Villmann. Generative versus discriminative prototype based classification. In *Advances in Self-Organizing Maps and Learning Vector Quantization*, volume 295, pages 123–132. Springer, 2014.
4. T. Kohonen. The self-organizing map. *Proc. of the IEEE*, 78(9):1464–1480, 1990.
5. D. of Statistics at Carnegie Mellon University. http://lib.stat.cmu.edu/datasets/.
6. A. Sato and K. Yamada. Generalized learning vector quantization. In *Advances in Neural Information Processing Systems*, pages 423–9. MIT Press, 1996.
7. P. Schneider, M. Biehl, and B. Hammer. Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21:3532–3561, 2009.
8. S. Seo and K. Obermayer. Soft learning vector quantization. *Neural Computation*, 15(7):1589–1604, 2003.

Prior knowledge for Core Vector Data Description

Frank-Michael Schleif[1], Xibin Zhu[2], Barbara Hammer[2]

[1] University of Birmingham, School of Computer Science
Edgbaston, Birmingham B15 2TT United Kingdom
Email: schleify@cs.bham.ac.uk
[2] University of Bielefeld, Faculty of Technology, CITEC
D–33594 Bielefeld, Germany
Email: {xzhu,bhammer}@techfak.uni-bielefeld.de

Abstract. Core vector data description aims at outlier detection using kernelized linear separations such as well known from support vector machines [15], but relying on an equivalent formulation for its efficient optimization as a minimum enclosing ball problem [16]. Besides the mere training data, auxiliary information might be available such as e.g. monotonicity of the mapping prescription. In this contribution we investigate in how far auxiliary information can be used in core vector data descriptions, thus yielding to faster and sparser models as compared to its counterparts which rely on a direct optimization.

1 Introduction

While machine learning methods such as support vector machines (SVM) provide state of the art classifiers with numerous successful industrial and scientific applications, they suffer not only from the fact that the decision function is given as a black box mechanism [12], but also the resulting models can be very large if large data sets are dealt with.

There exists a variety of technologies to avoid the black box character of machine learning techniques such as SVM: examples include sparse modeling [4], relevance learning [8], or explicit rule extraction [5]. One specific technology is to explicitly constrain the functional form of the model by incorporating prior knowledge [10, 11]: a wide range of constraints can be expressed in terms of derivatives of the function prescription. One key step is to express the derivatives of the function in a suitable form such that its constraints can be included into the training pipeline. This way, the form of the model is restricted to a shape which takes into account priorly known invariances of the application. Instantiations of this principle deal with monotonicity constraints for function approximation [14], the incorporation of functional characteristics [6, 9], or the incorporation of symmetries in the functional form [13], to name a few recent approaches. In this contribution, we are interested in ways to integrate functional knowledge of a general form into a machine learning model.

The addition of constraints enables an improved generalization ability of the model for data regions which are covered by incorporated invariances rather than explicit training data. Still, the resulting models can be very large provided large data sets are dealt with. The technique proposed in [16] provides one elegant way to arrive at considerably

sparser solutions. Relying on a technique which is typically referred to as *core* method. The key observation underlying this core technology is that a small subset actually suffices to characterize the full data approximately: a connected geometric problem, the minimum enclosing ball problem, can be approximately solved by efficient geometric algorithms as proved in [2], and which induce an approximate solution of the SVM problem via this solution. While leading to very sparse solutions for the classical unconstrained models, the formalism as introduced in [16] does not incorporate further prior constraints on the function.

In this contribution, we are interested in the question whether the addition of constraints is possible for core techniques to incorporate prior knowledge. We propose an extension of the core vector data description technique [16] which realizes data description [15] towards constraints which are expressed as linear inequalities of the function and its derivatives. This allows us to iteratively construct a core set which approximately describes the given data under constraints on the solution.

2 Core vector data description

We will first introduce support vector data description (SVDD) and its relation to core algorithms via a link of the dual problem of SVDD to a geometric problem, the minimum enclosing ball problem (MEB).

Assume data $\boldsymbol{x}_i \in S \subset R^n$ are given. Assume a fixed kernel $k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is chosen which is associated to the feature map Φ: $k(\boldsymbol{x}, \boldsymbol{y}) = \Phi(\boldsymbol{x})^t \Phi(\boldsymbol{y})$. The goal of support vector data description (SVDD) [15] is to find a generalized linear mapping

$$\boldsymbol{x} \mapsto \operatorname{sgn}(f(\boldsymbol{x})) = \operatorname{sgn}(\boldsymbol{w}^t \Phi(\boldsymbol{x}) - \rho)$$

which defines a separation of the given data to outliers by means of its sign, whereby the separation boundary corresponds to a linear separation in the feature space induced by Φ. The problem to find suitable parameters $\boldsymbol{w}$ and ρ for a given data set S can be formalized as optimization problem which aims at a separation of the given data from the origin with maximum margin. This leads to the following primal optimization problem:

$$\begin{array}{ll} \mathrm{SVDD(primal)} & \\ \min_{\boldsymbol{w},\rho,\xi_i} & \frac{1}{2} \cdot ||\boldsymbol{w}||^2 - \rho + \frac{C}{2} \cdot \sum_i \xi_i^2 \\ \text{such that} & \boldsymbol{w}^t \Phi(\boldsymbol{x}_i) \geq \rho - \xi_i \quad \forall i \end{array}$$

where $C > 0$ is a fixed constant, and the parameters ξ_i refer to the slack variables to allow for potential errors. Using the Karush-Kuhn-Tucker (KKT) conditions, the Lagrange dual problem becomes

$$\begin{array}{ll} \mathrm{SVDD(dual)} & \\ \max_{\alpha_i} & -\frac{1}{2} \cdot \sum_{ij} k(\boldsymbol{x}_i, \boldsymbol{x}_j) \alpha_i \alpha_j + \frac{1}{C} \cdot \sum_i \alpha_i^2 \\ \text{such that} & \alpha_i \geq 0 \quad \forall i \\ & \sum_i \alpha_i = 1 \end{array}$$

This dual problem can be directly optimized relying on linearly constraint convex quadratic optimization. The solution $\boldsymbol{w}$ and ρ of the primal problem can then be recovered from

the dual variables α_i. These are non-vanishing for support vectors only, hence we arrive at a sparse description. Still, its size is increasing with the size of the sample set S. Empirically, a linear dependency can usually be observed.

Instead of a direct optimization of this formalization, the approaches [16] propose to solve this problem in an iterative fashion, linking it to a geometric problem, the *minimum enclosing ball* problem (MEB). This problem consists in the objective to find a minimum ball which contains all data points $\boldsymbol{x}_i$. In kernelized form, its primal objective is given as

$$\begin{array}{ll} \mathrm{MEB(primal)} & \\ \min_{R^2,\boldsymbol{c}} & R^2 \\ \text{such hat} & ||\boldsymbol{c} - \Phi(\boldsymbol{x}_i)||^2 \leq R^2 \quad \forall i \end{array}$$

where $\boldsymbol{c}$ denoted the centre and R the radius of the ball. Again, the KKT conditions allow to simplify the Lagrangian dual to obtain the following form

$$\begin{array}{ll} \mathrm{MEB(dual)} & \\ \max_{\alpha_i} & -\frac{1}{2} \cdot \sum_{ij} k(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_i\alpha_j + \sum_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) \\ \text{such that} & \alpha_i \geq 0 \quad \forall i \\ & \sum_i \alpha_i = 1 \end{array}$$

Obviously, the dual SVDD and the dual MEB are equivalent provided $k(\boldsymbol{x}_i, \boldsymbol{x}_i) = \mathrm{const}$, which holds for the Gaussian kernel or normalized kernels, for example. This means that the SVDD and MEB are equivalent, and optimal dual variables α_i simultaneously offer an optimum solutions for both problems. Hence, instead of an optimization of SVDD, we can optimize MEB or its dual. Having solved the dual MEB (or dual SVDD), the KKT conditions allow us to retrieve a solution for the primal SVDD problem from the dual variables of the MEB because of the relation $\boldsymbol{w} = \sum_i \alpha_i \Phi(\boldsymbol{x}_i)$ and $\rho = \sum_{i,j} \alpha_i\alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_i \alpha_i^2/C$. Note that this representation leads to a description of outliers in terms of the kernel evaluated for the support vectors for which $\alpha_i \neq 0$ holds:

$$\boldsymbol{x} \mapsto \mathrm{sgn}\big(\sum_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) - \rho\big)$$

The question occurs whether there are alternatives to arrive at even sparser descriptions of the data. As pointed out in [2], there exists an efficient geometric algorithm which approximately solves the MEB problem and which eventually leads to even sparser solutions of the MEB with constant size support vector data descriptions. The proposed algorithm relies on the notion of core sets: given a set S and $\epsilon > 0$, a *core set* S_0 is a subset of S, $S_0 \subset S$, such that the following holds: assume R_0 and $\boldsymbol{c}_0$ refer to the centre and radius, respectively, of a minimum enclosing ball for S_0. Then it holds for all $\boldsymbol{x} \in S$ that

$$||\boldsymbol{x} - \boldsymbol{c}_0||^2 \leq R_0^2(1+\epsilon)^2$$

Hence the optimum solution of the MEB for the core set S_0 of S induces an ϵ-*approximate* solution for the whole set S.

Core algorithm:
choose $S_0 := \{\Phi(\boldsymbol{x}_i)\}$ for a random data index i
choose $\Phi(\boldsymbol{x}_j) \in S \backslash S_0$ with maximum $||\Phi(\boldsymbol{x}_i) - \Phi(\boldsymbol{x}_j)||^2$
set $S_0 := S_0 \cup \{\Phi(\boldsymbol{x}_j)\}$
repeat
    solve the MEB problem for S_0,
    this gives centre $\boldsymbol{c}$ and radius r
    if $\exists \Phi(\boldsymbol{x}_k) \in S$ with $|\Phi(\boldsymbol{x}_k) - \boldsymbol{c}|^2 > r^2(1+\epsilon)$
        set $S_0 := S_0 \cup \{\Phi(\boldsymbol{x}_k)\}$
until no such data can be found

Per construction, this algorithm terminates with a core set S_0 of S and an approximate solution for the SVDD for S represented by the solution for S_0.

Note that all steps can be solved relying on the kernel only rather than the feature map Φ: we can compute distances on the feature space based on kernels only. Further, we can formulate the MEB via its dual, resulting in a kernel form which yields solutions for the dual variables α_i which approximately solve the MEB, and, hence, also the SVDD.

Since S_0 is usually much smaller than S, this algorithm is faster than a direct optimization of the MEB (or the SVDD) for the full data set S. Further, the resulting solution is usually much smaller. Actually, it has been proven in [2] that a constant size set S_0 will suffice where the size depends on the quality of the approximation ϵ only. Hence a linear time algorithm and constant size support vector data description results. We refer to this approximate solution of the SVDD as core vector data description (CVDD) in the following.

3 Integration of prior knowledge

Often, prior knowledge of the learning problem is available in the form of constraints of the function f. Typical examples include the following settings:

- There is a priorly limited range U of acceptable values for the data, i.e. $f(\boldsymbol{x}) = -1$ for $\boldsymbol{x} \notin U$.
- Monotonicity of the function with respect to some or all coefficient dimensions R holds, i.e. $f(\boldsymbol{x}) = -1 \Rightarrow f(\boldsymbol{y}) = -1$ for all $\boldsymbol{y}$ with $y_k = x_k$ for $k \notin R$ and $y_k \geq x_k$ for $k \in R$.
- Limited variability/smoothness of the outlier function is given as indicated by a limited curvature of f.
- etc.

Such restrictions might be caused by knowledge about the sensitivity of sensors or their maximum range, for example.

Notably, all restrictions as specified above can be expressed in terms of (sets of) inequality constraints on the function f and its derivatives: monotonicity at a point $\boldsymbol{x}_i$ is equivalent to the derivative being positive at this position, a limitation of the range can be expressed as inequality $f(\boldsymbol{x}_i) < 0$ for points directly at the boundary, the curvature of the function can be expressed in terms of its second derivative, etc. See [11] for more details about these formulations.

Here, we formulate this approach in terms of SVDD. Essentially, the main observation is that constraints on the function f can be expressed efficiently using its dual representation

$$f(\boldsymbol{x}) = \sum_i \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i) - \rho$$

Hence derivatives with respect to coefficient k of $\boldsymbol{x}$ can be expressed as

$$\frac{\partial f(\boldsymbol{x})}{\partial x_k} = \sum_i \alpha_i \frac{\partial k(\boldsymbol{x}, \boldsymbol{x}_i)}{\partial x_k} - \rho$$

This constitutes a linear term in the dual variables α_i which depends on the derivatives of the kernel. For the Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}_i) = \exp(-||\boldsymbol{x} - \boldsymbol{x}_i||^2/(2\sigma^2))$, as an example, this term yields the linear term expression

$$\frac{1}{\sigma^2} \left(\mathbf{X}_j - \mathbf{1}\boldsymbol{x}_k\right)^t \operatorname{diag}\left(K\left(\boldsymbol{x}, \mathbf{X}^t\right)\right) \boldsymbol{\alpha}$$

where $\boldsymbol{\alpha}$ is the vector of dual parameters α_i, $\mathbf{X}$ is the data matrix, $\mathbf{X}_j$ its jth column, $\mathbf{1}$ refers to the vector with entries 1 and dimensionality equal to the number of data points, $\mathbf{K}$ refers to the Gram matrix evaluated at the indicated points, and diag is the operation which turns a vector into a matrix with the vector as diagonal. Similarly, higher order derivatives can also be expressed as linear terms of the dual variables α_i.

This observation offers an easy way to integrate constraints into SVDD which are linear in terms of the function $f(\boldsymbol{x})$ and its derivatives: Assume a linear constraint depending on f and its derivatives is given for some points $\boldsymbol{x}_i$. Then, because of the linearity of the derivatives of f with respect to α_i, we arrive at constraints of the form $L(\boldsymbol{\alpha}, \boldsymbol{x}_i) \geq 0$ with a linear term L. Hence we can easily enrich the dual SVDD by these constraints for any given finite set of points for which these constraints should be satisfied, using standard convex quadratic solvers for its optimization. Obviously, the inequalities can be enriched by slack variables in the standard way if a feasible solution cannot be guaranteed otherwise. We refer to this technique as SVDD with constraints (SVDD-C) in the following.

Putting a possibly high number of constraints to the dual SVDD results in an even less sparse support vector data description. Here we are interested in possibilities to integrate the core technique into SVDD-C to arrive at sparser solutions. We assume that there exists a number of constraints on f which can be expressed as linear constraints $L(\boldsymbol{\alpha}, \boldsymbol{x}_i) \geq 0$ for the dual parameters and some data points $\boldsymbol{x}_i$. Note that the core algorithm can be formalized in terms of dual variables only, such that we can easily enrich the algorithm by corresponding constraints. Assume I_S refers to the *indices* of all considered points in S:

Constraint core algorithm:

choose $I_{S_0} := \{i\}$ for random i
choose $j \in I_S \backslash I_{S_0}$ with maximum $||\Phi(\boldsymbol{x}_i) - \phi(\boldsymbol{x}_j)||^2$
set $I_{S_0} := I_{S_0} \cup \{j\}$
repeat
  solve the dual problem restricted to α_i where $i \in I_{S_0}$:
  $\max_{\alpha_i} \quad -\frac{1}{2} \cdot \sum_{ij} k(\boldsymbol{x}_i, \boldsymbol{x}_j)\alpha_i\alpha_j + \sum_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_i)$
  such that $\alpha_i \geq 0, \ \sum_i \alpha_i = 1$
  $L(\boldsymbol{\alpha}, \boldsymbol{x}_i) \geq 0$
  this gives centre $\boldsymbol{c}$ and radius r in kernel form:
    $\boldsymbol{c} = \sum_i \alpha_i \Phi(\boldsymbol{x}_i)$
    $R^2 = \sum_i \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}_i) - \sum_{ij} \alpha_i\alpha_j k(\boldsymbol{x}_i, \boldsymbol{x}_j)$
  if $\exists k \in S$ with
    $k(\boldsymbol{x}_k, \boldsymbol{x}_k) - 2\sum_i \alpha_i k(\boldsymbol{x}_k, \boldsymbol{x}_i) + \sum_{ij} k(\boldsymbol{x}_i, \boldsymbol{x}_j) > r^2(1+\epsilon)$
    or $L(\boldsymbol{\alpha}, \boldsymbol{x}_i) < 0$:
    set $I_{S_0} := I_{S_0} \cup \{k\}$
until no such index k can be found

Obviously, this algorithm yields an approximate solution for SVDD-C per construction. We refer to the method as CVDD-C in the following.

Unlike SVDD-C, the method starts with a small subset of the data, such that a sparse support vector data description can be expected which fulfills the additional constraints. We will confirm this expectation in experiments. Note that CVDD-C corresponds to the geometric problem posed by MEB where the location of the centre is limited by linear constraints. Since these constraints can be arbitrary if there are no restrictions on the kernel, we can no longer guarantee a fixed size core set in the worst case since the problem contains the problem to find a feasible solution for a given LP with arbitrary dimensionality as a subproblem. Still, the solution set is always smaller than the solution set found by SVDD-C.

4 Experiments

In the following we consider three different data sets to evaluate the Core Vector Data Description (CVDD) and the Support Vector Data Description (SVDD) with and without constraints. For all data sets, we have a set of *non*-outliers which constitutes the training set, and a set of data points for testing, which is given by a mix of outliers and non-outliers. In all settings, we incorporate prior knowledge by a monotonicity constraint as concerns one component of the data, as specified below. The tolerance parameter of the CVDD was set to $\epsilon = 0.005$.

DS1: The first dataset consists of three two-dimensional Gaussians. The first Gaussian generates the known *non*-outlier data. It is defined as $G_1(\boldsymbol{x}) = \exp\left(-\frac{(x_1-v_1)^2}{2\cdot 0.5^2} - \frac{(x_2-v_2)^2}{2\cdot 2^2}\right)$, $\boldsymbol{v} = (0, 0)$, and $\boldsymbol{\sigma} = (0.5, 2)$ with $N = 530$ points (500 for training and 30 for tests). G_2 is defined as a gaussian with centre $(-2.7, -0.08)$ and variances $\boldsymbol{\sigma} = (1.1, 0.06)$. G_3 is a Gaussian with center $(2.7, 0.08)$ and variances $\boldsymbol{\sigma} = (1.1, 0.06)$ both with 30 samples each. G_2 specifies the outlier data and G_3 is considered as unknown data which are expected to be *non*-outliers. For parameter studies 100 points where removed from the *non*-outlier points.

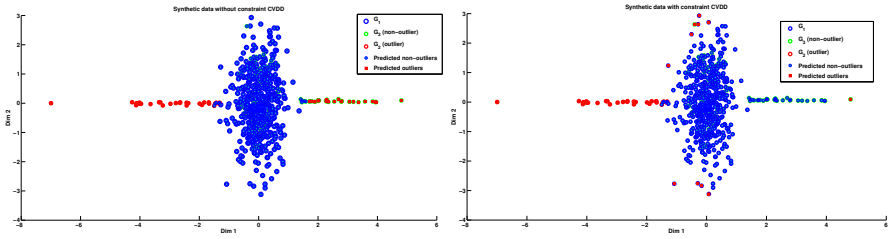


Fig. 1. Left: synthetic data set DS1 without constraints using CVDD. One can observe that all entries not belonging to G_1 are considered to be outliers. Right: the same data but trained using the constraint that f is monotonic in the first dimension. This enables a correct identification of the rightmost Gaussian, which does not contain outliers per construction.

The prior knowledge to be integrated consists in a monotonicity of f with respect to x_1, hence realizing that G_3 are *non*-outliers, which cannot be inferred directly from the data.

DS2: The second dataset (DS2) is the well known *Breast-cancer-Wisconsin* dataset available at UCI[3] with 699 samples and 9 features[4], normalized to $N(0,1)$ for each dimension. Originally proposed as a classification problem we will consider it as an outlier task. Thereby the known *non*-outliers used for training are points with values $\mathbf{x}_2 \in (-0.4, 1.0)$ of class 0 in the original data. The outlier data are formed by the class 1. In addition, the test set contains unobserved *non*-outliers which are data from class 0 with value $x_2 > 1$. For parameter studies, 20 points where removed from the *non*-outlier points. We incorporate monotonicity of f with respect to dimension x_2 as a constraint. The suitability of this constraint is supported by the decision tree classification as found in [3]. It accounts for the fact that *non*-outliers with $x_2 > 1$ can be detected, albeit not present in the training data.

DS3: The third dataset (DS3) is an extended version of DS1 with $N = 10000$ samples for the first Gaussian to test for the scalability of the approaches.

Results are shown in Table 1. For DS1, CVDD as well as SVDD can correctly describe the data, but the description is not able to identify the rightmost Gaussian as *non*-outlier, since these data are not represented in the training set. Incorporating constraints enables us to do so, as shown in Fig. 1. This effect is mirrored by an increase of

[3] http://archive.ics.uci.edu/ml

[4] Here we used the version provided in the Matlab Neural Network Toolbox

	DS1	DS2	DS3
SVDD	63.33|300|27.31	67.41|20|0.30	$n.a.$
CVDD	63.33|29|1.0	78.10|33|0.07	65.56|24|0.10
SVDD-C	81.11|300|29.52	90.17|20|0.50	$n.a.$
CVDD-C	93.33|4|1.36	89.00|29|0.23	84.44|4|27.96

Table 1. Test set accuracies for three outlier datasets. We also provide the runtime and number of support/core vectors in the model as tuple (accuracy|runtime|complexity).

the classification accuracy by 20% for SVDD-C versus SVDD and by 30% for CVDD-C versus CVDD. Interestingly, the size of SVDD and SVDD-C as compared to CVDD and CVDD-C is larger by one or two orders of magnitude, respectively. Hence integrating core techniques enables a sparser description of the data in these cases. Similarly, the computation time is reduced by two orders of magnitude due to the limited size problems which are solved for CVDD(-C) in comparison to SVDD(-C).

This effect becomes even more pronounced for DS3 with the same statistical characteristics but enlarged to 10000 samples. While CVDD is able to describe the data with core sets which have the same size as for the small problem, and the running time increases linearly only, the mere SVDD(-C) implementation did not converge within the given time limit. The classification accuracy of CVDD-C versus CVDD is improved by about 20%.

For DS2, results are similar albeit less pronounced: incorporating constraints allows a significant improvement of the classification accuracy, this way extending the support of the data description towards regions of the data space where no training examples have been present, but the constraints specify the structural invariance. The number of core vectors is not significantly reduced, being small already for the direct optimization. In consequence, training times are only mildly decreased for the core techniques (by a factor of two only for CVDD-C versus SVDD-C) since they involve an iteration over several small size optimization problems instead of only one for the direct method.

5 Conclusions

We have investigated the possibility to integrate prior knowledge and fast core methods in one approach. Due to the core technology, we arrive at an efficient and sparse core vector data description which is particularly suited for incremental settings. Further, the method generalizes beyond the data support due to the auxiliary information. We have developed an explicit algorithm which combines the core vector technique with auxiliary constraints which are represented as linear constraints of the data description functions and its derivatives. Albeit a fixed size core set can no longer be guaranteed for arbitrary kernels for principled reasons (LP being a subproblem if arbitrary kernels are permitted), small size core vector data descriptions have been reached in practical benchmarks, enabling training also for large data sets due to its speedup by several orders of magnitude. Thus, this offers a promising technique for modeling big data.

We expect that similar techniques can be used to constrain core vector classification or core vector regression based on auxiliary information such as a limited range, limited curvature, or similar. It will be a subject of future work to test the results for regression tasks and alternative constraints. Further, the test of the technology in settings with streaming data, thereby relying on the achieved sparse representation for presented windows similar to the patch approach as proposed in [1, 7] for (relational) clustering is the subject of ongoing work.

Acknowledgment

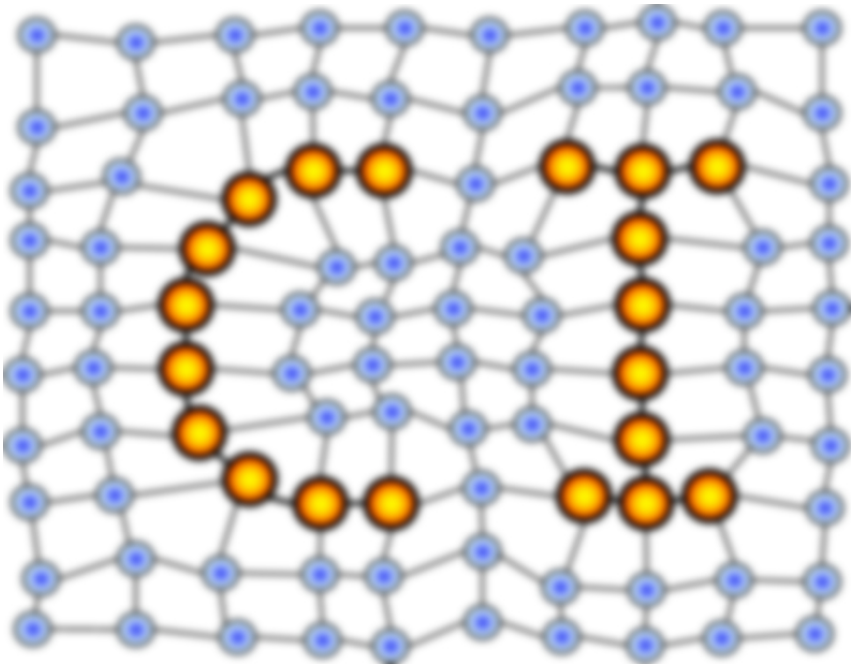
This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the Leading-Edge Cluster Competition and managed by the Project Management Agency Karlsruhe (PTKA). The authors are responsible for the contents of this publication. Funding in the frame of the centre of excellence 'Cognitive Interaction Technologies' (CITEC) is gratefully acknowledged. The first author was kindly supported by a Marie Curie Intra-European Fellowship (IEF) FP7-PEOPLE-2012-IEF (FP7-327791-ProMoS).

References

1. Alex, N., Hasenfuss, A., Hammer, B.: Patch clustering for massive data sets. Neurocomputing 72(7-9), 1455–1469 (2009)
2. Badoiu, M., Clarkson, K.L.: Optimal core-sets for balls. Comput. Geom. 40(1), 14–22 (2008)
3. Bennett, K.P.: Decision Tree Construction Via Linear Programming. In: Evans, M. (ed.) Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference. pp. 97–101. Utica, Illinois (1992)
4. Bhan, N., Baldassarre, L., Cevher, V.: Tractability of interpretability via selection of group-sparse models. In: Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on. pp. 1037–1041 (2013)
5. Diederich, J. (ed.): Rule Extraction from Support Vector Machines, Studies in Computational Intelligence, vol. 80. Springer (2008)
6. Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C., Garcia, R.: Incorporating functional knowledge in neural networks. Journal of Machine Learning Research 10, 1239–1262 (2009)
7. Hammer, B., Hasenfuss, A.: Topographic mapping of large dissimilarity data sets. Neural Computation 22(9), 2229–2284 (2010)
8. Hammer, B., Villmann, T.: Generalized relevance learning vector quantization. Neural Networks 15, 1059–1068 (2002)
9. Han, F., Huang, D.S.: A new constrained learning algorithm for function approximation by encoding *a priori* information into feedforward neural networks. Neural Computing and Applications 17(5-6), 433–439 (2008)
10. Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector machines for classification: A review. Neurocomputing 71(7-9), 1578–1594 (2008)
11. Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector regression. Machine Learning 70(1), 89–118 (2008)
12. Lisboa, P.J.G.: Interpretability in machine learning - principles and practice. In: Masulli, F., Pasi, G., Yager, R.R. (eds.) WILF. Lecture Notes in Computer Science, vol. 8256, pp. 15–21. Springer (2013)
13. Liu, X., Li, P., Gao, C.: Symmetric extreme learning machine. Neural Computing and Applications 22(3-4), 551–558 (2013)
14. Minin, A., Velikova, M., Lang, B., Daniels, H.: Comparison of universal approximators incorporating partial monotonicity by structure. Neural Networks 23(4), 471–475 (2010)
15. Tax, D.M.J., Duin, R.P.W.: Support vector data description. Machine Learning 54(1), 45–66 (2004)
16. Tsang, I.W., Kwok, J.T., Cheung, P.M.: Core vector machines: Fast svm training on very large data sets. Journal of Machine Learning Research 6, 363–392 (2005)

MACHINE LEARNING REPORTS

Report 02/2014



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.